

---

# Masterarbeit

---

Rocco Schenk

**Der Aufstieg von Low Code:**

**Ein Leitfaden für die Einführung  
von Low Code in deutschen  
Unternehmen**

Mittweida, 2022

## **Masterarbeit**

---

### **Der Aufstieg von Low Code:**

### **Ein Leitfaden für die Einführung von Low Code in deutschen Unternehmen**

Autor:

**Rocco Schenk**

Studiengang:

**Industrial Management (M.Sc.)**

Seminargruppe:

**ZM20w1-M**

Erstprüfer:

**Prof. Dr. Erich Dräger**

Zweitprüfer:

**LL.M. Tino Janke**

Einreichung:

**Mittweida, 28. Oktober 2022**

## **Masterthesis**

---

# **The Rise of Low Code: A Guide for the Implementa- tion of Low Code in German Companies**

author:

**Rocco Schenk**

course of studies:

**Industrial Management (M.Sc.)**

seminar group:

**ZM20w1-M**

first examiner:

**Prof. Dr. Erich Dräger**

second examiner:

**LL.M. Tino Janke**

submission:

**Mittweida, 28. October 2022**

## **Bibliografische Beschreibung:**

Schenk, Rocco:

Der Aufstieg von Low Code:

Ein Leitfaden für die Einführung von Low Code in deutschen Unternehmen

The Rise of Low Code:

A Guide for the Implementation of Low Code in German Companies

112 Seiten, Hochschule Mittweida, University of Applied Sciences, Institut für Wissenstransfer und Digitale Transformation, Masterarbeit, 2022

## **Abstract:**

In dieser Masterthesis wird die Thematik eines Einführungsprojekts für No Code/ Low Code-Plattform behandelt. Die Zielstellung der Arbeit ist die Entwicklung eines Leitfadens für die Einführung von No Code/ Low Code-Projekten. Zur Realisierung dieses Forschungsziels wurden 8 problemzentrierte Experteninterviews geführt. Diese Interviews wurden nach der Verschriftlichung mittels der qualitativen Inhaltsanalyse nach Kuckartz ausgewertet. Im speziellen wurde hier die strukturierende qualitative Inhaltsanalyse angewendet. Die Evaluation der Ergebnisse hat aufgezeigt, dass die meisten Experten während der aktiven Phase der Einführung einer solchen Plattform eine Form des agilen Vorgehens bevorzugen. Als überraschend hat sich herausgestellt, dass die Vorbereitung des Projekts einen großen Teil der Implementierung einnimmt und oft vernachlässigt wird. Aus diesen Erkenntnissen haben sich zwei mögliche Vorgehensweisen herausgestellt. Die eine ist ein 2-Projekt-Design, bei dem ein Vorbereitungsprojekt agil oder klassisch durchgeführt wird. Das zweite Projekt begleitet danach die aktive Phase der Einführung in einer agilen Vorgehensweise.

Die zweite mögliche Vorgehensweise ist ein hybrides Vorgehen, welches in einem Wasserfallrahmen stattfindet. In beiden Fällen wird die aktive Implementierungsphase als agiles Vorgehen durchgeführt.

# Inhalt

<b>Inhalt</b>	.....	<b>I</b>
<b>Abbildungsverzeichnis</b>	.....	<b>III</b>
<b>Tabellenverzeichnis</b>	.....	<b>IV</b>
<b>Abkürzungsverzeichnis</b>	.....	<b>V</b>
<b>1 Einleitung</b>	.....	<b>1</b>
1.1	<i>Forschungsfragen und Methodik</i> .....	2
1.2	<i>Aufbau der Arbeit</i> .....	3
<b>2 Low Code</b>	.....	<b>4</b>
2.1	<i>Das Problem des Fachkräftemangels</i> .....	4
2.2	<i>No Code/Low Code als Lösungsansatz</i> .....	5
2.2.1	<i>Vor- und Nachteile von No Code/Low Code</i> .....	6
2.2.2	<i>Anwendung von No Code/Low Code</i> .....	10
2.3	<i>Aufstieg von Citizen Development</i> .....	12
2.4	<i>Einordnung von No Code/Low Code im Softwareumfeld</i> .....	14
2.4.1	<i>Formen von Software im Unternehmen</i> .....	14
2.4.2	<i>Einordnung von No Code/Low Code</i> .....	16
<b>3 Vorgehensmodelle im Projektmanagement</b>	.....	<b>18</b>
3.1	<i>Lineare/Sequenzielle Modelle</i> .....	18
3.1.1	<i>Wasserfallmodell</i> .....	22
3.1.2	<i>V-Modell</i> .....	23
3.2	<i>Agiles Projektmanagement</i> .....	25
3.2.1	<i>Scrum</i> .....	27
3.2.2	<i>Kanban</i> .....	30
3.3	<i>Hybrides Projektmanagement</i> .....	33
3.4	<i>Change Projektmanagement</i> .....	34
3.5	<i>Zusammenfassung Kapitel Vorgehensmodelle</i> .....	36
<b>4 Forschungsfragen und Ablauf</b>	.....	<b>39</b>
4.1	<i>Das Interview als Teil der Forschung</i> .....	39
4.2	<i>Aufbau Leitfaden für Interview</i> .....	41
4.3	<i>Auswertung Interviews</i> .....	42
<b>5 Vorstellung Forschungsmethodik</b>	.....	<b>43</b>
5.1	<i>Theoretische Methodik</i> .....	43
5.1.1	<i>Qualitative vs. Quantitative Forschung</i> .....	43
5.1.2	<i>Vorgehen nach Kuckartz</i> .....	45
<b>6 Durchführung der Analyse</b>	.....	<b>51</b>
6.1	<i>Ablaufmodell der Analyse</i> .....	51
6.2	<i>Charakterisierung Ausgangsmaterial</i> .....	52
6.3	<i>Deduktive Kategoriendefinition</i> .....	54

II	Inhalt
6.4	<i>1. Codierung des Materials</i> ..... 55
6.5	<i>Erstellen von Subkategorien und 2. Codierungsvorgang</i> ..... 56
6.6	<i>Kategorisierung und Analyse</i> ..... 57
<b>7</b>	<b>Ergebnispräsentation</b> ..... <b>58</b>
7.1	<i>Kategoriensystem mit Subcodes</i> ..... 58
7.2	<i>Kategorienezusammenfassung</i> ..... 60
<b>8</b>	<b>Diskussion</b> ..... <b>90</b>
8.1	<i>Auswirkungen von NCLC-Plattformen auf Unternehmen</i> ..... 90
8.1.1	<i>Veränderte Arbeitsweisen</i> ..... 90
8.1.2	<i>Prozessveränderungen</i> ..... 91
8.1.3	<i>Kultur des Unternehmens</i> ..... 92
8.2	<i>Vorphase des Einführungsprojekts</i> ..... 93
8.2.1	<i>Organisatorische Vorbereitung</i> ..... 93
8.2.2	<i>Governance-, Daten- und IT-Sicherheitsrichtlinien</i> ..... 94
8.2.3	<i>Changemanagement</i> ..... 95
8.2.4	<i>Vorgehensweisen Initialisierungsphase/Initialisierungsprojekt</i> ..... 96
8.3	<i>Umsetzung der Implementierung</i> ..... 99
8.3.1	<i>Projektorganisation</i> ..... 99
8.3.2	<i>Vorgehensmodell agile Phase</i> ..... 101
8.3.3	<i>Skalierung und Befähigung</i> ..... 103
8.3.4	<i>Nachbereitung des Projekts</i> ..... 106
8.4	<i>Begrenzung der Forschung</i> ..... 106
<b>9</b>	<b>Schlussbetrachtung</b> ..... <b>109</b>
<b>10</b>	<b>Literaturverzeichnis</b> ..... <b>112</b>
<b>Anlagen</b>	..... <b>119</b>
<b>Anlage 1, Interviewleitfaden</b>	..... <b>I</b>
<b>Anlage 2, Interview EB01</b>	..... <b>VI</b>
<b>Anlage 3, EB02</b> .....	<b>XXII</b>
<b>Anlagen 4, EB03</b> .....	<b>XXXV</b>
<b>Anlagen 5, EB04</b> .....	<b>LI</b>
<b>Anlagen 6, EB05</b> .....	<b>LXIII</b>
<b>Anlagen 7, EB06</b> .....	<b>LXXIX</b>
<b>Anlagen 8, EB07</b> .....	<b>XCIV</b>
<b>Anlagen 9, EB08</b> .....	<b>CIV</b>
<b>Anlagen 10, Fallzusammenfassungen</b> .....	<b>CXX</b>
<b>Anlagen 11, Codebuch</b> .....	<b>CXXIX</b>
<b>Anlagen 12, Summery Grid</b> .....	<b>CLVI</b>
<b>Selbstständigkeitserklärung</b> .....	<b>373</b>

# Abbildungsverzeichnis

ABBILDUNG 1 - BEISPIEL EINER LICHTSTEUERUNG ÜBER FUNKTIONSPLAN (FUP) PROGRAMMIERSPRACHE ...	1
ABBILDUNG 2 - ALTERNATIVEN DER SOFTWAREBEREITSTELLUNG .....	17
ABBILDUNG 3 - ABGEWANDELTES PHASENMODELL .....	19
ABBILDUNG 4 - MAGISCHES DREIECK SEQUENZIELLES PM.....	21
ABBILDUNG 5 - V-MODELL.....	24
ABBILDUNG 6 - VERGLEICH MAGISCHES DREIECK TRADITIONELLER & AGILER ANSATZ .....	26
ABBILDUNG 7 - VORGEHEN SCRUM MIT EINER ITERATION.....	28
ABBILDUNG 8 - KANBAN-BOARD .....	32
ABBILDUNG 9 - ABLAUFMODELL PROBLEMZENTRIERTES INTERVIEW .....	41
ABBILDUNG 10 - ABLAUFMODELL NACH KUCKARTZ .....	46
ABBILDUNG 11 - FORMEN EINFACHER & KOMPLEXER ANALYSE.....	50
ABBILDUNG 12 - ABLAUFMODELL DER ANALYSE.....	51
ABBILDUNG 13 - AUSSAGEN DER EXPERTEN ZUM VORHANDENSEIN VON ROLLEN IM PROJEKT .....	81
ABBILDUNG 14 - ABLAUFMODELL INITIALISIERUNGSPROJEKTPHASE .....	99
ABBILDUNG 15 - SCHEMATISCHE DARSTELLUNG SEQUENZIELL-PARALLELE SKALIERUNG.....	104

# Tabellenverzeichnis

TABELLE 1 - WICHTIGE STAKEHOLDER FÜR DEN BETRIEB VON NCLC-PLATTFORMEN .....	10
TABELLE 2 - KURZSTECKBRIEF INTERVIEWPARTNER .....	53
TABELLE 3 - LISTE DEDUKTIVE HAUPTKATEGORIEN .....	55
TABELLE 4 - CODESYSTEM INKLUSIVE SUBCODES .....	58
TABELLE 5 - ERGEBNISPRÄSENTATION: ROLLEN UND STAKEHOLDER IM PROJEKT .....	81
TABELLE 6 - MÖGLICHE KPIS ZUR ERFOLGSMESSUNG .....	89

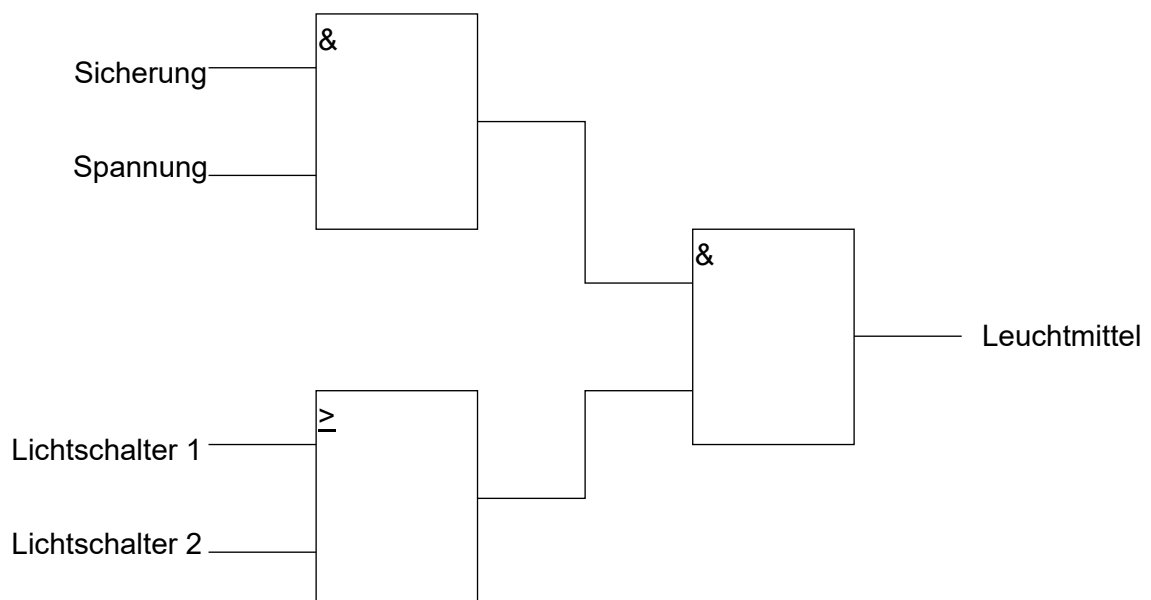


# Abkürzungsverzeichnis

<b>SPS</b>	Speicher Programmierbare Steuerung
<b>NCLC</b>	No Code/Low Code
<b>UI</b>	User Interface
<b>SaaS</b>	Software as a Service
<b>MVP</b>	Minimal Viable Product
<b>PM</b>	Projektmanagement
<b>CPM</b>	Critical Path Method
<b>EA</b>	Ende – Anfang Beziehung
<b>DoD</b>	Definition of Done
<b>WIP</b>	Work in Progress
<b>MA</b>	Mitarbeiter

# 1 Einleitung

Die Thematik der Vereinfachung von komplexen Themenbereichen ist ein großes Themenfeld. Zusätzlich zu dieser Vereinfachung entsteht der Bedarf für die Automatisierung von Prozessen. Dadurch sollten Prozesse schneller und einfacher von unterschiedlichen Personengruppen durchgeführt werden können oder komplett ohne zusätzliches Handeln erfolgen. Automatisierung beschreibt dabei die Gesamtheit verschiedener Steuerungs-, Regelungs- und Visualisierungsprozesse.<sup>1</sup> Eine Möglichkeit der Automatisierung für Maschinen basiert dabei auf der Programmiersprache Simatic S5 und S7 vom Hersteller Siemens. Diese Programmiersprache hat, im Gegensatz zu anderen Programmiersprachen wie HTML oder Python, eine Besonderheit, und zwar, dass sie grafisch realisiert werden kann.<sup>2</sup> Dies ermöglicht die Umsetzung komplexer logischer Verschaltungen zu einem komplexen Maschinen- und Anlagenablaufprogramm. Dafür werden lediglich Bausteine per Drag and Drop in der Programmoberfläche verschoben, welche bestimmte Funktionen erfüllen. Auf diese einfache Weise können Programme erstellt werden, ohne dabei auf Coding-Kenntnisse angewiesen zu sein. Dadurch können Facharbeiterberufe, wie der Elektroniker für Automatisierungstechnik, die eher auf das Instandhalten von elektronischen Komponenten ausgelegt sind, solche komplexen Programme selbst erstellen.<sup>3</sup> Dies ermöglicht Vorgehensweisen, wobei eine fachfremde Person eine Aufgabe aus einem anderen Fachbereich ausführt, mit einem geringeren Anteil an Befähigung für die Plattform, auf der gearbeitet wird. Ein Beispiel für ein Programm einer Speicher Programmierbaren Steuerung, kurz SPS ist in Abbildung 1 dargestellt:



**Abbildung 1 - Beispiel einer Lichtsteuerung über Funktionsplan (FUP) Programmiersprache**

<sup>1</sup> vg. We enreuther and Zastrow 2008, S. 2.

<sup>2</sup> vg. We enreuther and Zastrow 2008, S. 19.

<sup>3</sup> In der gesamten Arbeit wurde bei Personenbezeichnungen und personenbezogenen Hauptwörtern aus Gründen der besseren Lesbarkeit die männliche Form oder das generische Maskulinum verwendet. Es können dabei aber sowohl männliche, weibliche und diverse Personen gemeint sein.

Dies ist ein Anwendungsbeispiel für den Bedarf für ein niedrigschwelliges Angebot, um Programme umsetzen zu können, trotz mangelnder Programmierkenntnisse. Ähnliche Problemstellungen gibt es jedoch ebenfalls abseits der Maschinenprogrammierung: nämlich repetitive Workflows, welche manuell durch Mitarbeiter durchgeführt werden müssen und theoretisch automatisierbar wären.<sup>4</sup> Umfragen unter Unternehmen weisen auf eine steigende Nutzung von No-Code- und Low-Code-Plattformen hin. Diese Plattformen bieten ein niedrigschwelliges Angebot zur Automatisierung von Prozessen mit keinen oder wenig Programmierkenntnissen.<sup>5</sup> Die Entwicklung von Anwendungen, die es ermöglichen, solche Automatisierungen vorzunehmen, liegt originär im Aufgabenbereich der IT-Abteilung in Unternehmen. Die fortschreitende Digitalisierung und die steigende Aufgabenvielfalt der IT-Abteilungen sind Hürden, weshalb solche Entwicklungsanfragen oft nicht oder nur mit hoher zeitlicher Verzögerung erstellt werden, da andere Aufgaben eine höhere Priorität haben.<sup>6</sup> Verschärft wird diese Problemstellung dadurch, dass es einen erheblichen Fachkräftemangel im IT-Bereich gibt.<sup>7</sup>

Ein Lösungsansatz für diese Problemstellung ist der verstärkte Einsatz von No-Code- und Low-Code-Plattformen, um IT-Abteilungen zu entlasten und die Fachabteilungen zu befähigen, passende Lösungen selbst zu finden. Für Unternehmen stellen sich jedoch für die Planung und die Einführung einer Plattform verschiedene wichtige Fragen, wie diese möglichst effizient funktionieren kann. Für die Einführung von Unternehmensprozessen und neuer Unternehmenssoftware gibt es verschiedene Projektmanagementmodelle.<sup>8</sup> Die Fragestellung, welcher Ansatz der richtige für das eigene Unternehmen ist, sollte von Unternehmen genau abgewogen werden, da eine falsche Entscheidung zu weitreichenden Konsequenzen für ein Unternehmen führen kann. Wird das falsche Modell verfolgt, kann dies zu Kostensteigerungen oder erheblichen zeitlichen Verwerfungen führen. Aus diesem Grund ist es wichtig für Unternehmen, im Vorfeld eine Vorgehensweise zu wählen, die für das Unternehmen und die Plattform passt. Diese Überlegungen sollten sich ebenfalls auf die Auswahl der Plattform beziehen. Dazu sollten Unternehmen wissen, welche Auswirkungen No-Code-/Low-Code-Plattformen haben, um eine fundierte Entscheidung zu treffen und damit keine falschen Erwartungen entstehen. Dies kann einer Frustration und Ablehnung vorbeugen, die nach der Einführung entstehen können.

## 1.1 Forschungsfragen und Methodik

Das Ziel der vorliegenden Arbeit ist es, ein Vorgehensmodell zu evaluieren, welches Unternehmen bei der Einführung unterstützt und ihnen wichtiges Basiswissen zu den vorliegenden Themenfeldern bereitstellt. Zu diesem Zweck werden mithilfe von Experteninterviews folgende Forschungsfragen untersucht:

1. Welche Auswirkungen hat No Code/Low Code in deutschen Unternehmen?
2. Welches Vorgehensmodell eignet sich für die Einführung von No Code/Low Code in deutschen mittelständischen und großen Unternehmen?
  - a. Welche Voraussetzung muss ein Unternehmen mitbringen?
  - b. Welches Vorgehensmodell eignet sich am besten?
  - c. Welche Phasen, Rollen und Tools werden benötigt?

---

<sup>4</sup> vg . Schaffry 2022, S. 8.

<sup>5</sup> vg . ComputerWoche 2021; vg . Fre er 2022; vg . AP Ver ag 2022; vg . Bock and Frank 2021, S. 734.

<sup>6</sup> vg . techconsu t GmbH 2021, S. 5.

<sup>7</sup> vg . B tkom 2022.

<sup>8</sup> vg . Dechange 2020, S. 34 ff.

- d. Wie wird die Qualität sichergestellt und mit Änderungen umgegangen?
- e. Wie wird der Erfolg des Projekts gemessen?

Für die hier zu untersuchende Thematik sind Experteninterviews die geeignete Form der Evaluierung, um eine praxisrelevante Datengrundlage zu erhalten. Die Experten sind Projektleiter, die bereits an Einführungsprojekten teilgenommen haben und dadurch das Wissen erlangt haben, welche Methodiken sich als zielführend erwiesen haben und welche Problemstellungen beachtet werden müssen, um einen optimalen Projektverlauf zu gewährleisten. Diese praktischen Erfahrungen werden mithilfe der qualitativen Inhaltsanalyse strukturiert und die forschungsleitenden Informationen werden aus den Interviews extrahiert.<sup>9</sup> Durch einen standardisierten Leitfaden, der zum Zwecke der Interviewführung entwickelt wird, soll eine Vergleichbarkeit der Interviews hergestellt werden.

## 1.2 Aufbau der Arbeit

Die Kapitel 2 und 3 sollen für das Thema der Arbeit die theoretischen Grundlagen herstellen. Dabei wird in Kapitel 2 auf den Themenkomplex von No Code und Low Code eingegangen sowie bereits vorhandene wissenschaftliche Literatur aufgearbeitet. In diesem Kapitel soll aufgezeigt werden, wie die Funktionalität einer No-Code-/Low-Code-Plattform ist. Darüber hinaus erfolgt eine Vorstellung des zusammenhängenden Konzepts. Das Kapitel 3 beschäftigt sich mit dem zweiten Themenfeld der Arbeit. In diesem Kapitel werden die relevanten Projektmanagement-Modelle vorgestellt und die vorhandene Literatur aufgearbeitet. Dies dient der Klärung von Begriffen sowie der Vorstellung von in der Praxis eingesetzten Vorgehensmodellen und der Beantwortung der Frage, wie diese funktionieren. Es werden ebenfalls zu jedem Vorgehensmodellkonzept entsprechende beispielhafte Vorgehensmodelle im Detail vorgestellt.

Das Kapitel 4 erläutert die Evaluationsmethode für die Erhebung der Daten. Dabei wird auf die Relevanz des Experteninterviews sowie auf den Aufbau des Interviewleitfadens eingegangen. Abschließend wird kurz darauf eingegangen, wie die Daten weiterverarbeitet werden. Dieses Kapitel bietet einen fließenden Übergang in das Kapitel 5, wo die Methodik der qualitativen Inhaltsanalyse vorgestellt wird. Der Zweck des Kapitels ist es, ein Verständnis zu erzeugen, um die in Kapitel 6 durchgeführte Analyse nachvollziehen zu können und den theoretischen Hintergrund zu verstehen. In Kapitel 6 wird die Umsetzung der Analyse erläutert. Dazu werden ein Ablaufmodell der Methode und im weiteren Verlauf die Arbeit mit den transkribierten Interviews und den nachfolgenden Schritten der Kategorienbildung vorgestellt. Den Abschluss des Kapitels bildet das in Kapitel 7 vorgestellte finale Kategoriensystem mit den dazugehörigen Kodierungen. Die Ergebnisse der Analyse werden deskriptiv dargelegt und die Expertenmeinungen zu den verschiedenen Fragen des Leitfadens zusammengefasst. Dazu wird zuerst aufgezeigt, welche vorbereitenden Maßnahmen durch die Experten genannt wurden, danach, welche Erkenntnisse für das Vorgehensmodell gewonnen werden konnten.

In Kapitel 8 werden die Ergebnisse in Bezug auf die Forschungsfragen aufgearbeitet und diskutiert. Hier werden ebenfalls die Hürden und Beschränkungen, die im Verlauf der Arbeit entstanden sind, aufgezeigt und es wird ein Ausblick auf zukünftige Forschungsthemen gegeben. Im abschließenden Kapitel 9 wird zunächst ein Fazit für die Arbeit formuliert und anschließend wird eine Zusammenfassung der Arbeit vorgestellt.

---

<sup>9</sup> vg . Kuckartz and Räd ker 2022, S. 15.

## 2 Low Code

### 2.1 Das Problem des Fachkräftemangels

Der steigende Bedarf nach Informationstechnologien, im Zusammenhang mit den Digitalisierungsbemühungen der Firmen, führt in gleichem Maße zu einem steigenden Bedarf nach Fachkräften in diesem Bereich.<sup>10</sup> Dies führt zu einer erheblichen Problemstellung für deutsche Unternehmen, da es wegen unbesetzter Stellen zu Verlusten in den Bereichen Cybersecurity, Innovations- und Wettbewerbsfähigkeit kommt.<sup>11</sup> Diese Problemstellung zeigt sich unter anderem in Unternehmen und Unternehmensabteilungen, bei denen der Fokus auf der IT liegt. Daraus resultiert ein gesteigerter Bedarf für Unternehmen im Bereich der IT-Berufe. Darunter sind Berufe zu verstehen wie Data Analysts, Social-Media-Experten oder Cybersecurity-Experten.<sup>12</sup> Einer der Ursprünge für diese Problematik liegt vor allem darin, dass nicht ausreichend Nachwuchskräfte gewonnen werden können. Daraus resultierend entsteht die Situation für vorhandene Fachkräfte, dass sie eine große Auswahl an Optionen haben bzgl. der Wahl ihres Arbeitgebers. Im Kontext der Globalisierung werben immer mehr Arbeitgeber um diese Fachkräfte. Das führt zu einem sehr starken Wettbewerb, in dessen Verlauf es viele Unternehmen schwer haben, IT-Fachkräfte zu gewinnen. Die Entscheidung von IT-Fachkräften für ein Unternehmen erfolgt anhand verschiedener Kriterien, die sich nicht ausschließlich auf das Gehalt beziehen.<sup>13</sup>

Die Kultur eines Unternehmens ist ein wichtiges Kriterium zur Lösung dieses Problems. Dabei ist die Wahrnehmung, wie ein Unternehmen gesehen wird, nicht nur wichtig für die Akquirierung neuer Arbeitskräfte, sondern auch für das Halten und Weiterentwickeln der vorhandenen Mitarbeiter.<sup>14</sup> Eine gute Firmenkultur kann dazu führen, dass Mitarbeiter sich eher bereit erklären, bei Veränderungen im Unternehmen diese mitzutragen und sich auch selbst in neue Aufgabenfelder einzuarbeiten, durch Weiterbildung oder Ähnliches. Sind die Mitarbeiter in einem Unternehmen nicht dazu bereit, kann es schwierig werden für Unternehmen, sich an geänderte Markt- oder Kundenanforderungen anzupassen. Dies ist in besonderer Weise problematisch in sich schnell wandelnden Branchen und Wissensfeldern wie dem IT-Bereich.

Mit diesem Aspekt geht die Problematik des Up-or-out-Karrierewegs eng einher. Dies bedeutet, entweder man steigt auf oder man verlässt das Unternehmen.<sup>15</sup> Für deutsche Unternehmen kann dies bedeuten, dass ein Abfluss von Wissen stattfindet, und zwar in dem Moment, wo ein Mitarbeiter das Unternehmen verlässt. Dieser Mechanismus kann bei unvorbereiteten Unternehmen vor allem in der IT kritisch werden, da es, wie bereits erwähnt, immer schwieriger wird, die unbesetzten Stellen mit adäquaten Nachfolgern neu zu besetzen. Dieser Prozess kann vor allem durch fehlende Wertschätzung noch beschleunigt werden. Passiert innerhalb eines Unternehmens der Übergang zu mehr Digitalisierung, können Mitarbeiter abgehängt werden, die diesen Übergang nicht schnell genug vollziehen können. Die Folge: Wichtiges Wissen und motivierte Mitarbeiter verlassen das Unternehmen, da sie sich nicht wertgeschätzt fühlen. In diesem Zusammenhang erschließt sich als Zweites das

<sup>10</sup> vg. Btkom 2022; vg. Specht 2020; vg. Roth and He mann 2022, S. 14; vg. Bock and Frank 2021, S. 734.

<sup>11</sup> vg. Deutscher Industr e- und Hande skammertag 2021; Kön g 2020.

<sup>12</sup> vg. Rechste ner 2016, S. 1.

<sup>13</sup> vg. Schare ka 2020.

<sup>14</sup> vg. Rechste ner 2016, S. 4.

<sup>15</sup> vg. Rechste ner 2016, S. 5.

Problem, dass die Leistungsfähigkeit von Mitarbeitern sinken kann, was zu einer schrumpfenden Performance- und Innovationsfähigkeit führt.

Diese Abwanderung von Fachkräften führt zu einer immer stärker werdenden Überlastung der IT-Abteilungen durch die steigenden Anforderungen, welche durch die Digitalisierung verursacht werden. Der Mangel an Personal im Zusammenspiel mit immer größer werdenden Aufgaben-Backlogs für die IT stellt Unternehmen vor große Herausforderungen, für die dringend Lösungen gefunden werden müssen.

## 2.2 No Code/Low Code als Lösungsansatz

Um diesem Problem des Fachkräftemangels und der damit verbundenen Überforderung der IT-Abteilung entgegenzuwirken, wird verstärkt nach Lösungen gesucht. Ein Lösungsansatz, der sich in den letzten Jahren entwickelt hat, sind sogenannte No-Code- oder Low-Code-Plattformen, kurz NCLC-Plattformen.<sup>16</sup> Der Begriff beschreibt eine Art von Softwareumgebungen, mit dem Zweck, einerseits die Entwicklung von Software voranzubringen und andererseits einfache Nutzer ohne spezifischen IT-Background zu befähigen, eigene Programme zu erstellen.<sup>17</sup> Diese einfachen Nutzer werden auch als Citizen Developer bezeichnet und die Art der Befähigung als Citizen Development.<sup>18</sup> Die auf Grundlage einer NCLC-Plattform erstellten Anwendungen oder Applikationen werden durch den Citizen Developer erstellt. Diese Anwendungen werden weiterführend als NCLC-Anwendungen oder NCLC-Applikationen bezeichnet.

Bei NCLC-Plattformen soll die Entwicklung von Programmen und Apps, im Idealfall ohne Programmierkenntnisse, erfolgen können.<sup>19</sup> Dabei erfolgt die Entwicklung der Anwendung durch das Zusammensetzen von grafischen Bausteinen. Dieses Vorgehen wird als No-Code-Entwicklung bezeichnet. Im Gegensatz dazu werden im Falle von Low-Code-Entwicklungen geringe Programmierkenntnisse vorausgesetzt. Allerdings können diese Anwendungen auch für IT-Fachkräfte interessant sein. Durch NCLC-Plattformen werden diese befähigt, auf einfache Weise Arbeitsprozesse zu automatisieren und dadurch Anwendungen schneller entwickeln zu können. Die Art der Programme lässt sich in Erweiterung zu den Modell-getriebenen Programmen, oder Model Driven Programs, betrachten.<sup>20</sup> Die Art der Programmierung nutzt Modelle, um die Komplexität von Kodierung zu reduzieren. Das Unterscheidungskriterium zwischen den Modell-getriebenen Programmen und Low-Code-Programmen erfolgt dadurch, dass sich Low-Code-Applikationen in dem Punkt unterscheiden, dass die Beschränkungen größer sind als bei rein Modell-getriebenen Programmen. Das bedeutet, NCLC fokussiert sich auf ein konkretes Anwendungsszenario, weshalb die NCLC-Plattform und deren Applikationen an dieses Szenario angepasst werden müssen. No-Code- oder Low-Code-Anwendungen sind zum jetzigen Zeitpunkt verschiedenen Limitierungen unterworfen und aus diesem Grund nicht überall anwendbar. Bei der Verwendung von Künstlicher Intelligenz oder ähnlichen Anwendungsfällen, die eine sehr komplexe programmtechnische Vorarbeit benötigen, sind diese Plattformen noch nicht vollständig verwendbar. Es werden dafür tiefgreifende Kenntnisse von Coding oder Hilfestellungen durch Personen benötigt, welche diese Kenntnisse besitzen. Es existieren jedoch bereits Ansätze von Unternehmen, um dieses Problem zu lösen.

<sup>16</sup> vg . ComputerWoche 2021; vg . Fre er 2022; vg . AP Ver ag 2022; vg . Bock and Frank 2021, S. 734.

<sup>17</sup> vg . Bock and Frank 2021, S. 733.

<sup>18</sup> vg . Borgman and Hoogsteen 2022, S. 4720.

<sup>19</sup> vg . Detscher 2021, S. 508.

<sup>20</sup> vg . Benac and Mohd 2022, S. 527.

Zur Realisierung von NCLC-Anwendungen hat sich eine große Anzahl von Plattformen entwickelt. Diese Plattformen lassen sich als eine Sammlung von Werkzeugen zur Entwicklung von NCLC-Anwendungen definieren.<sup>21</sup> Dies ermöglicht Unternehmen eine schnelle Anpassung von Prozessen an die sich schnell verändernden Gegebenheiten in der Unternehmenswelt.

Eines der großen Beispiele für den Bedarf für solche Plattformen ist die Covid-Krise gewesen, die im Jahr 2019 begonnen hat.<sup>22</sup> In dieser Zeit mussten Unternehmen sich in kürzester Zeit an veränderte Problemstellungen und gesetzliche Anforderungen anpassen. Unternehmen mussten Prozesse für die Testung von Mitarbeitern im Unternehmen etablieren. Eine weitere Anforderung war die Archivierung von Daten, damit diese im weiteren zeitlichen Verlauf zur Überprüfung und Überwachung des Status herangezogen werden konnten. Damit sind Informationen gemeint wie zum Beispiel, ob ein Mitarbeiter geimpft, genesen oder getestet ist, auch bekannt als 3G-Regeln. Die zugehörigen Datenmengen mussten sortiert und archiviert werden. Vor allem bei mittleren und großen Unternehmen, wo eine Nachverfolgung via Stift und Zettel völlig unwirtschaftlich gewesen wäre, mussten schnell Lösungen gefunden werden. Für diesen und weitere ähnlich gelagerte Use Cases konnten Low-Code-Plattformen eine Lösung bieten, die schnell, effizient, auf das Unternehmen angepasst und kostengünstig entwickelt werden konnten. Am Beispiel der Stadt New York lässt sich dies aufzeigen: Die Stadt New York hatte ein Portal zum Nachverfolgen von Infizierten innerhalb von 3 Tagen entwickelt und eingesetzt, und das alles ohne größeren Programmieraufwand.

Wichtig ist in diesem Zusammenhang zu erwähnen, dass NCLC-Applikationen keine eigenständige Software sind, sie bauen stattdessen nur auf einer vorher durch Programmierer entwickelten Plattform auf, der NCLC-Plattform. NCLC-Anwendungen sind lediglich eine Erweiterung dieser Plattformen, wodurch Nicht-Entwickler befähigt werden, eigene Anwendungen zu erstellen. Diese Anwendungen sind jedoch den Regeln und Gesetzen der Plattform unterworfen, auf denen sie basieren. Bei einer schlecht entwickelten Plattform, welche zur Anwendungsentwicklung genutzt wird, wird wahrscheinlich auch eine schlechte Anwendung entstehen. Genauso wenig, wie ein schlechter Prozess zwingend durch Übertragung auf den NCLC-Bereich zu einem guten Prozess wird.<sup>23</sup>

## 2.2.1 Vor- und Nachteile von No Code/Low Code

Die Vorteile von NCLC-Plattformen sind vielfältig und werden von deutschen Unternehmen und der Wissenschaft wie folgt aufgeschlüsselt:

1. NCLC-Anwendungen zeichnen sich durch eine leichte Nutzung aus, da die Anwendungen nicht von Grund auf neu programmiert werden müssen, sondern auf einer vorhandenen Plattform aufbauen, welche die Tools zum Erstellen der Anwendungen zur Verfügung stellt.<sup>24</sup> Dadurch wird die Anwendungserstellung wesentlich vereinfacht, weil ein großer Teil der Vorarbeit für Anwendungen entfällt.
2. Ein weiterer Aspekt ist die Verkürzung der Entwicklungszeit von Fachanwendungen. Nutzer von NCLC-Plattformen können sich schnell und leicht an neue Vorgaben

---

<sup>21</sup> vg. Waszkowsk 2019, S. 374.

<sup>22</sup> vg. Woo 2020, S. 961.

<sup>23</sup> vg. Süddeutsche Zeitung 2015.

<sup>24</sup> .A. SAP 2022.

anpassen, indem sie eine passende Anwendung entwickeln oder eine vorhandene weiterentwickeln. In diesem Zusammenhang zeigt ein Paper von Forester auf, dass unter anderem die Entwicklungszeit von Administrationssystemen in Verbindung mit einem Web Channel mithilfe von NCLC von 2,7 Jahren auf 13 Wochen verkürzt werden kann.<sup>25</sup> Diese Zahlen stehen im Kontrast zu einer klassischen Entwicklung mittels Coding.

3. Die Automatisierung von Prozessen wird einhergehend mit der Beschleunigung der Entwicklung, die in Punkt 2 beschrieben wird, gleichzeitig erhöht. Dies wird unter anderem dadurch erreicht, dass viele der Anwendungen eine Vorschlagfunktion besitzen. Diese Vorschlagfunktionen werden mittels KI-Anwendungen realisiert und zeichnen sich durch verbesserte Vorschläge aus.<sup>26</sup>
4. Die Senkung von Entwicklungszeiten und die Beschleunigung von Prozessen führen häufig dazu, dass es zu einer Kostensenkung kommt. Dies begründet sich in der Senkung von Personalkosten, da die Mitarbeiter mehr Zeit für andere Aufgaben haben und diese erfüllen können. Als Aspekt, der zur Kostensenkung beiträgt, ist die Verbesserung von Prozessen zu nennen und dadurch verbunden das schnellere Erfüllen von Aufgaben. Die Senkung der Personalkosten ergibt sich daraus, dass bei gleichbleibenden Personalkosten mehr Aufgaben erledigt werden.
5. Die Fähigkeit, innerhalb eines Prozesses Daten zu sammeln, zu teilen, zu verarbeiten und zu speichern eröffnet die Möglichkeit, diese Daten effektiver zu integrieren und flexibler zu verwenden. Dadurch können Prozesse effektiver optimiert und angepasst werden. NCLC-Plattformen ermöglichen es dem User, dies zu realisieren, und das führt zu besseren und sichereren Entscheidungen für den Prozessverlauf.
6. Die verbesserte Flexibilität wurde bereits in Kapitel 2.2 mit dem Beispiel des New Yorker Corona-Portals hervorgehoben. Diese Flexibilität lässt sich innerhalb des Entwicklungsprozesses von Anwendungen ebenfalls bemerken. Durch die einfache Bedienung der NCLC-Plattformen können verschiedene neue Ideen schnell getestet und, daraus resultierend, gezielte Verbesserungen vorgenommen oder nicht zielführende Ideen verworfen werden. NCLC-Plattformen ermöglichen es, durch diesen Vorteil Ideen zu verfolgen, die in einem ersten Schritt auf einer Metaebene definiert werden und in einem späteren Prozessabschnitt konkretisiert für die Fachabteilung oder den Nutzer erstellt werden.<sup>27</sup>
7. Die schnelle Entwicklungszeit und die niedrigen Kosten können förderlich für die Customer Relationship sein. Das ist wichtig, da eine enge Kundenbindung für Unternehmen vielleicht eine der wichtigsten Kennzahlen ist, auf die geachtet werden muss.
8. Zwei der großen Probleme der heutigen Zeit sind, wie bereits angesprochen, die Datensicherheit und der Datenschutz. In Bezug auf Unternehmen, welche Daten verarbeiten, die zu wichtig sind oder aus regulatorischen Gründen nicht auf Servern außerhalb des Unternehmens gespeichert werden dürfen, können NCLC-Plattformen helfen, das Risiko zu senken. Dabei wird dem Unternehmen trotzdem ermöglicht, mit den Daten arbeiten zu können. Wichtig hervorzuheben ist dabei, dass dies kein primärer Zweck von NCLC-Plattformen und -Anwendungen ist und diese keinen Cybersecurity-Beauftragten ersetzen können.

---

<sup>25</sup> vg. Richardson and Rymer 2016, S. 2.

<sup>26</sup> vg. Bock and Frank 2021, S. 738.

<sup>27</sup> vg. Taesra and G. S. 2021, S. 347.



9. Durch die Überlastung von vorhandenen IT-Abteilungen, sei es durch den Fachkräftemangel oder aufgrund der Aufgabenflut, welche IT-Abteilungen im Zuge der Digitalisierung erfahren, bleiben viele ausstehende Aufgaben auf der Strecke. Die Befähigung von Nicht-IT-Mitarbeitern, Lösungen für sich selbst zu bauen, unterstützt, dass diese ausstehenden Aufgaben schneller bearbeitet werden können. Dies kann sich auch positiv auf die Zufriedenheit der Mitarbeiter auswirken. Das wird dadurch erreicht, dass Aufgaben reduziert werden und damit der Druck von der IT-Abteilung weggenommen wird und für die Nicht-IT-Mitarbeiter keine langen Wartezeiten für ihre Problemlösungen mehr entstehen.
10. Governance-Risiken sind ein Problem, mit dem viele Unternehmen zu kämpfen haben.<sup>28</sup> Der Begriff Governance beschreibt das Regelwerk, wie mit der Plattform interagiert wird und wie diese nach den Regularien des Unternehmens verwendet werden darf.<sup>29</sup> Dieses Problem wird auch als Schatten-IT oder Wildwuchs bezeichnet und definiert den Zustand im Unternehmen, dass nicht genehmigte Programme sowie Applikationen verwendet werden. Dieses Problem kann entstehen, wenn Mitarbeiter versuchen, ein vorhandenes Problem zu lösen, indem sie sich ein Programm herunterladen, das nicht durch die IT genehmigt wurde. Ein weiteres Problem ist, dass von Datenbanken, welche die gleichen Daten enthalten, verschiedene Versionen existieren, weil Mitarbeiter diese unkontrolliert abspeichern und zur Erledigung ihrer Aufgaben verteilen. Es entsteht ein Verlust der Kontrolle darüber, welche Programme verwendet werden, wo Datenbanken abgespeichert sind und wo aktuelle Daten liegen. Dadurch können unter anderem Sicherheitsrisiken oder Lizenzprobleme entstehen. NCLC-Plattformen können nicht nur dadurch helfen, dass sie die Notwendigkeit für solche Lösungen für die Mitarbeiter entfernen, sondern es bei vielen Plattformen möglich ist, bestimmte Regeln für die Entwicklung von Anwendungen festzulegen. In diesem Zusammenhang können ebenfalls Einschränkungen vorgenommen werden: Wer darf was an der Applikation verändern bzw. nutzen. Die Einführung von NCLC ist als ein gesamtheitliches Konzept zu verstehen, bei dem nicht nur eine neue Software eingeführt wird, sondern darüber hinaus neue Prozesse in Zusammenarbeit mit den Fachmitarbeitern implementiert werden. Dadurch wird weiterführend eine Verbesserung der Sensibilisierung erreicht.
11. Als ein weiterer Vorteil für Nutzer kann hier als Möglichkeit genannt werden, plattformübergreifend auf die verschiedenen NCLC-Anwendungen zuzugreifen.<sup>30</sup> Dies ist abhängig von der gewählten Plattform, jedoch bietet eine große Anzahl von NCLC-Plattformen diese Funktionalität an. Mit plattformübergreifender Nutzung ist die Fähigkeit gemeint, die Applikationen geräteunabhängig verwenden zu können. Eine Integration von verschiedenen Programmen, die nicht Teil der Plattform sind, ist oftmals möglich. Daten können zum Beispiel aus einer Exceltabelle durch die NCLC-Applikation extrahiert und weiterverarbeitet werden, durch verschiedene Schnittstellen zwischen den Programmen.

Die hier genannten Vorteile sind nicht ohne Nachteile zu verstehen.<sup>31</sup> Diese Nachteile ergeben sich zum Teil direkt aus den Vorteilen, wenn während der Implementierung und Anwendung falsch vorgegangen wird. Dadurch können die Risiken in ähnlichem Maße steigen, weshalb verschiedene Dinge besonders beachtet werden sollten.

---

<sup>28</sup> vg. Kopper and Westner 2016, S. 1687.

<sup>29</sup> vg. Le me ster 2021, S. 9.

<sup>30</sup> vg. Ta esra and G. S. 2021, S. 347.

<sup>31</sup> vg. B ge ow 2020.

## Nachteile von NCLC-Plattformen

1. Wie im Punkt 4 der Vorteile bereits dargelegt, ist die Qualität der NCLC-Anwendung in hohem Maße davon abhängig, wie die Qualität der zugrundeliegenden Plattform ist. Angesichts dessen ist bei der Auswahl der Plattform darauf zu achten, dass diese auch zum Unternehmen und den dort vorhandenen Prozessen und der eventuell bereits vorhandenen Software passt. Eine NCLC-Plattform, die für ein Unternehmen die perfekte Lösung ist, muss nicht zwingend in gleichem Maße eine perfekte Lösung für ein anderes Unternehmen sein. Aus diesem Grund sollte bei der Auswahl der Plattform bedacht vorgegangen werden.
2. Eine Fehlannahme, die sich aus der Anwendung von NCLC-Anwendungen ergeben kann, kann sein, dass eine funktionierende IT-Abteilung gar nicht oder nur teilweise benötigt wird, da die Anwendung Aufgaben aus diesen Abteilungen herauszieht und die Fachmitarbeiter befähigt, bestimmte Anwendungen selbst zu erstellen. Dies führt allerdings zu Security- und Compliance-Risiken. Der Anwender der NCLC-Plattform kennt den zugrundeliegenden Code meist nicht und selbst wenn er diesen kennen würde, versteht er ihn nicht. Um dem entgegenzuwirken, ist eine funktionierende IT-Abteilung nötig. Diese überwacht diese Risiken und steuert der Entwicklung entgegen, um Schaden vom Unternehmen abzuhalten.
3. Eine funktionierende IT-Abteilung ist ebenfalls nötig, um eine unternehmensweite Governance einzuhalten. Durch die einfache Anwendungserstellung kann es schnell passieren, dass der Vorteil, dass die Governance verbessert wird, sich wieder negiert und zum Gegenteil umkehrt. Ohne klare Regeln, nach denen NCLC-Anwendungen erstellt werden, kann es zu einer neuen Schatten-IT kommen. Dieser Fall kann eintreten, sollten Anwender anfangen, ohne vorher festgelegte Regeln Anwendungen zu entwickeln und das ohne Rücksicht auf im Unternehmen vorhandene Abläufe oder zugrundeliegende Vorgaben.<sup>32</sup>
4. Vor allem bei komplexen Anwendungen können NCLC-Plattformen an ihre Grenzen stoßen, weil ihr Zweck lediglich darauf ausgelegt ist, einfache Prozesse und Workflows für Fachabteilungen zu optimieren und zu automatisieren. Wird eine Anwendung zu komplex oder bietet die NCLC-Plattform für bestimmte, im Unternehmen verwendete Software, keine entsprechende Verbindung für Kollaboration, braucht es einen erfahrenen Entwickler mit Programmierkenntnissen, um diese Lücke zu schließen.<sup>33</sup> Aus diesem Umstand ergibt sich ein Bedarf an geschultem IT-Personal, welches hier unterstützen oder diese komplexen Aufgaben realisieren kann.
5. Die Verwendung einer NCLC-Plattform kann zu dem Problem führen, dass ein Unternehmen sich in eine starke Abhängigkeit zu dem Unternehmen, welches die Plattform bereitstellt, begibt. Diese Situation wird als Vendor lock-in bezeichnet.<sup>34</sup> Diese Situation kann zwar durch eine Diversifizierung mit verschiedenen Plattformen gelöst werden, wirft allerdings im Gegenzug das Problem höherer Kosten auf, da mehr Anwender und IT-Personal auf verschiedenen Plattformen befähigt werden müssen. Dabei sind nicht nur höhere Kosten zu berücksichtigen, sondern in gleichem Maße gilt dies für die Motivation der Mitarbeiter, sich in diese unterschiedlichen Plattformen einzuarbeiten, denn primär sind die Mitarbeiter der Fachabteilung nicht dafür

---

<sup>32</sup> vg. techconsult GmbH 2021, S. 15.

<sup>33</sup> vg. Taesra and G. S. 2021, S. 347.

<sup>34</sup> vg. Bgeow 2020; vg. Tay 2021.

eingestellt worden, weshalb dies statt zu besseren Prozessen zu schlechteren Prozessen und Arbeitsabläufen führen kann.

6. Mitarbeiterqualifizierung ist ein großes Thema, das sich zu einem großen Nachteil der Plattformen entwickeln kann. Wie bereits erwähnt, werden bei NCLC Fachmitarbeiter befähigt, eigene kleine Anwendungen zu erstellen. Dafür müssen die entsprechenden Mitarbeiter aber qualifiziert und geschult werden. Für eine erfolgreiche Umsetzung ist es wichtig, dass die entsprechenden Mitarbeiter dazu bereit sind, sich zu informieren und sich immer weiter zu qualifizieren, weil NCLC-Plattformen nicht statisch in ihrer Entwicklung bleiben, sondern sich regelmäßig weiterentwickeln. Sind die Mitarbeiter nicht dazu bereit, sich weiterzubilden, können die Einführung und die Anwendung einer NCLC-Plattform daran scheitern. Dies kann zu einem erhöhten Kostenfaktor werden, der nicht unterschätzt werden darf.<sup>35</sup> Im gleichen Maße sind Unternehmen in der Pflicht, ihren Mitarbeitern diese Befähigung zu ermöglichen.

## 2.2.2 Anwendung von No Code/Low Code

Unter Beachtung der bekannten Vor- und Nachteile ergeben sich verschiedene Business-Anwendungsszenarien, im Bereich von Enterprise-Ressource-Planning, Customer-Relationship-Management, Supply-Chain-Management, zur Analyse vorhandener Daten, Human-Ressource-Management, Kundenportale etc. Die sich stetig ändernden Anforderungen an diese Bereiche decken sich mit den Vorteilen von NCLC und sind deshalb als Anwendungsgebiete geeignet.

Zur erfolgreichen Anwendung von NCLC in Unternehmen lassen sich verschiedene Stakeholder identifizieren, die in 5 essenziellen Rollen zusammengefasst werden können, die Krejci, Satu und Missonier in ihrer Studie herausstellen.<sup>36</sup>

**Tabelle 1 - Wichtige Stakeholder für den Betrieb von NCLC-Plattformen**

Fachmitarbeiter/ Citizen Developer	Die Entwicklung der NCLC-Anwendung wird besonders getrieben von den Fachmitarbeitern selbst. Im Gegensatz zum traditionellen Prozess der Unternehmensentwicklung, wo vor allem die Innovation von den oberen Hierarchieebenen vorangetrieben wird, ist dies im Falle von NCLC teilweise umgekehrt. NCLC-Plattformen befähigen Fachmitarbeiter, einen aktiven Anteil im Entwicklungsprozess einzunehmen, weshalb eine gesteigerte Motivation gemessen wurde.
IT-Entwickler	Professionelle Entwickler werden trotz der Möglichkeiten von NCLC-Plattformen immer noch gebraucht. Auf der einen Seite als technischer Ansprechpartner für die Fachmitarbeiter, die NCLC-Anwendungen erstellen, da diesen der IT-Hintergrund fehlt und die Anwendungen, wie bereits vorher festgestellt, gewissen Limitationen unterliegen. Auf der anderen Seite sind sie enorm wichtig, wenn diese Anwendungen dann in die vorhandenen Produktivsysteme integriert werden sollen, und zur Einhaltung von Sicherheitskriterien und Governance.

<sup>35</sup> vg. techconsult GmbH 2021, S. 15.

<sup>36</sup> vg. Krejci, Iho, and Missonier 2021, S. 6.

NCLC-Hersteller	Diese Rolle ist besonders wichtig in der Zeit der Einführung der NCLC-Plattform und der Integration in das Unternehmen. Der Hersteller übernimmt eine wichtige Rolle dabei, den Fachmitarbeiter im Umgang mit der Plattform zu unterweisen. Dasselbe gilt bei der Kommunikation mit der IT, wenn diese Anwendungen dann in die vorhandene Unternehmens-IT überführt werden.
End-User	Die zukünftigen Endnutzer, oder End-User, einer der entwickelten Anwendungen sind in gleichem Maße wichtige Stakeholder wie die Fachmitarbeiter, welche die Anwendungen entwickeln sollen. Diese Nutzer werden oftmals bei Implementierungsprozessen vernachlässigt, obwohl diese frühzeitig wertvolles Feedback geben können, damit die fertige Anwendung schnell und leicht von ihnen nach der Entwicklung und Integration genutzt werden kann.
Geschäftsleiter/ Geschäftsführung/ Führungspersönlichkeiten	Bis zu einem gewissen Grad übernehmen Führungspersönlichkeiten die Rolle von Sponsoren, insbesondere indem sie das Team dazu ermutigen, Ideen auf der Plattform zu entwickeln, den Fortschritt zu verfolgen, positiven Input zu geben, das Team zu unterstützen und die resultierenden NCLC-Anwendungen bei den Fachabteilungen, die diese anwenden sollen, sowie bei weiteren Mitarbeitern zu bewerben.

Mithilfe dieser Rollen können NCLC-Plattformen und -Applikationen in vielen verschiedenen Bereichen eingesetzt werden. Drei wichtige Bestandteile von NCLC-Applikationen, welche den Einsatzzweck beeinflussen, sind: der Ersteller der Applikationen, die Server und die Funktionalitäten, die durch die Plattformen bereitgestellt werden.<sup>37</sup>

Beispielhaft sollen einige dieser Anwendungsfelder benannt werden, um die Einsatzmöglichkeiten von NCLC aufzuzeigen.

- Hinsichtlich der Kundenkommunikation wird es für Unternehmen immer wichtiger, rund um die Uhr erreichbar zu sein. Finanziell kann so etwas zu einer starken Steigerung der Kosten führen und es ist deshalb wenig sinnvoll, dafür eine Person die komplette Zeit zu beschäftigen. Um dieses Problem zu lösen, werden NCLC-Plattformen vermehrt im E-Commerce und in den Kundendienstumgebungen für Chatbots genutzt.<sup>38</sup> NCLC erleichtert die Verwendung und den Zugang zu solchen Programmen. Der technische Aufwand, die Entwicklung und die Instandhaltung sind für Firmen wesentlich geringer als bei traditionellen Systemen aus Eigenentwicklung.
- Als ein weiteres Anwendungsfeld von NCLC tritt der Bereich Augmented Machine Learning auf.<sup>39</sup> Der Bedarf für Firmen, Daten mithilfe von maschinellem Lernen auszuwerten und weiterzuverarbeiten, ist spätestens durch viele Digitalisierungsanwendungen gegeben. Die zugrundeliegende Problemstellung dabei ist hier, dass viele Unternehmen weder die Fachkräfte haben, solche Modelle für ihre Zwecke zu entwickeln, noch die Möglichkeit besitzen, diese Modelle auf dem neuesten Stand zu halten. NCLC-Plattformen können dabei helfen, da sie die Anwender dazu

<sup>37</sup> vg. Taesra and G. S. 2021, S. 347.

<sup>38</sup> vg. Benac and Mohd 2022, S. 527.

<sup>39</sup> vg. Benac and Mohd 2022, S. 528.

befähigen, ihre eigenen Modelle zu trainieren, und sie die Hürden herabsenken, solche Programme einsatzfähig zu machen und zu erhalten.

- In vielen Unternehmen werden Daten unterschiedlich ausgewertet. Für viele Anwender oder Citizen Developer kann es von Vorteil sein, die Anwendung als sogenannte Canvas-App zu erstellen.<sup>40</sup> Dies führt zu einer Verbesserung des User Interfaces, kurz UI, sowie der Übersichtlichkeit der Handhabung des Programms. Dies ist verbunden mit den NCLC-Plattformen, die im Bereich Mobile-Apps aktiv sind. NCLC sorgen dafür, dass der Bedarf für weitreichende Programmierfähigkeiten sinkt, um solche Apps zu erstellen. Dies ist unabhängig davon, ob die Apps im Unternehmen und innerhalb einer Abteilung eingesetzt werden oder für Kunden außerhalb des Unternehmens zur Verfügung stehen sollen. Vor allem in Verbindung mit anderen im Unternehmen genutzten Programmen können so erfolgreich Geschäftsprozesse automatisiert werden, ohne dabei die Usability der Anwendung zu verlieren, weil sie zu kompliziert für Endanwender ist, um sie zu nutzen.

Diese Use Cases sollen exemplarisch einige Felder aufzeigen, in denen NCLC bereits erfolgreich genutzt wird. Es gibt jedoch eine noch viel größere Auswahl an Möglichkeiten, welche aufzuzählen den Rahmen dieser Arbeit überschreiten würde.

## 2.3 Aufstieg von Citizen Development

Wie in Kapitel 2.1 herausgestellt, ist ein starker Bedarf für Fachkräfte vorhanden. Dies wird verursacht durch den steigenden Bedarf für Digitalisierung in Unternehmen und ein gesteigertes Aufgabenaufkommen in den IT-Abteilungen vieler Unternehmen. Ein Bestandteil dieses Mangels äußert sich darin, dass unternehmensinterne Entwicklungen einen erhöhten zeitlichen Bedarf für ihre Fertigstellung haben.<sup>41</sup> No Code/Low Code kann in diesem Zusammenhang diesem Mangel entgegenwirken, indem es IT-Personal durch sehr einfache Tools dazu befähigt, schnell neue Entwicklungen voranzutreiben auf Basis einer vorhandenen No-Code-/Low-Code-Plattform, wie dies in New York geschehen ist am Anfang der Corona Krise.<sup>42</sup> Der Vorteil hierbei ist, dass der IT-Entwickler ein Programm nicht von Grund auf neu programmieren muss, sondern bereits auf einer vorhandenen Basis aufbauen kann. Dies kann zu einer Verbesserung der Start-to-release-Zeit führen.

Der zweite Lösungsansatz hinsichtlich dieses Mangels ist es, den riesigen Aufgaben-Backlog der IT-Abteilungen zu entlasten und diese Arbeiten in anderen Bereichen zu verteilen.<sup>43</sup> Die Befähigung von Fachkräften, sich an der Entwicklung von Unternehmens- und Geschäftssoftware zu beteiligen, wird als Citizen Development bezeichnet.<sup>44</sup> Diese Befähigung von Nicht-IT Personal ist eine Stärke von NCLC-Plattformen. Das Konzept ist eine Erweiterung vom Konzept der Endnutzer-Entwicklung. Der Unterschied zwischen diesen beiden Konzepten ist, dass die Endnutzer-Entwicklung sich auf ein spezielles Softwareartefakt fokussiert, wohingegen der Citizen Developer sich darauf fokussiert, nicht nur Lösungen für eine Person zu entwickeln, sondern Lösungen für eine Gruppe von Menschen zu entwickeln und diesen die entwickelten Applikationen zur Verfügung zu stellen.

---

<sup>40</sup> vg. Benac and Mohd 2022, S. 529–531.

<sup>41</sup> vg. G mann 2021; vg. Specht 2021; vg. Institute der deutschen Wirtschaft 2020.

<sup>42</sup> vg. Woo 2020, S. 961.

<sup>43</sup> vg. Sanchez et al. 2019, S. 4.

<sup>44</sup> vg. Borgman 2022, S. 4717.

Diese Befähigung von Citizen Developern hat verschiedene Auswirkungen auf Organisationen und Unternehmen. Zunächst verspricht man sich, wie bereits erwähnt, den Fachkräftemangel zu lösen. Ein weiterer Aspekt, der mit der Einführung dieses Konzeptes erreicht werden soll, ist eine Steigerung der Nutzung von Digitalisierungspotenzialen und den damit verbundenen Vorteilen für Unternehmen.

Die fächerübergreifende Befähigung von Non-IT-Personal, eigene IT-Anwendungen hervorzubringingen, führt dazu, dass Anforderungen an Anwendungen wesentlich spezifischer entwickelt werden können. Durch das detaillierte Wissen in Bezug auf die vorhandenen fachspezifischen Prozesse in einer Organisationseinheit oder Fachabteilung können Anwendungen aufgabenbezogen entwickelt werden.<sup>45</sup> Dieses Wissen wird durch die Citizen Developer eingebracht. Dadurch soll das Ziel erreicht werden, dass Fehler vermieden werden, die durch mangelhaftes Verständnis des Prozesses oder eine fehlerhafte Kommunikation entstehen. Dies führt zu einer zeitlichen Effizienzsteigerung bei der Entwicklung und der Implementierung von Anpassungen an der Anwendung sowie zu einer Entlastung der IT durch das Auslagern von Aufgaben.<sup>46</sup>

Wie bereits bezüglich der generellen Vorteile von NCLC-Plattformen formuliert, kann es zu einer Steigerung der Mitarbeitermotivation und der Produktivität kommen. Die Anpassung von Applikationen auf die direkten Bedürfnisse von Fachpersonal führt in der Folge zu einem verbesserten Workflow. Citizen Development ist die Grundlage, um diese positiven Aspekte der NCLC-Plattformen erreichen zu können. Sollte es zu einer Einführung einer NCLC-Plattform kommen, die nicht durch die Fachanwender angenommen wird, kann dies ein Punkt sein, der zum Scheitern des Projekts führt. Daher wird dieser Aspekt hier gesondert herausgestellt. Bei einer unzureichenden Implementierung von Citizen Development können verschiedene unerwünschte Aspekte hervortreten, die zu beachten sind. Unter anderem kann nach der Einführung einer NCLC-Plattform bei mangelhafter Implementierung des Citizen-Development-Konzepts eine Schatten-IT entstehen. Als Schatten-IT wird eine IT-Struktur bezeichnet, die sich der Kontrolle der regulären IT-Abteilung immer weiter entzieht.<sup>47</sup> Wenn die NCLC-Plattform zwar freigegeben wurde, die Mitarbeiter aber nicht befähigt wurden, sie zu nutzen, und nicht mit der im Unternehmen herrschenden Governance für solche Anwendungen vertraut gemacht wurden, kann dies zu erheblichen Risiken führen. Diese Risiken unterscheiden sich je nach Anwendungsgebiet der Anwendung.<sup>48</sup> Eine weitere unerwünschte Folge von Schatten-IT kann sein, dass Sicherheitsrisiken für Unternehmen entstehen. Dies kann geschehen, wenn Anwendungen erstellt werden ohne Einhaltung von Governance-Kriterien und Richtlinien zur IT-Security. Dieser Mangel an Rahmenbedingungen kann für Unternehmen Sicherheitsrisiken eröffnen.

Ein weiterer Aspekt ist, dass es zu Problemen in den Prozessen führen kann, welche die Anwendung eigentlich verbessern sollten. Ursächlich dafür können eine fehlerhafte Implementierung in der IT-Unternehmensumgebung oder fehlerhaft erstellte Anwendungen sein. Diese Situation ist besonders kritisch, wenn diese Anwendungen für den Kunden zugänglich sind und so die Unternehmensreputation beschädigt wird. Das kann zu einem noch größeren Schaden für Unternehmen führen. Eine weitere Auswirkung einer schlechten Governance kann sein, dass es zu einem gesteigerten Mehraufwand der IT-Abteilung kommt, die ursprünglich entlastet werden sollte.

Citizen Developer sind im Ursprung Fachmitarbeiter und keine IT-Mitarbeiter, weshalb die Anforderungen an die Citizen Developer in einem beherrschbaren Maß gehalten werden

---

<sup>45</sup> vg. Borgman 2022, S. 4717.

<sup>46</sup> vg. Paternò and Wu f 2017, S. 51.

<sup>47</sup> vg. Zimmermann and Rentrop 2012, S. 288.

<sup>48</sup> vg. Microsoft 2022; vg. Augsten 2019.

sollten. Das bedeutet, um Governance-Risiken zu verringern, müssen Mitarbeiter auf den vom Unternehmen gewählten Plattformen befähigt werden.<sup>49</sup> Infolge dieser Befähigung können Governance-Risiken verringert werden, da die Mitarbeiter Plattformen und Anwendungen gemäß der Vorgabe durch die IT-Abteilung nutzen. Dies kann dazu führen, dass Kostensteigerungen vermieden werden können, die entstehen, wenn die IT-Abteilung sich plötzlich neben ihren normalen Aufgaben noch um die Problemlösung für die fehlerhaften Applikationen von Citizen Developern kümmern muss.

Das Citizen Development wird eine IT-Abteilung nicht ersetzen können. Um komplexe Anwendungen im Unternehmen zu realisieren, wird die IT-Abteilung weiter benötigt. Primär haben Citizen Developer die Aufgabe, durch ihr Fachwissen bestimmte Applikationen für ihre Fachabteilung zu entwickeln und in diesen Fachbereichen auf diese Weise die Prozesse zu verbessern. Sie sind dabei allerdings an die Fähigkeiten und Limitationen der jeweiligen Plattform gebunden, welche Grundlage der Applikation ist. Wenn Anwendungswünsche zu komplex werden oder die Befähigung der Citizen Developer unzureichend ist, kann dies zu erheblichen Verwerfungen in Unternehmen führen.

Ein weiterer wichtiger Aspekt, der für den Erfolg von Citizen Development eine Rolle spielt, ist ein Wandel in der Kultur vieler Unternehmen.<sup>50</sup> Die Einführung von Citizen Development ist ein Bestandteil eines übergeordneten Prozesses, den Unternehmen durchlaufen sollten, wenn Risiken für ein Scheitern des Projekts reduziert werden sollen. Vertrauen gegenüber den Citizen Developern muss geschaffen werden und die IT-Abteilung muss in diesen Bereichen von einem Entwickler von Lösungen zu einem Unterstützer für die Lösungsentwickler werden. Es müssen Schnittstellen zwischen den Citizen Developern und den IT-Abteilungen implementiert werden, um ein kollaboratives Arbeiten zwischen diesen zu ermöglichen. Sollen die Vorteile des Citizen Developments als Faktor im Unternehmen wirksam werden, sollte jedoch ein Umdenken bei der Unternehmensleitung und den Vorgesetzten vorhanden sein. Entsteht kein Vertrauen gegenüber den Citizen Developern und wird ein zu starker Druck aufgebaut, kann dies eine Ursache für das Scheitern einer erfolgreichen Implementierung von NCLC sein. Es besteht die Gefahr, dass Citizen Developer sich abwenden, weil sie sich in dieser Rolle unwohl fühlen, oder dass es bei mangelndem Vertrauen zu einer erheblichen Kostensteigerung kommt. Dies kann vordringlich passieren, wenn ein Unternehmen jeden Schritt der Citizen Developer kontrollieren will und dadurch ein erhöhter Ressourcenbedarf für die Überwachung entsteht.

## 2.4 Einordnung von No Code/Low Code im Softwareumfeld

### 2.4.1 Formen von Software im Unternehmen

In Unternehmen werden verschiedene Formen von Software eingesetzt. Zu Beginn der betrieblichen IT-Nutzung wurde dabei hauptsächlich auf selbst entwickelte Systeme zurückgegriffen.<sup>51</sup> Diese Software wird auch als Individual Software bezeichnet, da sie auf die individuellen Bedürfnisse eines Unternehmens spezifisch angepasst werden kann. Diese Form von Softwareentwicklung kann durch externe Dienstleister ebenfalls realisiert werden und wird in gleichem Maße an die Bedürfnisse des Unternehmens angepasst, wie das bei

---

<sup>49</sup> vg. V o no 2021.

<sup>50</sup> vg. V o no 2021; vg. Rentrop 2019.

<sup>51</sup> vg. Krcmar 2015, S. 211.

Eigenentwicklungen geschehen würde. Heute wird diese Art der Softwareentwicklung meist genutzt, wenn es für das Unternehmen keine passende Standardlösung gibt. Durch den Einsatz von Individual Software können Anwendungsfälle und Use Cases realisiert werden, die ein sehr unternehmensspezifisches Profil besitzen und sich nicht so effizient durch alternative Softwarelösungen realisieren lassen wie Standardsoftware. Diese Steigerung des Detaillierungsgrades der Software wirkt sich in den meisten Fällen jedoch negativ auf Entwicklungskosten zum Erstellen und Betreiben der Lösung aus. Es wird ein erhöhter Aufwand nötig für den Betrieb und die Instandhaltung der Software, zum Beispiel mit regelmäßigen Updates.

Der Individualsoftware steht die Standardsoftware gegenüber. Diese Software ist nicht spezifisch auf das Unternehmen zugeschnitten und wird für viele mögliche Use Cases zur Verfügung gestellt. Sie kann in bestimmten Maßen allerdings auf die Bedürfnisse von Unternehmen angepasst werden. Diese Anpassungen besitzen jedoch meist nicht den gleichen Detailgrad, wie es bei Individualsoftware der Fall ist.<sup>52</sup> Aufgrund der Standardisierung der Software ist diese wesentlich kostengünstiger und eignet sich deshalb für eine große Bandbreite von Unternehmen wie kleinere, mittlere und große Unternehmen. Bei Standardsoftware ist ein Kauf nicht zwingend erforderlich, sondern diese kann gemietet werden, was sich positiv auf die Kosten auswirken kann. Diese Art des Software-Erwerbs heißt Software as a Service, kurz SaaS. Ein weiterer Aspekt, der für Standardsoftware spricht, ist, dass diese in den meisten Fällen sofort verfügbar ist, im Gegensatz zu einer Eigen- oder Individualentwicklung, die erst nach den Kundenanforderungen erstellt werden muss. Der Betrieb und die Instandhaltung der Software zeichnen sich durch einen niedrigen Personalaufwand aus. Das führt ebenfalls zu einer Senkung der Kosten, da meist der Hersteller der Software die Updates für die Software und den Support dafür übernimmt. Es wird also weniger IT-Personal benötigt, was zu einer Entlastung der Personalsituation und des Aufgaben-Backlogs führt. Problematisch bei der Standardsoftware ist hingegen, dass diese in den seltensten Fällen vollständig auf die Anforderungen der Anwender passt. Dies ist bedingt durch den standardisierten Charakter der Software, der möglichst viele Anwendungsfälle im Aufgabengebiet abdecken soll. Dies kann zu zusätzlichen Kosten führen, wenn das Unternehmen eine etwas individuellere Anpassung benötigt. Als weiterer Faktor ist zu berücksichtigen, dass innerbetriebliche Prozesse auf die Software angepasst werden müssen. Dies wird erforderlich, weil die Anpassungsmöglichkeiten solcher Standardsoftware bzw. der Bearbeitungsaufwand für spezifische Anpassungsanforderungen begrenzt sind. Dies kann den Einsatz von Standardsoftware unrentabel werden lassen, da der Aufwand dann demjenigen für die Entwicklung einer Individualsoftware gleichkommt.

Eine Sonderstellung nimmt das End-User- oder Power-User-Computing ein.<sup>53</sup> In diesem Zusammenhang lassen sich die Power-User charakterisieren als erfahrene Fach- und Hauptanwender, wohingegen End-User die konventionellen Anwender sind. Hierbei wird die Entwicklung von Arbeitsplatzsoftware an den End-User übertragen. Dabei entwickelt der End-User Kleinstanwendungen für seinen eigenen Arbeitsbereich zur Verbesserung der eigenen täglichen Arbeit. Die Problemstellung, die dabei entsteht, ist ähnlich der des Citizen Developments, dass Insellösungen oder eine Schatten-IT entstehen können. Im Gegensatz zum Citizen Development, wo nur bestimmte Personen der Fachabteilungen Anwendungen entwickeln, ist es beim End-User-Development potenziell jeder Anwender, der Lösungen entwickelt. Es kann hierdurch die grundlegende Problemstellung entstehen, dass viele verschiedene Lösungen erstellt werden, die von der IT nicht gemonitored

---

<sup>52</sup> vgl. Le me ster 2021, S. 330.

<sup>53</sup> vgl. Krcmar 2015, S. 211.



werden. Das bindet Kapazitäten der entsprechenden End-User oder IT-Mitarbeiter. Das kann zu Governance- und Sicherheitsrisiken führen.

Ein weiteres Problem ist, dass durch die verschiedenen Lösungen, welche auf gleichen Datensätzen beruhen, viele verschiedene Datenbanken entstehen, die den gleichen Dateninhalt besitzen, aber auf die End-User-Anwendung angepasst sind. Diese Dateien können nur schwer alle auf dem gleichen Stand gehalten werden, was zu Schnittstellenverlusten oder Fehlern führt, wenn eine veraltete Datenbank von einem Mitarbeiter für den betrieblichen Zweck verwendet wird. Ein zusätzlicher Aufwand für Unternehmen ist dabei, dass es bei diesem Konzept erforderlich ist, dass der End-User-Entwickler Kenntnisse über Programmierung besitzt, um die entwickelte Anwendung verwenden zu können.<sup>54</sup> Aufgrund der Tatsache, dass es potenziell jeden Mitarbeiter betreffen kann, ist mit einem erhöhten Aufwand für die Befähigung zu rechnen.

## 2.4.2 Einordnung von No Code/Low Code

Wie bereits erwähnt, ist Citizen Development der End-User-Entwicklung sehr ähnlich und kann als eine Erweiterung dieses Systems verstanden werden.<sup>55</sup> Diese Hierarchie der Softwarebereitstellungen in Unternehmen ist in Abbildung 2 dargestellt. Einer der Hauptunterschiede ist, dass End-User-Entwicklung meist mehr eingeschränkt ist auf eine spezifische Softwareanwendung und sehr stark an deren Grenzen gebunden ist. Wie bereits erwähnt, wird bei der End-User-Entwicklung potenziell jeder Mitarbeiter Anwendungen für seinen persönlichen Anwendungsfall entwickeln. Ein Beispiel wäre in diesem Fall, dass aus einem ERP-System Daten extrahiert werden und von einem End-User in einer Excel-Tabelle weiterverarbeitet werden. Diese Excel-Tabelle hat der End-User auf sich persönlich angepasst, meist ohne Dokumentation, wie die Formeln funktionieren und woher die Daten kommen. Im Weiteren ist der End-User sehr stark an die Beschränkungen des Programms Excel gebunden. Will er tieferegehende Automationen und Auswertungen nutzen, wie Visual Basic for Applications, werden weitreichende Programmierkenntnisse nötig. Dies kann das Fähigkeitslevel einiger Mitarbeiter übersteigen, weshalb der Output solcher Entwicklungen stark schwanken kann. Die Daten werden also nicht vollständig verwertet, sondern überwiegend nur in begrenztem Maße verändert oder erweitert. Es finden keine konkrete tiefgreifende Verarbeitung der Daten und daraus resultierende Automationen statt, um mit diesen Daten weiterzuarbeiten. Begleitend zu diesem Problem kommt die bereits erwähnte Problemstellung von Insellösungen und Schatten-IT.

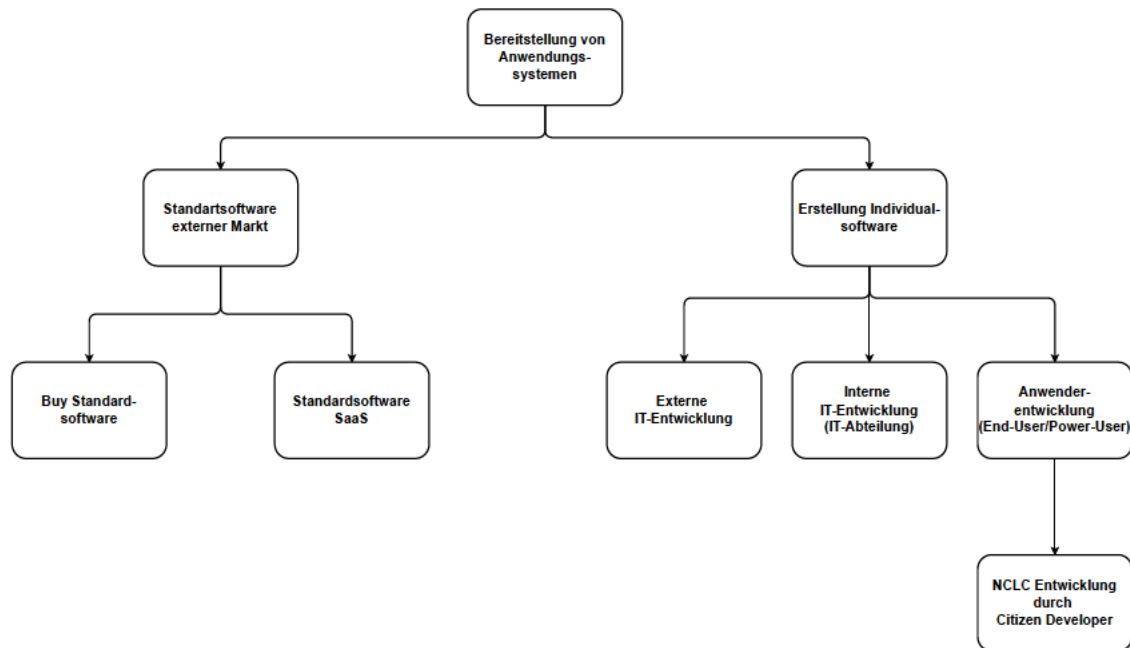
Eine NCLC-Plattform und Citizen Development zielen darauf ab, eine Gruppe von Nicht-IT-Personal zu befähigen, eigene Applikationen zu bauen, die sie spezifisch auf ihre Bedürfnisse anpassen können, also ihre eigene Version von Excel zu entwickeln. Dadurch soll es den End-Usern ermöglicht werden, die Daten entsprechend automatisiert weiterzuverarbeiten. Hier wird die Entwicklung von einem definierten Personenkreis innerhalb der Abteilungen vorangetrieben und diese sind als Ansprechpartner vor Ort. Dies als ganzheitliches Konzept betrachtet, mit einer Begleitung durch die IT-Abteilung und einer implementierten Governance- und Sicherheitsstrategie, führt zu einer teilweisen Aufhebung der Problemstellungen, die durch das klassische End-User-Development entstehen. Es wird für End-User möglich, weitgehende Automatisierungen selbst zu nutzen und diese werden, im Zusammenspiel mit dem Fachwissen der Fachabteilung, integriert in die Prozesse der

---

<sup>54</sup> vg. Paternò and Wu f 2017, S. 2.

<sup>55</sup> vg. Borgman 2022, S. 4718.

Fachabteilung. Dies ermöglicht eine bessere Entwicklung mit den spezifischen Anforderungen der Fachabteilung. Es ist nicht mehr zwingend erforderlich, eine IT-Abteilung voll einzubinden, die eine komplexe Anwendung erst erstellen und dann nacharbeiten muss, weil es eventuell zu Fehlern in der Kommunikation gekommen ist. Die Verluste in der Kommunikation entstehen durch eine schlechte Kommunikation mit der IT-Abteilung und durch die Gegebenheit, dass das IT-Personal oft nicht das Fachwissen des Fachmitarbeiters hat. Dadurch kommt es zu schlechteren Anwendungen, welche eventuell nachgearbeitet werden müssen, um einen optimalen Nutzen für die Fachabteilung zu erzielen.



**Abbildung 2 - Alternativen der Softwarebereitstellung<sup>56</sup>**

NCLC und Citizen Development vereinen durch diese Eigenschaften die Vorteile von Standardsoftware und Individualsoftware und kompensieren die Nachteile in gewissem Maße. Wie stark diese Kompensation ausfällt, ist abhängig von der gewählten Plattform. Als ein ganzheitliches Konzept wird der positive Aspekt erzielt, Applikationslösungen auf die jeweiligen Bedürfnisse der Abteilung anzupassen, ohne dabei die Vorteile von Standardsoftware wie Kostenfaktor und Support komplett zu verlieren, da der Plattformbetreiber dies zum größten Teil übernimmt. Es wird ein erhöhter Aufwand an Governance und Betreuung durch die IT-Abteilung erzeugt, was bei erfolgreicher Implementierung jedoch die Kostenersparnis und Aufwandsersparnis nicht oder nur teilweise reduzieren sollte.

Ein aktuelles Problem in diesem Bereich ist, dass Fachbereiche am wenigsten zufrieden mit den NCLC-Plattformen sind.<sup>57</sup> Hier ist die Zufriedenheit mit den NCLC-Plattformen am geringsten. Dies kann sich darin begründen, dass nur eine Minderheit an Unternehmen den Bereich Citizen Development als zufriedenstellend einstuft. Ursächlich für diesen Zustand kann eine mangelhafte Implementierungsstrategie sein, oder dies kann durch Fehler erklärt werden, die während des Einführungsprojekts gemacht wurden. Das Hauptpotenzial der NCLC-Plattformen liegt in der Anwendung durch Power-User und Citizen Developer. Angesichts dessen ist eine erfolgreiche Einführung wichtig, um nicht die Motivation zur Nutzung der Plattform zu reduzieren und dadurch eine suboptimale Nutzung oder ein eventuelles Scheitern zu riskieren.

<sup>56</sup> vg. Krcmar 2015, S. 212; vg. Borgman 2022, S. 4717.

<sup>57</sup> vg. Schaffry 2022, S. 20.

### 3 Vorgehensmodelle im Projektmanagement

Unter dem Begriff Vorgehensmodell wird eine Sammlung von Regelungen und Definitionen verstanden, welche sich verschiedener Elemente und Techniken bedienen und zur Umsetzung eines spezifischen Ziels benötigt werden.<sup>58</sup> Im Bereich des Projektmanagements gibt es verschiedenartige Vorgehensmodelle, die je nach Projektanforderungen und Ausgangsbedingungen Anwendung finden.<sup>59</sup> Das Vorgehen innerhalb der Modelle wird unterschieden in folgende Kategorien:

- Sequenzielle bzw. lineare Vorgehen,
- Überlappende oder parallele Vorgehen,
- Wiederholende bzw. iterative Vorgehen.

Zur Beantwortung der in dieser Arbeit gestellten Forschungsfragen werden die in diesem Kapitel vorgestellten Vorgehensweisen mit den nachfolgenden Kriterien aufgeschlüsselt:

- Voraussetzungen für Modell,
- Wichtige Rollen im Projekt,
- Projektorganisationsform,
- Vorgehen im Projekt,
- Vor- und Nachteile,
- Wie wird mit Änderungen umgegangen?,
- Wie wird Qualität gesichert bzw. der Erfolg des Projektes gemessen?

Durch diese Aufschlüsselung soll es ermöglicht werden, eine Vergleichbarkeit der Modelle herzustellen. Dadurch wird es im weiteren Verlauf der Arbeit möglich sein, ein Vorgehensmodell herauszuarbeiten, welches an qualitativen Kriterien messbar ist.

#### 3.1 Lineare/Sequenzielle Modelle

Lineare Modelle, auch traditionelle Modelle genannt, folgen einem klaren zeitlich gegliederten Ablauf verschiedener Projektphasen.<sup>60</sup> Es gilt das Prinzip, dass sich die Phasen von der Grobplanung hin zum Detail entwickeln, um dadurch ein hohes Maß an Planungssicherheit zu erzeugen. Dies wird in Abbildung 3 einmal idealtypisch dargestellt. Der Zweck ist, dass ein Projekt in verschiedene, übersichtliche Teilabschnitte aufgeteilt wird. Diese sequenziellen Modelle werden häufig in Bereichen genutzt, wo ein hohes Maß an Sicherheit benötigt wird für die Bereiche Scope, Budget und Cost. Beispielhaft können hier genannt werden: das Baugewerbe sowie Eventprojekte.<sup>61</sup> Ein traditioneller Ansatz ist allerdings dort nicht zwingend erforderlich, es geht nur um die Verdeutlichung der Besonderheiten von traditionellen Modellen. Aufgrund dieses Mechanismus des traditionellen Projektmanagements ist es möglich, verschiedene Meilensteine und Korrekturpunkte im Projekt zu etablieren, um Projektfortschritt und Zwischenergebnisse zu messen. Die Anzahl der verschiedenen Phasen ist abhängig von den verschiedenen Einflussfaktoren des jeweiligen Projektes, wie die Art des Projektes, das Risiko, der Umfang und die Bedeutung für die Organisation. Ein weiterer Bestandteil von traditionellen Modellen ist es, zu berücksichtigen, wie viel Einfluss durch den Auftraggeber selbst gewünscht ist. Änderungen am Projekt lassen sich meist nur mit erhöhtem Aufwand realisieren, weshalb eine ausführliche Dokumentation der Projektanforderungen nötig ist. Am Anfang eines linearen Projektes stehen, in der Theorie, die Anforderungen und am Ende erhält der

<sup>58</sup> vg. Versteegen 2002, S. 29; vg. Epp ng 2011, S. 17; vg. Kneuper and Müller-Luschnat 1998, S. 16.

<sup>59</sup> vg. Dechange 2020, S. 33.

<sup>60</sup> vg. Kuster et al. 2011, S. 17.

<sup>61</sup> vg. Peters and Scheiter 2021, S. 16.

Projektauftraggeber einen fertigen Projektoutput, unter anderem in Form eines Produktes oder einer Dienstleistung.



**Abbildung 3 - abgewandeltes Phasenmodell<sup>62</sup>**

Die verschiedenen Phasen lassen sich wie folgt beschreiben:

1. Die Problemanalysephase/Initialisierungsphase, in der die Anforderungen und Zielstellungen des Projektes genau spezifiziert und Vorbereitungen für das Projekt getroffen werden. Diese Phase beinhaltet das Erfassen des Problems und den Entschluss, dieses zu beheben. Die Problemstellung kann hierbei bereits konkret formuliert sein, allerdings auch erst als vage Wahrnehmung vorliegen. Voraussetzung für diese Phase ist, dass die involvierten Organisationseinheiten es als Problem akzeptieren und an einer Lösung mitarbeiten wollen, in Form von Ressourcen organisatorischer, personeller oder finanzieller Art. Das optimale Ergebnis dieser Phase ist eine Projektvereinbarung. Ein in der Praxis erprobtes Mittel für die Dokumentation ist der Projektsteckbrief.<sup>63</sup> Dort werden unter anderem der Projektumfang, die Projektziele sowie das Projektbudget genannt und Aussagen zur Ressourcenzuteilung getroffen.<sup>64</sup> Der Projektsteckbrief ist eine erste Absichtserklärung und soll für Kommittent bzw. Selbstverpflichtung zwischen den Projektbeteiligten sorgen.
2. In der konzeptionellen Grundlagenphase werden erste Entwürfe zur Durchführung analysiert und die Realisierbarkeit des Projektes geprüft. Gleichzeitig werden in dieser Phase zusätzlich alternative Lösungswege gestaltet und überprüft. Es wird geprüft, welche Relationen und Mechanismen im Problemfeld wirken, wie der genaue Projektumfang ist und was nicht Teil des Projektes ist. Es wird analysiert, ob das identifizierte Problem die Ursache für den unerwünschten Effekt ist oder es noch eine Ursache auf einer höheren Ebene gibt. Welche Projektrisiken sind vorhanden und welche Auswirkungen haben diese auf den Verlauf des Projektes?

<sup>62</sup> .A. Dechange 2020, S. 34; .A. Kuster et a . 2011, S. 14.

<sup>63</sup> vg . Mens ng 2015, S. 21.

<sup>64</sup> vg . Käne 2020, S. 46.

Wichtig ist, dass es zu einer klaren Zielformulierung des gewünschten Ergebnisses kommt und deutlich wird, welcher der evaluierten Lösungswege das bestmögliche Ergebnis liefert. Ergebnis dieser Phase ist eine erste umfassende Planung des Ablaufs, was eine Voraussetzung für die weiteren Phasen bildet.<sup>65</sup> In dieser Phase werden wesentliche Entscheidungen getroffen, die den gesamten Verlauf des Projektes beeinflussen, und die deshalb entsprechend präzise und sorgfältig durchgeführt wird.

3. Die Phase der detaillierten Gestaltung beinhaltet die Ausarbeitung der ausgewählten Lösungskonzepte.<sup>66</sup> In dieser Phase soll es zu einer detaillierten Planung der verschiedenen Lösungen kommen, die Beurteilungen enthält für die Sinnhaftigkeit, Machbarkeit und wirtschaftliche Realisierung. Zu diesem Zweck wird für jede mögliche Lösungsvariante ein Konzept erstellt, welches für die spätere Detailplanung wichtig ist. Diese Konzepte enthalten einen Rahmenplan mit Meilensteinen, Einsparungsmöglichkeiten, Definitionen von Teilprojekten und möglichen Investitionsentscheidungen. Anhand dieser Daten wird eine Lösungsvariante ausgewählt. Danach gilt es, die ausgewählte Lösung im Detail zu planen. Dabei werden detaillierte Lösungskonzepte entwickelt und Entscheidungen über den Verlauf des Projektes getroffen. In einem weiteren Schritt kommt es zur Konkretisierung von Teillösungen, damit diese effektiv eingebunden werden können. Es werden ebenfalls Vorbereitungen für ein Change Request Management durchgeführt, um ein standardisiertes Vorgehen zu ermöglichen, im Falle von Änderungswünschen im Verlauf des Projektes.
4. Die Realisierung des in Phase 3 ausgewählten Konzeptes findet in Phase 4 statt, die Realisierungsphase genannt wird. Dabei werden erste Prototypen entwickelt und gefertigt.<sup>67</sup> Im Anschluss an das Prototyping kommt es zu einer Finalisierung des Projektoutputs. Es wird eine Dokumentation angefertigt, welche unter anderem die Bedienungsanleitungen, Prozessbeschreibungen, Regelungen für Informationsflüsse und Störungsmeldungen enthält, sowie, wenn benötigt, Wartungs- und Instandhaltungskonzepte oder Ähnliches. Eventuell entwickelte Teilsysteme werden in die Gesamtlösung integriert und das Ergebnis des Projektes kann bei kleinen Projekten an den Kunden übergeben werden. Bei größeren Projekten empfiehlt es sich jedoch, dies in der 5. Phase zu tun.
5. Die 5. Phase eines Projektes beschäftigt sich mit der Einführung des Projektes und der Nachbereitung sowie der Projektbewertung.<sup>68</sup> In dieser Phase kommt es zur finalen Einführung des Projektergebnisses und einer abschließenden Bewertung des Projektes. Bei sehr komplexen Systemen empfiehlt es sich, die Einführung schrittweise vorzunehmen und mit den Erfahrungen der einzelnen Einführungsschritte den Fortschritt voranzutreiben. Diese Phase kann sehr schwierig und langwierig sein. Wichtig ist in dieser Phase der Einführung, dass das im Projekt gewonnene Wissen durch die Projektbeteiligten adäquat für alle übrigen Nutzer des Ergebnisses transferiert wird und eine gute Zusammenarbeit einen reibungslosen Ablauf ermöglicht. Falls nicht in Phase 4 geschehen, erfolgt nun spätestens die vollständige Übergabe des Projektergebnisses an den Kunden.

Am Ende eines jeden Projektes kommt es zu einer Auswertung des Projektverlaufs. Dies geschieht bei erfolgreichen wie bei nicht erfolgreichen Projekten. In

---

<sup>65</sup> vg. Kuster et al. 2011, S. 20.

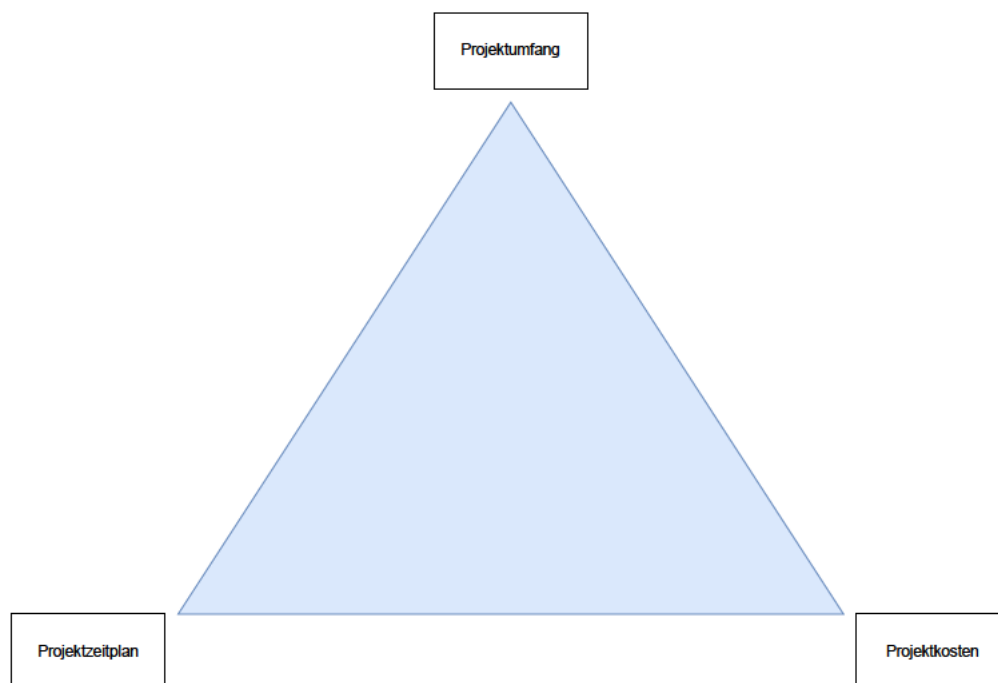
<sup>66</sup> vg. Kuster et al. 2011, S. 21.

<sup>67</sup> vg. Kuster et al. 2011, S. 22.

<sup>68</sup> vg. Kuster et al. 2011, S. 23.

diesem letzten Schritt werden Schlussabrechnungen, Vervollständigung der Projektdokumentation und Planung der Archivierung durchgeführt. Es kommt zu einem Projektabschluss mit dem Projektteam und dem Auftraggeber, wo ein Projekt-rückblick durchgeführt wird, bei dem die Erfahrungen und Fehler aus dem Projekt festgehalten werden.

Ein Nachteil des sequenziellen Frameworks liegt in der Problematik von Änderungswünschen. Sollte es passieren, dass während des Projektverlaufs größere Änderungswünsche auftauchen, kann dies zu Problemen in initialen Umfangs-, Kosten- oder Zeitplanungen führen. Aufgrund dessen befindet sich das sequenzielle Projektmanagement in einem ständigen Zielkonflikt.<sup>69</sup> Dies wird am besten durch das in Abbildung 4 gezeigte magische Dreieck des sequenziellen Projektmanagements aufgezeigt. Jede Änderung zugunsten einer der drei Dimensionen an den Ecken des Dreiecks erfolgt zuungunsten von mindestens einer anderen Dimension – beispielsweise verursacht eine größere Änderung des Projektumfangs, dass das Projektmanagement die Planungsphasen erneut durchlaufen muss, was zulasten von Kosten und Zeitplan geht. Jede Dimension steht in direkter Relation zu den anderen und kann in der Regel nicht verändert werden, ohne die anderen zu beeinflussen. Dieser Nachteil hat dazu geführt, dass andere Vorgehensmodelle sich vom statischen Vorgehen des sequenziellen Ansatzes abgewandt haben und zu flexibleren Vorgehensmodellen gewechselt wurde.



**Abbildung 4 - magisches Dreieck sequenzielles PM<sup>70</sup>**

Dies hat den Vorteil, dass es zu einer umfassenden und gründlichen Planung innerhalb des Projekts kommt. Voraussetzung dafür ist, dass die meisten Anforderungen des Projektes klar sind und sich diese nicht mehr wesentlich verändern.<sup>71</sup> Dies führt zu einer starken Planungssicherheit, da Termine, Qualität, Quantität und Liefergegenstände von Teilprojekten vorher festgelegt werden. Das Budget und die sachlichen sowie personellen

<sup>69</sup> vg. Keßler und Wnkehofer 2004, S. 55.

<sup>70</sup> in Anlehnung an Keßler und Wnkehofer 2004, S. 55.

<sup>71</sup> vg. Käne 2020, S. 70.

Ressourcen werden vorher festgelegt und sind dadurch klar kalkulierbar. Dieses Vorgehen entspricht einer klassischen Top-down-Struktur und ist weitgehend etabliert, weshalb es bei dieser Art von Projekten keine größeren Akzeptanzprobleme geben sollte.<sup>72</sup> Die Nachteile lassen sich aus den Vorteilen schlussfolgern. Projekte, bei denen die Anforderungen nicht bekannt sind oder einem ständigen Veränderungs- und Anpassungsprozess unterliegen, scheiden für dieses Vorgehen aus. Vor allem in der Softwareentwicklung ist dies der Fall, weshalb dort auf die in Kapitel 3.2 vorgestellten agilen Methodiken bevorzugt zurückgegriffen wird.<sup>73</sup>

Abschließend kann festgehalten werden, dass sich der traditionelle Ansatz primär für das Management von komplizierten Anforderungen, die bereits bekannt sind, eignet. Dem gegenüber steht ein erhebliches Problem beim Management von komplexen Projekten, deren Anforderungen unbekannt sind oder sich ändern.<sup>74</sup>

### 3.1.1 Wasserfallmodell

Das Wasserfallmodell, auch engl. Waterfall model genannt, beschreibt einen phasenorientierten Ansatz, der in einer klaren zeitlichen und logischen Abfolge voneinander getrennt ist.<sup>75</sup> Dies ist das klassischste der linearen Modelle und wurde bereits 1970 durch Winston Royce in seiner Systematik vorgestellt.<sup>76</sup> Im strengsten Fall kann eine neue Projektphase erst begonnen werden, wenn die vorherige beendet wurde. Es besteht eine klare Ende-zu-Anfang-Beziehung (EA).

Bei diesem Modell wird am Anfang geklärt, welche Qualität die Zielergebnisse erfüllen müssen. Die Planung erfolgt zu Beginn und Änderungen sind nur über ein umfangreiches Changemanagement möglich. Changemanagement beschäftigt sich mit dem Reagieren auf Veränderungen im Projekt und ist abzugrenzen von Change-Projekten, die sich mit der Veränderung von Menschen beschäftigen.<sup>77</sup> Am Phasenende wird der Output der Phase überprüft und gegebenenfalls muss die Phase erneut durchlaufen werden. Im Falle eines Fehlers, welcher erst in der nachfolgenden Phase festgestellt wird, kann in die vorhergehende Phase zurückgekehrt werden, somit können Korrekturen durchgeführt werden.<sup>78</sup>

Wichtige Rollen im Wasserfallmodell sind der Projektleiter, der Projektauftraggeber und die Teammitglieder, die je nach Strukturform des Projektes dem Projekt direkt zugeordnet werden oder in ihren jeweiligen Fachabteilungen verbleiben. Die Vorteile dieses Modells liegen darin, dass es schnell erfassbar und leicht verständlich ist.<sup>79</sup> Der Fokus liegt auf der Qualität des Projektoutputs, durch regelmäßiges Überprüfen der vorherigen Phase. Durch die Vorausplanung der Phasen ermöglicht das Wasserfallmodell eine gute Planungssicherheit. Es ist auch möglich, durch Methodiken wie Critical Path Method, kurz CPM, eine Parallelisierung von verschiedenen Arbeitspaketen vorzunehmen. Diese Methodik ist ein Verfahren aus der Netzplantechnik und ermöglicht es dem Anwender, einen grafischen Ablauf der Arbeitspakete zu erstellen und dadurch die wichtigen Zeiten eines

---

<sup>72</sup> vg. Becker 2013, S. 118.

<sup>73</sup> vg. Dechange 2020, S. 38.

<sup>74</sup> vg. Habermann 2013, S. 96.

<sup>75</sup> vg. Dechange 2020, S. 33.

<sup>76</sup> vg. Royce 1970, S. 328 ff.

<sup>77</sup> vg. Kuster et al. 2011, S. 30.

<sup>78</sup> vg. Stöher, Förster, and Brehm 2018, S. 25.

<sup>79</sup> vg. Stöher, Förster, and Brehm 2018, S. 25.

Arbeitspakets, wie den frühesten und spätesten Start, zu erfahren.<sup>80</sup> Die CPM baut dabei auf dem grafischen Plan des Netzplanes auf. Als Nachteil ist jedoch zu erfassen, dass diese Methodik einer sehr starren Struktur folgt und Änderungen in den Dimensionen des Projektes sich nur durch größeren Aufwand realisieren lassen. Das Ergebnis des Projektes ist erst am Ende verfügbar, genauso wie das Feedback des Projektauftraggebers, was das Problem aufwirft, dass erst am Ende der Erfolg gemessen werden kann. Geht etwas schief oder gibt es ein Missverständnis zwischen den Projektbeteiligten, kann es passieren, dass dies zu spät auffällt.

### 3.1.2 V-Modell

Das V-Modell zeichnet sich dadurch aus, dass die Phasen in einer namensgebenden V-Form angeordnet werden, wie in Abbildung 5 dargestellt.<sup>81</sup> Im linken Arm wird nach dem Top-down-Prinzip die Erstellung des Lieferobjektes geplant und dem Bottom-up-Prinzip folgend werden im rechten Arm die Validität und Verifizierung des Projektliefergegenstands beschrieben. Wichtige Rollen sind, wie bei traditionellen Modellen üblich, der Projektleiter und das Projektteam. Der Projektauftraggeber gibt meist nur am Anfang die Anforderungen vor und wird über Zwischenergebnisse sowie das Endergebnis informiert. Die Phasen im V-Modell sind ein Nachfolgemodell des Wasserfallmodells, was in der Version V-Modell 92 noch stark erkennbar war.<sup>82</sup> In der Version V-Modell 97 waren jedoch bereits iterative Ansätze zu bemerken. Auf der linken Entwicklungsseite beginnt die erste Phase mit der Feststellung der Anforderungen an den Projektoutput.<sup>83</sup> In der zweiten Phase werden unter anderem die Projektprozesse und Datenmodelle definiert. Die dritte Phase zeichnet sich durch die Designentwicklung der Projektoutputs aus und wird durch das Projektteam durchgeführt. Die letzte Phase der Entwicklungsseite ist die Implementation: Während dieser Phase wird der Projektoutput durch das Projektteam implementiert und betriebsbereit gemacht. Dadurch kann in der nächsten Phase, der Testphase, damit begonnen werden, die Validierung der Projektergebnisse durchzuführen. Im Anschluss an die Testphase erfolgt die Komponententestphase. In dieser Phase werden die einzelnen Bestandteile des Projektoutputs getestet, um daran anschließend in der nächsten Phase mit den Integrationstests die Schnittstellen zu überprüfen. Sobald der Integrationstest abgeschlossen ist, kann mit der Systemtestphase begonnen werden. Hier wird der gesamte Projektoutput als Ganzes betrachtet und getestet. In der letzten Phase wird dann der Projektoutput mit dem Endkunden zusammen abgenommen.

Das V-Modell legt in seinem Vorgehen besonderen Wert auf Qualität und die vollständige Erfüllung der Anforderungen.<sup>84</sup> Die Kontrolle über Qualität und Anforderungserfüllung einer Phase erfolgt in der jeweils darauffolgenden Phase. In der Analysephase wird beispielsweise überprüft, ob die Anforderungen, welche in der Anforderungsphase festgelegt wurden, richtig umgesetzt werden. Dieser Vorgang erfolgt in jeder Phase des Modells. Eine weitere Besonderheit dieses Modells besteht in der Fehlersuche. Zum Beispiel wird während des Integrationstests ein Fehler festgestellt, dann wird in die im V-Modell gegenüberliegende Phase zurückgesprungen, um dort den Fehler zu finden. Es wird also von der Testseite in die Entwicklungsseite zurückgesprungen. Ist der Fehler gefunden, wird

---

<sup>80</sup> vg. Käne 2020, S. 186.

<sup>81</sup> vg. Dechange 2020, S. 35.

<sup>82</sup> vg. Versteegen 2002, S. 32.

<sup>83</sup> vg. Bertram, Bönniggen, and Bröh 1993, S. 60. Achten Sie hier bitte auf die Schreibweise!

<sup>84</sup> vg. Stöhr, Förster, and Brehm 2018, S. 25.



das Modell erneut durchlaufen, beginnend mit der Phase, in die zurückgesprungen wurde. Ein anderes Beispiel ist hier: Wenn während des Abnahmetests ein Fehler entdeckt wird, muss in die Anforderungsphase zurückgesprungen werden und von dort aus die Fehlersuche beginnen, durch alle Phasen hindurch, bis der Abnahmetest erneut erreicht wird. Zur Realisierung dieses Vorgehens werden bereits in den Phasen der Entwicklungsseite die Testfälle und Kriterien entwickelt, die dann in den entsprechenden gegenüberliegenden Phasen angewendet werden, um das Ergebnis zu überprüfen. Dies bedeutet, dass bereits während der Anforderungsphase die Testfälle für den Abnahmetest oder während der Designphase die Testfälle für den Integrationstest entwickelt werden. Diese Aspekte der Kontrolle und Fehlersuche werden Verifikation und Validierung genannt. Die Stärken und Schwächen dieses Modells ähneln denjenigen des Wasserfallmodells. Es gibt auch hier sehr klare Vorgaben zum Projektoutput und die Struktur des Vorgehensmodells ist einfach gehalten. Es sind bereits zu Beginn der Projektumfang und die Kosten bekannt, was ein hohes Maß an Planungssicherheit ermöglicht. Der Auftraggeber ist nur zu Beginn involviert und muss sich nicht mit den Details beschäftigen. Daraus ergibt sich allerdings der direkte Nachteil, dass erst am Ende der Erfolg bewertet werden kann. Die weiteren Nachteile bestehen hier, genauso wie im Wasserfallmodell, darin, dass Änderungsanforderungen nur mit einem erhöhten Kostenaufwand bzw. einem erhöhten Zeitaufwand umgesetzt werden können. Die Dimensionen des Projektes sind immer noch in starker Abhängigkeit zueinander.

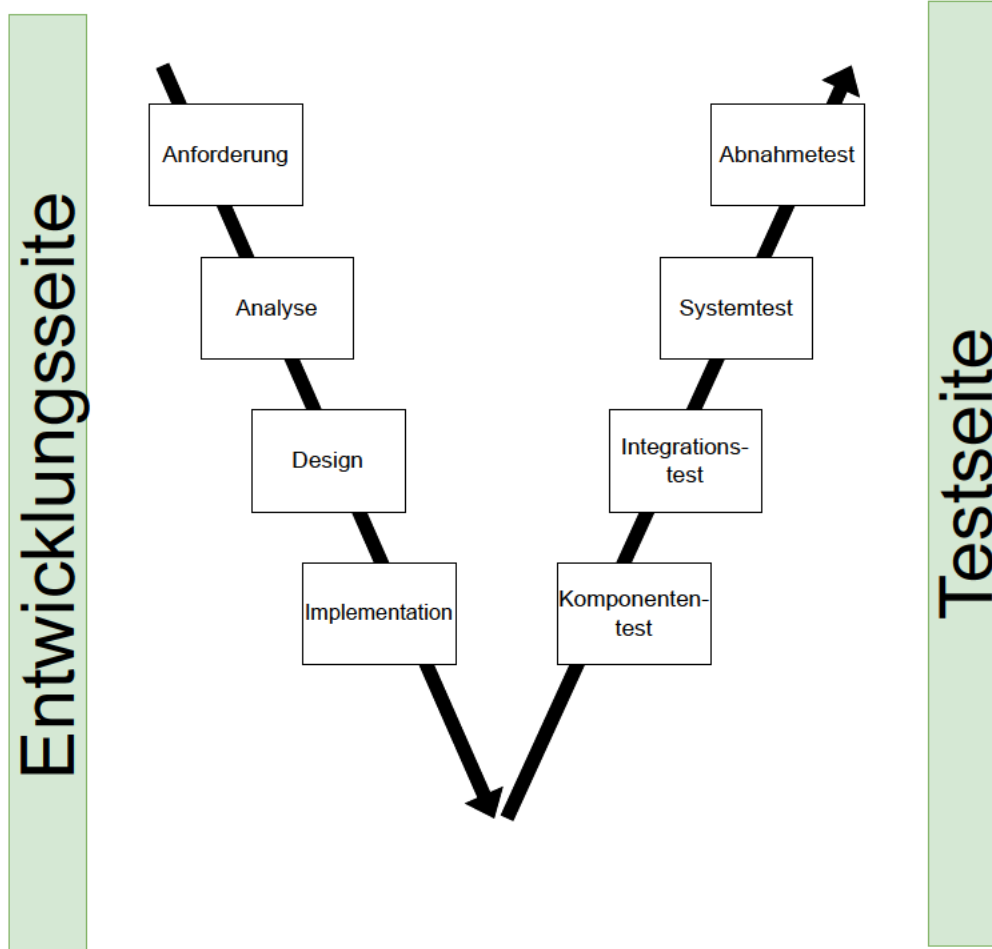


Abbildung 5 - V-Modell<sup>85</sup>

<sup>85</sup> n An ehnung Käe 2020, S. 60; n An ehnung Stöh er, Förster, and Brehm 2018, S. 26.

## 3.2 Agiles Projektmanagement

Agiles Management hat sich aus dem Bedarf heraus entwickelt, die Schwächen des klassischen traditionellen Projektmanagements zu lösen.<sup>86</sup> Es sollte möglich sein, in möglichst kurzer Zeit eine Software zu entwickeln, die den Kundenwünschen entspricht, ohne dass alle Anforderungen an die Software bereits bekannt sind. Dadurch wird ein hohes Maß an Flexibilität erreicht. Fehlentwicklungen können schnell korrigiert werden und dadurch werden Risiken minimiert. Wichtige Grundgedanken dieser Vorgehensweise wurden im Agilen Manifest von 2001 festgehalten, wo die Autoren sich auf verschiedene grundlegende Werte des agilen Projektmanagements geeinigt haben.<sup>87</sup> Das Agile Manifest wird von vielen verschiedenen Quellen als ein wichtiger Grundsatz des agilen Projektmanagements angesehen.<sup>88</sup> Daraus entstanden sind vier Leitsätze, an denen sich das agile Management orientiert.<sup>89</sup> Durch diese findet eine Gewichtung verschiedener Werte statt, die in einem Projekt existieren und es kommt dahingehend zu einer Hierarchie, welcher Wert bei einer Entscheidung zu bevorzugen ist. Die Leitsätze lauten wie folgt:<sup>90</sup>

- „Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge“,
- „Funktionierende Software ist wichtiger als umfassende Dokumentation“,
- „Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlungen“,
- „Reagieren auf Veränderung ist wichtiger als das Befolgen eines vorher festgelegten Plans“.

Eine Anmerkung zu den vier Leitsätzen wird hinzugefügt: „Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“<sup>91</sup>

Die Prinzipien, denen das agile Vorgehen laut den Autoren folgen soll, umfassen zwölf Punkte:

- Höchste Priorität hat die Zufriedenheit des Kunden, das soll erreicht werden durch frühe und regelmäßige Auslieferung von wertvoller Software.
- Anforderungsänderungen sind zu jedem Zeitpunkt des Projekts zu begrüßen. Das agile Vorgehen nutzt Veränderung, um dem Kunden einen Vorteil gegenüber seiner Konkurrenz zu geben.
- Funktionierende Software soll regelmäßig ausgeliefert werden. Dabei ist die kürzeste Zeitspanne zu bevorzugen.
- Fachexperten und Entwickler müssen während der gesamten Projektlaufzeit regelmäßig, wenn es sinnvoll ist, sogar täglich, zusammenarbeiten.
- Baue Projekte um motivierte Personen auf und gib ihnen die Ressourcen, die sie zur Unterstützung brauchen. Vertraue darauf, dass die Aufgaben erledigt werden.
- Von Angesicht zu Angesicht ist die beste Methodik, um Informationen innerhalb des Entwicklerteams zu übermitteln.
- Als Maßstab für den Fortschritt des Projekts ist funktionierende Software der beste Maßstab.
- Den Blick ständig auf technische Perfektion und gutes Design zu richten, fördert die Agilität im Projekt.
- Einfach ist immer besser.

---

<sup>86</sup> vg . Kuster et a . 2011, S. 29; vg . Dechange 2020, S. 38; vg . Madauss 2020, S. 39.

<sup>87</sup> vg . Beck et a . 2001.

<sup>88</sup> vg . Kuster et a . 2011, S. 29; vg . Stöh er, Förster, and Brehm 2018, S. 26; vg . Dechange 2020, S. 39.

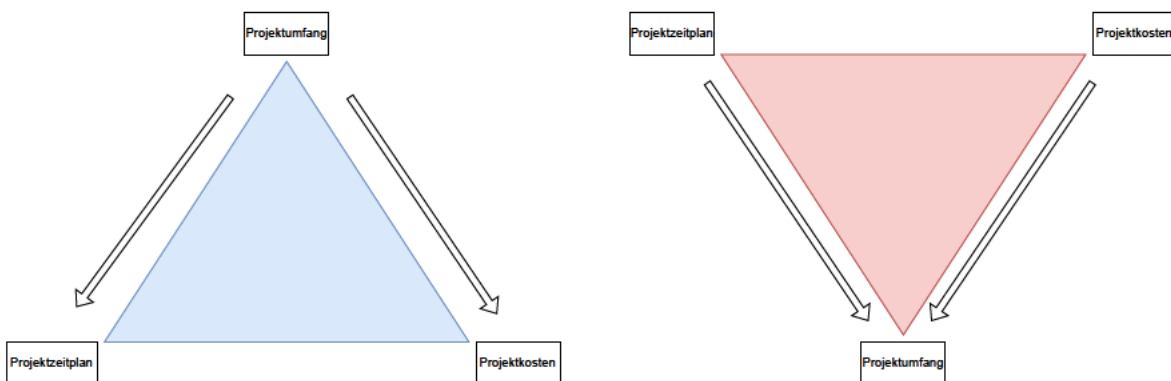
<sup>89</sup> vg . Beck et a . 2001.

<sup>90</sup> vg . Beck et a . 2001.

<sup>91</sup> vg . Beck et a . 2001.

- Die Projektteams wissen am besten, wie diese sich organisieren sollten.
- In regelmäßigen Zeitabschnitten, wenn nötig, sogar täglich, reflektiert das Team, welche Probleme aufgetaucht sind und welche Maßnahmen nötig sind, diese Probleme zu beseitigen.

Diesen Prinzipien des agilen Projektmanagements folgend, verändert sich das in Kapitel 3.1 in Abbildung 4 vorgestellte Magische Dreieck. Dabei verändern sich die Relationen zwischen den Dimensionen. Im Falle des agilen Ansatzes geben Kosten und Zeit den Rahmen für die Änderungen vor, die im Projekt durchgeführt werden können. Durch den iterativen Charakter dieses Vorgehensmodells kann der Umfang flexibel geändert werden, ohne zwingend höhere Kosten und größere Zeitänderungen nach sich zu ziehen. Dies ist in Abbildung 6 dargestellt, wobei die Pfeile die Abhängigkeiten der Dimensionen aufzeigen, wie sie sich im Vergleich zum jeweils anderen Dreieck verhalten.



**Abbildung 6 - Vergleich Magisches Dreieck traditioneller & agiler Ansatz<sup>92</sup>**

Diesen Prinzipien folgend, haben sich verschiedene Projektmanagement-Strategien entwickelt, welche die Probleme des klassischen phasenorientierten Modells lösen. Zu nennen ist die Problematik, dass die traditionellen Modelle vorwiegend sehr unflexibel sind und das Kundenfeedback erst sehr spät erfolgt. Zwei der bekanntesten Methoden, die sich aus diesem Änderungsbedarf entwickelt haben, sind unter anderem die Scrum-Methodik und die Kanban-Methodik, die ursprünglich aus dem Lean Management stammt.<sup>93</sup>

Die Vorteile agiler Ansätze liegen in der hohen Flexibilität des Projekts in Bezug auf sich ändernde Anforderungen und die stetige Validierung von Zwischenergebnissen.<sup>94</sup> Diese Stärke spielt agiles Vorgehen vorwiegend bei Projekten aus, bei denen nicht von Anfang an alle Anforderungen an das Projekt bekannt sind und eine Planung nach traditionellen Modellen nicht möglich ist. Durch die Restrukturierung in einfachere, kleinere Elemente kann ein ständiges, sich selbst verbesserndes Projekt gestaltet werden. Durch die Selbstverantwortung im Team steigt die Motivation und das Commitment innerhalb des Teams zum Projekt, was für den Projektoutput förderlich sein sollte. Nachteilig an diesem Ansatz ist jedoch, dass es zu Problemen kommen kann, wenn der agile Ansatz nicht bekannt ist in der Organisation und somit Akzeptanzprobleme auftreten können. Als weiterer Nachteil ist aufzuführen, dass es durch die Vereinfachung komplizierter Projekte zu Planungsproblemen auf einer übergeordneten Gesamtprojektplanungsebene kommen kann. In diesen Fällen kann es sich eher anbieten, auf ein hybrides Modell zurückzugreifen, welches in Kapitel 3.3 besprochen wird. Für Organisationen, die eine strikte hierarchische Organisationsform besitzen, bei der Befehlsgeber und Befehlsempfänger klar geregelt sind, könnte es sein,

<sup>92</sup> n An ehnung an Peters and Sche ter 2021, S. 140.

<sup>93</sup> vg . Komus and Kuberg 2020, S. 3.

<sup>94</sup> vg . Käne 2020, S. 75.

dass die einzelnen Teammitglieder mit einem plötzlichen selbstverantwortlichen Vorgehen Anschlusschwierigkeiten haben, da dies nicht dem normalen Zustand in der Organisation entspricht. Es muss also darauf geachtet werden, dass die Methodik zur Organisation passt und die Akzeptanz für ein neues Vorgehen vorhanden ist. Angesichts dessen sind eventuelle zusätzliche Schulungsmaßnahmen nötig. Agiles Vorgehen in das Projektteam einzubinden wird schwieriger, umso größer das Projektteam ist, allerdings wird dafür an Lösungsansätzen wie SAFe gearbeitet.<sup>95</sup> Die Messung des Erfolgs in agilen Methodiken erfolgt über Zwischenvalidierung von Arbeitspaketergebnissen während der verschiedenen Iterationen, in der sich die Projektarbeitsphasen selbst wiederholen, bis alle Arbeitspakete fertiggestellt wurden. Dies wird in den Methodiken unterschiedlich realisiert.

### 3.2.1 Scrum

Eine der bekanntesten Methodologien des agilen Managements ist die Scrum Methodologie.<sup>96</sup> Mit 84 % ist Scrum der meistgenutzte agile Ansatz, wie in einer Studie der Hochschule Koblenz untersucht wurde.<sup>97</sup> Scrum setzt auf drei wichtige Prinzipien, um die Komplexität des Vorgehens zu reduzieren. Es handelt sich dabei um Transparenz, zyklische Kontrolle und Adaption. Transparenz wird benötigt, weil die Zusammenarbeit zwischen den Teammitgliedern sehr eng geschieht und die Arbeiten der Teammitglieder teilweise miteinander korrelieren. Durch Transparenz ist es dem Organisator und dem Team möglich, schnell zu erkennen, welche Fortschritte erzielt worden sind, wo das Team steht und welche Hindernisse jedem einzelnen Teammitglied im Weg stehen. Dies wird ausführlich dokumentiert und diskutiert in regelmäßigen Meetings. Diese Meetings werden als Daily Scrum bezeichnet, da sie oftmals sogar täglich stattfinden.<sup>98</sup> Offene Kommunikation und das Teilen von Wissen sind die Grundlage dieses Prinzips. Wichtig ist es dabei, eine einheitliche Grundlage für alle herzustellen. Es sollte jeder unter dem Begriff Erfolgskriterien oder Out of Scope das Gleiche verstehen, um Unklarheiten zu vermeiden und Missverständnissen vorzubeugen. Das Prinzip der periodischen Überprüfung und der Kontrolle auf Funktionalität des Projektoutputs ist ursächlich für einen der Vorteile, die agiles Projektmanagement gegenüber dem traditionellen Projektmanagement hat. In diesem Zusammenhang überprüft das Team, ob die im Scrum verwendeten Vorgehensweisen zum Erreichen des Sprintziels beitragen. Eine Überprüfung ohne Veränderungen wird als unsinnig betrachtet.<sup>99</sup> Aus diesem Prinzip entsteht auch der Bedarf für das letzte Prinzip, die Adaption oder auch Anpassung genannt. Am Ende einer jeden Iteration sollte die Anforderungsliste angepasst werden. Es erfolgt eine Anpassung an veränderte Rahmenbedingungen, um Ziele effizienter zu erreichen. Solche Anpassungen müssen bei Bedarf oder dem Prinzip Überprüfung folgend möglichst schnell und ohne unnötigen zeitlichen Verzug umgesetzt werden.

Als zentraler Bestandteil im Scrum wird ein kleines Scrum-Team gesehen.<sup>100</sup> Das Team hat keine hierarchischen Strukturen oder kann in weitere Teilteams aufgeteilt werden. Alle Mitglieder sind gleichberechtigte Experten, die sich nur auf eine Sache konzentrieren, und zwar das Projektziel. Dabei besitzt das Team das gesamte Wissen, das zur Zielerreichung benötigt wird. Die fehlende Hierarchie führt zu einem wichtigen Vorteil von Scrum, und zwar dass die Teams selbst organisiert sind. Es wird dabei intern entschieden, was, wie, von

<sup>95</sup> vg . Komus and Kuberg 2020, S. 25.

<sup>96</sup> vg . Dechange 2020, S. 290.

<sup>97</sup> vg . Komus and Kuberg 2020, S. 3.

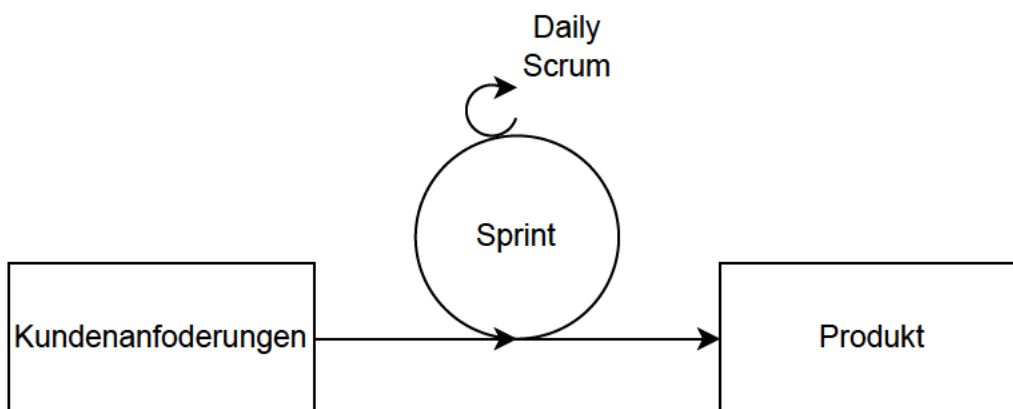
<sup>98</sup> vg . Schwaber and Suther and 2020, S. 10.

<sup>99</sup> vg . Schwaber and Suther and 2020, S. 4; vg . Dechange 2020, S. 290.

<sup>100</sup> vg . Käne 2020, S. 74.

wem und wann erledigt wird. Dadurch kann ein Scrum-Team agil reagieren, um so innerhalb eines Sprints seine Ziele zu erreichen. Ein Sprint ist ein Scrum-Event, in welchem das Projektteam an der Fertigstellung vorher definierter Arbeitspakete und Ziele arbeitet. Am Ende eines jeden Sprints sollte als Output ein MVP, ein Minimal Viable Product, entstehen.<sup>101</sup> Dieses MVP ist ein minimal brauchbares Produkt, das getestet wird durch den Endnutzer und das Feedback fließt in die nächste Sprintrunde ein. Dieses wiederholt sich so lange, bis alle Aufgaben, die im Product Backlog enthalten sind, abgearbeitet wurden. Im Product Backlog sind zu Beginn des Projekts alle Arbeitspakete gelagert, welche im Verlauf des Projekts bearbeitet werden sollen. Im Sprint Backlog werden dann die Arbeitspakete übernommen, die das Team in diesem Sprint bearbeiten will. Die Arbeitspakete für das Sprint Backlog werden aus dem Product Backlog übernommen.<sup>102</sup> In dem Fall, dass der Bedarf zur Bearbeitung von Arbeitspaketen größer ist als die Fähigkeiten des kleinen Scrum-Teams es hergeben, sollte darüber nachgedacht werden, mehrere zusammengehörige Scrum-Teams zu erstellen, um so zeit- und ressourceneffizient weiterzuarbeiten. Der Nachteil wäre, dass hier neue Teammitglieder für das Projekt herangezogen werden müssen.

Wichtige Rollen im Team sind der Scrum Master, der Product Owner und die Developer.<sup>103</sup> Der Scrum Master hat zusammen mit dem Product Owner eine Ergebnisverantwortlichkeit bei der Einführung von Scrum. Diese Ergebnisverantwortlichkeit ist eine organisatorische Verantwortlichkeit, der Scrum Master ist nicht fachlich am Projektoutput beteiligt. Er kümmert sich hingegen darum, dass das Scrum-Team sowie die Organisation die zugrundeliegenden Mechanismen von Scrum kennen. Ein Scrum Master hat die Aufgabe, innerhalb des Scrum-Teams die Mitglieder zu befähigen, ihre Performance zu verbessern. Developer sind diejenigen, die innerhalb des Scrum-Teams den Projektoutput erstellen und dafür verantwortlich sind, dass die Sprintziele erreicht werden. Developer stehen sich gegenseitig als Experten zur Verfügung und versuchen bei Hindernissen, diese mit den anderen Mitgliedern des Scrum-Teams aus dem Weg zu räumen. Die Developer entscheiden unter anderem, welche Ziele im Sprint erreicht werden sollen und auf welche Weise dies geschehen soll. Die Projektziele und Arbeitspakete erhält das Team durch den Product Owner, denn der Product Owner besitzt Kenntnisse darüber, welche Produkt- und Kundenanforderungen vorliegen und er bringt diese auch in regelmäßigen Abständen mit ein.



**Abbildung 7 - Vorgehen Scrum mit einer Iteration<sup>104</sup>**

<sup>101</sup> vg . Peters and Sche ter 2021, S. 142.

<sup>102</sup> vg . Schwaber and Suther and 2020, S. 9.

<sup>103</sup> vg . Schwaber and Suther and 2020, S. 5.

<sup>104</sup> n An ehnung Dechange 2020, S. 291.

Wichtige Bestandteile von Scrum sind die sogenannten Scrum-Artefakte.<sup>105</sup> Die Artefakte im Scrum dienen dem Zweck, Transparenz zu schaffen für Schlüsselinformationen. Dadurch wird im Scrum ermöglicht, eine einheitliche Grundlage herzustellen für die Scrum-Beteiligten, welche Zugriff auf diese Artefakte haben.

Das Product-Backlog-Artefakt wurde bereits angesprochen und dient dem Zweck, den Fortschritt des Produkt-Ziels messbar zu machen. Das bedeutet, den Fortschritt messbar zu machen, inwieweit das Ist-Produkt dem zukünftigen Soll-Zustand des Produktes bereits entspricht. Dies wird erreicht, indem Transparenz und Fokus durch bereitgestellte Informationen verbessert werden. Es beinhaltet eine Liste von Dingen, die zur Produktverbesserung beitragen. Aus diesem Backlog erhält das Scrum-Team die Arbeit, die erledigt werden muss. Das zweite Artefakt im Scrum ist der Sprint Backlog. Analog zum Product Backlog dient dieses zum Erreichen eines spezifischen Zieles, in diesem Fall dem Sprint-Ziel. Das Sprint-Ziel wird während der gemeinsamen Sprintplanung erstellt und dann in das Sprint Backlog abgelegt. Es ist Aufgabe der Developer, während des Sprints das Ziel zu erfüllen. Sollte sich jedoch herausstellen, dass dies aus unbekanntem Gründen nicht in der geforderten Qualität möglich ist, können diese mit dem Product Owner den Umfang des Backlogs neu verhandeln, ohne dabei das Sprint-Ziel zu gefährden. Das letzte wichtige Artefakt ist das Inkrement. In diesem ist die Definition of Done essenziell, denn in der Definition of Done, kurz DoD, werden die Kriterien hinterlegt, die das Produkt erfüllen muss, um als Erfolg oder fertig zu gelten.<sup>106</sup> Dies wird auch zum Messen des Erfolgs genutzt. Der Output eines jeden Inkrements ist ein Teil des Produkt-Ziels und wurde gründlich geprüft. Alle Inkremente bauen aufeinander auf und es wurde sichergestellt, dass sie zusammen funktionieren. Alle Inkremente zusammen ergeben das Produkt-Ziel. Während eines Sprints können ein Inkrement oder mehrere Inkremente abgearbeitet werden. Inkremente werden immer dann erfolgreich gebildet, wenn eine Aufgabe oder ein Arbeitspaket aus dem Product Backlog die DoD erfüllt und somit als abgeschlossen betrachtet werden kann. Es ist möglich, Inkremente schon eher an den Kunden zu liefern, wenn es nötig ist.

Der Ablauf des Scrum-Vorgehens spiegelt seinen iterativen Charakter optimal wider. Wie in Abbildung 7 zu sehen ist, kann das Vorgehen am besten mit einer sich selbst wiederholenden Schleife dargestellt werden. In einem ersten Schritt werden die Kundenanforderungen durch den Product Owner in das Product Backlog eingebracht. Während der Sprintplanung wird durch das Team entschieden, welche Aufgaben aus dem Product Backlog in das Sprint Backlog übernommen werden. Diese werden dann während des Sprints abgearbeitet. Während des Sprints wird ein Daily Scrum durchgeführt. Wie bereits oben erwähnt, ist dies ein kurzes tägliches Meeting, bei dem das Team mit dem Scrum Master zusammenkommt.<sup>107</sup> Während dieses Meetings wird darüber gesprochen, welche Hindernisse es gerade im Team gibt, welche Aufgaben die DoD wie weit erfüllen und welche Aufgaben bis zum nächsten Daily Scrum bearbeitet werden sollen. Am Ende eines jeden Sprints erfolgt das Sprint Review, wo es zur Abnahme der im Sprint entwickelten Inkremente kommt. Hier wird vom Development-Team dem Product Owner gezeigt, welche funktionsfähigen Inkremente während des Sprints erstellt werden konnten. Dadurch können funktionierende Inkremente möglichst frühzeitig eingesetzt werden und einen realen Nutzen erzeugen. Es kommt im weiteren Verlauf auch zur Sprintretrospektive.<sup>108</sup> Hier werden Lessons Learned Feedbacks besprochen und mögliche Verbesserungen für den nächsten Sprint abgeleitet.

---

<sup>105</sup> vg. Schwaber and Suther and 2020, S. 11.

<sup>106</sup> vg. Dechange 2020, S. 298.

<sup>107</sup> vg. Dechange 2020, S. 300.

<sup>108</sup> vg. Dechange 2020, S. 301.

Danach beginnt das Vorgehen erneut in dieser Reihenfolge, bis das Gesamtprojektziel erreicht wurde.

Änderungswünsche lassen sich bei dieser agilen Methode unkompliziert einbauen. Durch das iterative Vorgehen können Änderungswünsche im Product Backlog oder während der Sprintplanung eingebracht und in den entsprechenden Task, die Aufgabe, zu den entsprechenden Arbeitspaketen und Aufgaben aufgenommen werden. Tasks bilden die Grundlage für die Aufgaben und enthalten alle wichtigen Informationen, die benötigt werden, um diese Aufgaben zu erfüllen. Der ständige Austausch mit dem Product Owner und die Dokumentation der DoD dienen der Qualitätssicherung und sie ermöglichen es, den Erfolg zu messen. Die Realisierung erfolgt über das Product Backlog und dem damit verbundenen Produkt-Ziel.

### 3.2.2 Kanban

Eine weitere agile Methodik ist Kanban, das nach der bereits zitierten Studie der Hochschule Koblenz nach Scrum die zweitbeliebteste agile Methodik ist.<sup>109</sup> Die Kanban-Methodik entstammt dem Lean Management Model von Toyota und bedeutet so viel wie Flow (dt. Fluss).<sup>110</sup> Der Name zeigt auf, welche Zielstellung Kanban verfolgt. Es geht darum, einen möglichst gleichmäßigen Arbeitsfortschritt zu ermöglichen, welcher gut zu planen ist. Es gibt laut dem Begründer von Kanban sieben wichtige Voraussetzungen, die erfüllt sein müssen, um Kanban erfolgreich nutzen zu können.<sup>111</sup> Diese wurden ursprünglich für die Produktion bei Toyota erstellt, lassen sich allerdings auf das heutige Projektmanagement anpassen.

1. Fließfertigung – Der Fluss der Aufgaben darf niemals ins Stocken geraten. Erhöhung eines Standardisierungsgrades für die Anforderungen an das Produkt und eine strenge Taktung der Abarbeitung der Aufgaben.
2. Verkleinerung der Losgrößen – Reduzierung von Aufgabenfülle und Reduzierung von Stand-by-Time. Es sollte vermieden werden, Aufgaben völlig zu überladen und dadurch eine Überarbeitung der Ressource zu verursachen.
3. Glättung der Produktion – Glätten von Prozessen. Kanban erstreckt sich über mehrere Stufen und sollte deshalb genau geplant werden, um zu große Schwankungen zu vermeiden. Jede Schwankung würde sich auf die vorherigen Schritte auswirken.
4. Verkürzung und Standardisierung von Transportzyklen – Verkürzung der Zeiten, in denen eine Aufgabe wechselt oder nötige Ressourcen beschafft werden.
5. Kontinuierliche Produktion – Kontinuierliche Aufgabenabarbeitung. Ziel von Kanban ist es, eine kontinuierliche Auslastung sicherzustellen.
6. Adressen bestimmen – Auf einer Kanban-Karte müssen alle Informationen und Ressourcen erfasst sein, damit derjenige, der sie erhält, die Aufgabe erfüllen kann, ohne eine weitere Person fragen zu müssen. Es muss eine eindeutige Zuordnung von Ressourcen und Verarbeitungsstellen ermöglicht werden.
7. Festlegung der Verpackungsform – Die Form, wie das Output weitergegeben wird, wird dadurch bestimmt, welche Anforderungen der nachfolgende Prozess hat.

Kanban im Kontext des Projektmanagements ist eine Methode, um eine große Anzahl von parallelen Aufgaben auf ein überschaubares Maß zu reduzieren. Das Vorgehen im Kanban

---

<sup>109</sup> vg . Komus and Kuberg 2020, S. 52.

<sup>110</sup> vg . Bertagno 2022, S. 84.

<sup>111</sup> vg . Takeda and Meynert 2013, S. 181.

erfolgt nach dem Pull-Prinzip.<sup>112</sup> Das Pull-Prinzip beschreibt die Art, wie eine Aufgabe zugeteilt wird. Diese Form der Aufgabenzuteilung ist dadurch definiert, dass nicht der Vorgänger die Aufgabe freigibt und dem Nachfolger zuschiebt, das wäre das Push-Prinzip, sondern der Nachfolger sich diese selbst vom Vorgänger holt, sobald der Nachfolger bereit ist, diese zu bearbeiten.<sup>113</sup> Durch dieses Vorgehen soll ein störungsfreier Arbeitsfluss realisiert werden.

Kanban in Projektmanagement-Modellen einzubinden, erfolgt immer nach einem ähnlichen Muster. Als Visualisierungstool wird ein sogenanntes Kanban-Board verwendet. Auf dem Kanban-Board, wie in Abbildung 8 dargestellt, werden Aufgaben nach dem Pull-Prinzip in den verschiedenen Phasen des Projektes bewegt. Die Anzahl der Phasen ist projektabhängig, nur die des Backlogs ist fest, in Abbildung 8 durch die Phase To-do dargestellt, und die Erledigt-Phase sollte ebenfalls immer vorhanden sein, um ein höchstmögliches Maß an Transparenz zu ermöglichen. Diese beiden Phasen stehen für noch zu erledigende Aufgaben und die Aufgaben, die bereits erledigt wurden. Die einzelnen Phasen sind in die Kategorien „In Bearbeitung“ und „Fertig“ aufgeteilt, damit die darauffolgende Station weiß, welche Aufgaben sie sich holen kann. Jede Station hat ein Worklimit, das bedeutet, es gibt eine Grenze, wie viele Kanban-Karten in einer Phase und damit Station vorhanden sein dürfen, um eine Überlastung der Station zu vermeiden. Dieses Limit heißt WIP-Limit, Work in Progress Limit, und ermöglicht einen möglichst optimalen Workflow mit kurzen Durchlaufzeiten und hohem Durchsatz von Aufgaben. Wie bereits weiter oben beschrieben, sind kurze Durchlaufzeiten eine wichtige Anforderung an Kanban-Systeme und sie definieren die Zeitspanne, die ein Workitem in Form einer Kanban-Karte vom To-do bis zur Erledigt-Phase benötigt. Der Durchsatz steht für die Anzahl der Workitems, die pro Zeitspanne abgearbeitet werden. Aufgrund der Limitierung von Personal und Ressourcen kommt es immer wieder zu Engpässen. Durch diese Methodik können Engpässe besser identifiziert werden und es kann eine Nachjustierung erfolgen. Im Zusammenhang mit dem WIP-Limit und der Kanban-Karte in der Kategorie „Fertig“ kann visualisiert werden, welche Phasen einen niedrigeren Durchsatz haben und wo ein Engpass entsteht. Zur Analyse von Kanban werden die bereits erwähnte Durchlaufzeit, der Durchsatz, die Wartezeit und der kumulative Fluss herangezogen.<sup>114</sup> Der kumulative Fluss beschreibt die Kanban-Karten pro Phase. Die Wartezeit beschreibt die Zeit, die ein Workitem benötigt, um von der Kategorie „Fertig“ in der vorherigen Station zum Nachfolger weiterzugehen. Je länger die Wartezeiten sind, umso unproduktiver ist das Kanban-System bzw. umso größer ist die Wahrscheinlichkeit, dass es irgendwo ein Problem gibt. In Verbindung mit einer Scrum-Vorgehensweise würde dies im Daily Scrum angesprochen und nach einer Lösung gesucht werden.

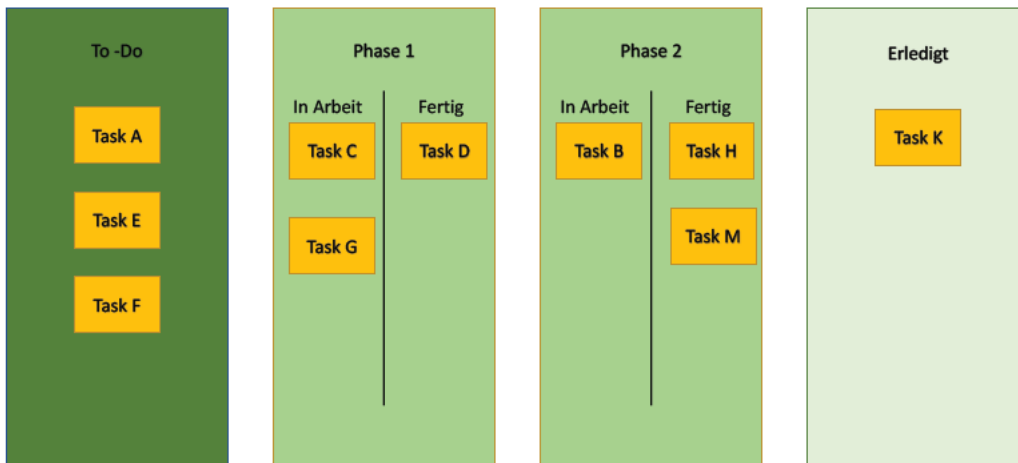
---

<sup>112</sup> vg . Takeda and Meynert 2013, S. 178.

<sup>113</sup> vg . Dechange 2020, S. 306.

<sup>114</sup> vg . Dechange 2020, S. 309.





**Abbildung 8 - Kanban-Board<sup>115</sup>**

Die Vorteile von Kanban zeigen sich primär in der Visualisierung von Aufgaben und der Information darüber, wo Engpässe im Projekt auftauchen. Durch das WIP-Limit wird eine Überlastung der einzelnen Teammitglieder verhindert und die Aufgabenabarbeitung bleibt im Flow. Dadurch kann eine hohe Planungsflexibilität erreicht werden, ohne jedoch darauf verzichten zu müssen, auf geänderte Prioritäten reagieren zu können, da viele Aufgaben parallel erledigt werden können. Ein weiterer wichtiger Vorteil von Kanban ist die gesteigerte Teambeteiligung, weil das Team in seiner Gesamtheit zusammenwirkt und viel diskutiert wird. Bei Problemen bei einer Aufgabe können sich die Teammitglieder aneinander wenden, um so schnell ein Hindernis zu beseitigen. Der iterative Ansatz wird vor allem deutlich beim Bearbeiten von Aufgaben und bei dem Wiederbefüllen des Backlogs. Durch dieses Vorgehen können Änderungen schnell in das Projektteam hineingebracht werden. Ein Nachteil ist hier, dass Kanban in größeren Projekten nicht als eigenständiges Modell bestehen kann und daher häufig dann mit der Scrum-Methodik in Verbindung genutzt wird.<sup>116</sup> Kanban eignet sich nur bei kleinen Projektteams und kurzfristigen Aufgaben, deren Priorität situationsbedingt betrachtet wird. Entgegen einigen Meinungen ist es jedoch trotzdem ein Vorgehensmodell, da, wie in dieser Arbeit bereits definiert, Vorgehensmodelle eine Zusammenstellung von Elementen sind, die auf einem Wert oder mehreren Werten basiert. Ein Vorgehensmodell kann aus strukturierten oder unstrukturierten Elementen bestehen. Kanban ist ein Beispiel für eine unstrukturierte Zusammenstellung von Elementen. Es lassen sich vier Elemente herausarbeiten, die Kanban charakterisieren und damit die Definition als Vorgehensmodell erfüllen.<sup>117</sup>

- Arbeit wird nach dem Pull-Prinzip weitergegeben.
- Die Menge an Arbeit wird durch das WIP limitiert.
- Es herrscht Transparenz hinsichtlich Informationen.
- Es findet eine kontinuierliche Verbesserung der Arbeitsabläufe statt.

Die Eignung für eher kleine Projekte erfolgt aus dem eher unstrukturierten Charakter und dem Fehlen einiger Elemente, wie projektübergreifender Risikobewertung. Vorgehensmodelle können allerdings innerhalb eines anderen Vorgehensmodells parallel, sequenziell

<sup>115</sup> n An ehnung Dechange 2020, S. 307.

<sup>116</sup> vg . Dechange 2020, S. 306.

<sup>117</sup> vg . Epp ng 2011, S. 53.

oder integriert verwendet werden. Beispiele dafür sind die in Abschnitt 3.3 vorgestellten Vorgehensmodelle. Im Kontext des agilen Ansatzes werden jedoch nicht sequenzielles und agiles Vorgehen, sondern zwei agile Ansätze gemischt.

### 3.3 Hybrides Projektmanagement

Unter dem Begriff Hybrides Projektmanagement werden Modelle verstanden, die sich aus agilem und sequenziellem Ansatz zusammensetzen.<sup>118</sup> Es ist nicht festgelegt, wie stark welcher von beiden Ansätzen innerhalb des hybriden Vorgehensmodells vertreten ist, weshalb sich viele verschiedene Modelle entwickeln lassen. Die individuellen Anforderungen an das Projekt gestalten in gleichem Maße die Voraussetzungen dafür, wie diese Methode durchgeführt wird. Aufgrund dessen lassen sich eher Ausschlusskriterien gestalten, warum dieser Ansatz nicht verwendet wird. Bei einem Projekt, das kaum Planungssicherheit benötigt und völlig flexibel gestaltet werden muss, wird man eher einen rein agilen Ansatz verwenden. Im Kontrast dazu werden Projekte, die eine hohe Planungssicherheit benötigen und bei denen der Detailgrad der Planung weitreichend ist, eher auf traditionelle Weise durchgeführt, da das Beimischen agiler Methodiken nicht zum Erfolg des Projektes beitragen würde.

Es werden drei verschiedenen Arten von hybriden Modellen und Ausprägungen unterschieden:

- Verteilt, sequenzielle Ausprägung,
- Verteilt, parallele Ausprägung,
- Integrierte Ausprägung.

Bei der verteilt, sequenziellen Ausprägung werden einzelne Projektphasen agil oder klassisch durchgeführt, dabei wird unter anderem das Entwicklungsprojekt aus Gründen der Planungssicherheit sequenziell nach dem Wasserfallmodell durchgeführt und die Prototypenentwicklung wird aufgrund von benötigter hoher Flexibilität auf agile Weise umgesetzt. Im Kontrast dazu steht die verteilt, parallele Ausprägung. Beide Ansätze, sowohl der traditionelle als auch der agile Ansatz, werden parallel angewendet, je nach Anforderung des Projektes. Bei der verteilt, parallelen Ausprägung können Arbeitspakete je nach Anspruch an das individuelle Arbeitspaket in einer agilen Vorgehensweise angewandt werden und, zeitlich gleich verteilt, bei einem anderen der klassische Ansatz. Beides passiert aber im Unterschied zur verteilten, sequenziellen Ausprägung parallel und nicht nacheinander. Bei der integrierten Ausprägung erfolgt die Anwendung von Vorgehensmodellen innerhalb des Projektes gleichzeitig auf die gleichen Elemente. Es kann beispielsweise in einem nach dem Wasserfallmodell gemanagten Projekt die agile Methodik des Daily Scrums oder Kanban-Boards angewendet werden, ohne dass dies den Grundcharakter des klassischen Projektmanagements verändert. Welches Vorgehensmodell in einem hybriden Ansatz überwiegt, entscheidet sich auf der Basis der Anforderungen des Projektes. Wie bereits in den Kapiteln 3.1 und 3.2 dargelegt, eignen sich diese für verschiedene Projektarten.

Der Vorteil von hybriden Projekten ist, dass der Projektmanager ermächtigt wird, für die individuellen Anforderungen eines Projektes ein passendes Modell zu entwickeln und er dadurch auf Einzelfälle reagieren kann. Diese Stärke spielen hybride Modelle aus, wenn vor Beginn von Projekten noch viele Unklarheiten bestehen, die sich nicht über ein traditionelles Modell planen lassen.<sup>119</sup> Über einen verteilt, sequenziellen Ansatz können in einer

---

<sup>118</sup> vg. Dechange 2020, S. 311.

<sup>119</sup> vg. Käne 2020, S. 76.

ersten Phase durch ein agiles Vorgehen die genauen Anforderungen an den Projektoutput ermittelt werden und in einer zweiten Phase, zum Beispiel, die Realisierung in einem traditionellen Modell durchgeführt werden. Der Vorteil des agilen Projektmanagement: Mit seiner kontinuierlichen Rückkopplung von Feedback und der Validierung von Teilergebnissen in einem traditionellen System sichert es die Qualität des Projektes und damit den bestmöglichen Projektoutput, welche den Wünschen des Projektauftraggebers entsprechen. Damit ist es ist möglich, eine langfristige Planung mittels grober Projektvorgaben durchzuführen sowie die Möglichkeit zu haben, diese Schritt für Schritt zu konkretisieren und an neue Anforderungen an das Projekt anzupassen. Die Verbesserungen innerhalb des Projektteams dürfen in diesem Zusammenhang nicht vernachlässigt werden. Ein gesteigertes Team-Commitment und das eigenverantwortliche, selbstorganisierte Handeln jedes Einzelnen steigern ebenfalls die Motivation im Team und können für dessen persönliche Weiterentwicklung förderlich sein.<sup>120</sup>

Dem gegenüber stehen jedoch Nachteile, die berücksichtigt werden müssen und die Eignung von hybridem Vorgehen für bestimmte Projekte ausschließen. Die Menge an Möglichkeiten, die das hybride Vorgehen beinhaltet, führt zu dem Problem, dass die Projektbeteiligten eine sehr hohe Methodenkompetenz mitbringen müssen. Diese Voraussetzung ist jedoch in der Realität nicht immer gegeben, was zu einem erhöhten Befähigungsaufwand für die Teammitglieder führt. Dieser Punkt, die große Anzahl an Möglichkeiten, kann auch bei den Projektstakeholdern zu Akzeptanzproblemen führen, wenn diese bis jetzt nur nach einem Ansatz gehandelt haben und eine Veränderung des altbekannten Vorgehens unerwünscht ist. Innerhalb des Projektteams kann es in diesem Zusammenhang ebenfalls zu Problemen kommen, weil eine typische Top-down-Führungsstruktur mit einem selbstverantwortlichen Handeln gemischt wird. Ist dieses aber nicht verbreitet in der Organisation, kann es zu Irritationen im Projektteam kommen. Für Fälle, in denen sich die Anwendung einer reinen Vorgehensweise nach klassischem oder agilem Modell empfiehlt oder das Beimischen von Methodiken keinen Mehrwert bringt, ist es nicht zielführend, ein hybrides Vorgehen anzuwenden.

### 3.4 Change Projektmanagement

Einen anderen Fokus auf das Projektmanagement legt das veränderungsgetriebene Vorgehen, das auch als Change Projektmanagement bezeichnet wird.<sup>121</sup> Das Change Projektmanagement grenzt sich dadurch vom Changemanagement innerhalb der agilen und traditionellen Ansätze ab, dass es eine andere Zielstellung verfolgt. Changemanagement in agilen und traditionellen Ansätzen soll dahingehend für einen geregelten Ablauf sorgen, wie mit Änderungen innerhalb des Projekts umgegangen wird. Das Change Projektmanagement erfüllt jedoch einen anderen Zweck. Bei der Erlangung eines besseren Verständnisses, was diese Kategorie der Vorgehensmodelle besonders macht, sollte der Satz „Der Weg ist das Ziel“ als Grundlage dienen.<sup>122</sup> Unter dieser Art von Projekten lassen sich Vorhaben zusammenfassen, bei welchen Änderungen elementarer Art weitreichende Auswirkungen auf das Projekt haben und bei denen der Projektoutput über Grenzen von Bereichen wirkt. Es steht also nicht das Ziel im Vordergrund, sondern der Zielerreichungsprozess. Es geht um die Steuerung dieses Prozesses, um ein bestmögliches Ergebnis zu

---

<sup>120</sup> vg. Käne 2020, S. 77.

<sup>121</sup> vg. Kuster et al. 2011, S. 30.

<sup>122</sup> vg. Lauer 2019, S. 5.

erhalten. Bei dieser Art von Projekten wird die angestrebte Änderung durch die Führung, Organisationseinheit oder Personengruppe herbeigeführt. Eine wichtige Voraussetzung für eine veränderungsgetriebene Vorgehensweise ist das Bewusstsein innerhalb des zu verändernden Bereichs. Wie bereits erwähnt, liegt der Fokus bei dieser Vorgehensweise nicht auf einem Produkt oder einer Dienstleistung, also was am Ende des Projekts steht, sondern auf dem Kreieren von Veränderungsprozessen, welche eine Auswirkung auf den Menschen direkt haben.<sup>123</sup> Eine scharfe Trennung zwischen den verschiedenen vorher genannten Methodiken ist daher nicht möglich. Es kommt bei der Umsetzung häufig zu Vermischungen der Vorgehensweisen. Beim Change Projektmanagement liegt der Zielfokus auf den Menschen und das grenzt diese Methode von den anderen Methodiken ab.<sup>124</sup> Der Bedarf für Change Projektmanagement kann innerhalb einer Organisation oder eines Projekts von verschiedenen Punkten abhängig sein, unter anderem von den folgenden:<sup>125</sup>

- Individuen bilden eine wichtige Grundlage für Organisationen. Oftmals die kleinste Einheit, sind sie von entscheidender Bedeutung für den Erfolg von Veränderungen in Organisationen. Dabei ist nicht nur das richtige Skillset ein wichtiger Faktor, sondern auch die mentale Einstellung.
- Organisationsstruktur beinhaltet Ablauf- und Aufbauorganisation sowie Ressourcenmanagement und Strategie der Organisation. Diese Struktur kann unter der Problematik leiden, dass im Verlauf der Zeit diese Struktur nicht an neue Gegebenheiten angepasst wird oder sie durch eine zu offene Beschreibung nicht mehr ordentlich ausgeführt wird.
- Die Organisationskultur kann ein starker Treiber für Wandel und Veränderung in einer Organisation sein. Diese Kultur entwickelt sich aus den Individuen der Organisation heraus und hat dadurch einen großen Effekt auf die Zielerreichung, da diese von den Organisationsmitgliedern gestaltet wird.

Ein funktionierendes Risikomanagement ist bei einem Change-Projekt besonders wichtig, da die angestrebte Leistung durch die zu verändernde Organisationseinheit und deren Mitarbeiter selbst erbracht wird.<sup>126</sup> Überholte Vorgehensweisen, die Veränderung bedürfen, weil diese nicht mehr dem aktuellen Stand entsprechen, müssen verändert werden. Dieser Prozess verursacht oft Gegenkräfte wie Ängste und Widerstände. In diesem Zusammenhang fallen oft Sätze wie „Das haben wir schon immer so gemacht“ oder „Das machen wir hier nicht so“.<sup>127</sup> Wie bereits erwähnt, ist eine wichtige Voraussetzung für ein Veränderungsprojekt ein Bewusstsein, dass die Veränderung nötig ist. Aus diesem Grund muss auf die wichtigen Erfolgsfaktoren für Veränderungsprojekte geachtet werden.<sup>128</sup> Diese sind entscheidend für den erfolgreichen Verlauf des Projekts.

- Die richtige Führung während des Wandels,
- Positive Zielsetzungen,
- Kommunikation,
- Partizipation der Betroffenen,
- Stärkung des Zusammenhalts durch Integration von Projektbeteiligten in Projektteams,
- Personelle Entwicklung,
- Die richtige Organisation des komplexen Projekts,

---

<sup>123</sup> vg . Dechange 2020, S. 206.

<sup>124</sup> vg . Lauer 2019, S. 212.

<sup>125</sup> vg . Lauer 2019, S. 7.

<sup>126</sup> vg . Kuster et a . 2011, S. 30.

<sup>127</sup> He mann 2014.

<sup>128</sup> vg . Lauer 2019, S. 87/ 98/ 113/ 125/ 153/ 173/ 185/ 199/ 217/ 231; vg . Se fer e n 2022, S. 9.

- Ein gezielter Einsatz von äußeren Beratern und das Aufrechterhalten einer Veränderung, um zu verhindern, dass die zu beseitigende Situation erneut eintritt.

In diesem Zusammenhang durchläuft eine Organisation verschiedene Phasen der Veränderung. Als Beispiel für ein solches Modell sei das 3-Phasenmodell genannt, welches die Phasen Auftauen, Verändern und Einfrieren beschreibt.<sup>129</sup> Ursprünglich war dieses Modell für soziale Veränderungen in einer Gesellschaft gedacht und lässt sich auch auf Veränderungsprojekte übertragen. Die Phasen beschreiben das Vorgehen, dass der zu verändernde Gegenstand erst aufgetaut wird, um ihn verändern zu können, danach erfolgt die Veränderung und im letzten Teil wird der geänderte Gegenstand wieder eingefroren. Diese grobe Beschreibung der Phasen, die eine Organisation während des Veränderungsprozesses durchläuft, lässt sich weiter aufgliedern, um ein erhöhtes Adaptionspotenzial für die Praxis zu erreichen.<sup>130</sup>

### 3.5 Zusammenfassung Kapitel Vorgehensmodelle

Die in diesem Kapitel vorgestellten Vorgehensmodelle weisen auf wichtige Kriterien hin, die ein Vorgehensmodell erfüllen sollte, um innerhalb einer Organisation erfolgreich für die Durchführung eines Projektes benutzt werden zu können. Für alle Vorgehensmodelle ist es wichtig, dass die spezifischen Voraussetzungen für das Modell bekannt sind, unter denen es genutzt werden kann.

Im Falle der traditionellen Modelle sind die Anforderungen an das Projekt, dass der Projektumfang vordefiniert und eine hohe Planbarkeit hergestellt werden sollte. Feste Termine, das Einhalten des Budgets und ein möglichst gleichbleibender Projektumfang sind wichtige Voraussetzungen für ein Projekt, das auf diese Weise durchgeführt wird. Dadurch können sehr komplizierte Projekte innerhalb einer Organisation durchgeführt werden und es wird eine Sicherheit hergestellt für den Projektauftraggeber. Es erfolgt eine ausführliche Dokumentation der Projektbestandteile und für den Projektoutput. Wichtige Rollen im Projekt sind der Projektleiter und die Teammitglieder. Der Projektauftraggeber gibt am Anfang des Projekts einmal seine Vorgaben in das Projekt und erhält am Ende einen fertigen Projektoutput. Wichtige Rollen außerhalb des Projekts sind Leiter von Organisationseinheiten, von denen Ressourcen benötigt werden für das Projekt. Für den Erfolg des Projekts ist eine klare Projektstruktur erforderlich, in der Kompetenzen und Hierarchien geklärt sind. Der Aufbau und die Durchführung des Projekts erfolgen in Phasen, deren Anforderungen und Output geplant werden. Änderungen am Projektumfang während des Projektablaufs können nur umständlich eingearbeitet werden und führen meistens zu einer Erhöhung der Kosten und/ oder des Projektzeitplanes. Der Erfolg des Projekts wird am Ende gemessen anhand klarer Kriterien, die bereits am Anfang des Projekts unter anderem im Projektsteckbrief festgelegt werden.

Bei den agilen Methodiken hingegen ist der Anspruch eher auf eine möglichst flexible Arbeitsweise ausgerichtet. Ein Projekt, das nach diesem Modell bearbeitet wird, hat in den meisten Fällen nur sehr grobe Anforderungen und die Details werden erst während des Projektes ausgearbeitet. Das Team muss selbstständig und selbstverantwortlich arbeiten und Änderungswünsche werden nach Möglichkeit umgesetzt. Eine Validierung der Zwischenergebnisse, um einen stetigen Projektfortschritt zu messen, ist durch die Organisation gewünscht, um dadurch neuen Input sowie geänderte Anforderungen in das Projekt zu

---

<sup>129</sup> vgl. Lew n 1947, S. 34 ff.

<sup>130</sup> vgl. Dräger 2020, S. 25.

bringen. Prozesse und Werkzeuge sind nicht so wichtig wie Individuen und Interaktionen. Dokumentation ist nachrangig gegenüber einem funktionierenden Output oder Zwischenergebnis. Die Zusammenarbeit mit dem Projektauftraggeber und dem Endnutzer des Outputs ist wichtiger als Vertragsverhandlungen, und die Möglichkeit, auf geänderte Anforderungen zu reagieren, ist wichtiger, als einem strikten Plan zu folgen. Wichtige Rollen sind in diesem Zusammenhang der Projektleiter, Endnutzer des Projektoutputs, der Projektauftraggeber sowie das gesamte Entwicklungsteam. Der Projektleiter fungiert als Ergebnisverantwortlicher, jedoch liegt die Entscheidungskompetenz für einzelne Aspekte beim gesamten Projektteam. Ein agiles Team folgt keiner klaren hierarchischen Struktur. Jedes Mitglied des Projektes hat seine eigenen Verantwortungsbereiche und es wird durch diese selbst und durch das gesamte Team entschieden, wie das weitere Vorgehen ist. Hindernisse werden gemeinschaftlich beseitigt. Aufgrund dieser Organisation ist ein kleines, agiles Team erforderlich. Die Durchführung erfolgt in mehreren Iterationen und es wird, wie bereits erwähnt, durch das Team entschieden, welche Arbeitspakete bearbeitet werden. Als Iteration wird das mehrmalige Durchlaufen eines Prozesses bezeichnet. Jeder einzelne Durchgang entspricht einer Iteration. Ein Projekt benötigt so viele Iterationen, wie es dauert, alle Arbeitspakete fertigzustellen, bis der Projektoutput die Wünsche des Endnutzers erfüllt. Änderungswünsche werden durch den iterativen Charakter zeitnah in die Planung eingebracht und entsprechend ihrer Priorität bearbeitet. Der Erfolg eines Projektes wird daran gemessen, wie die Zufriedenheit des Endnutzers mit dem Projektoutput ist.

Hybrides Vorgehen ist eine Mischung aus dem traditionellen und dem agilen Ansatz. Bei diesem Vorgehen richtet sich die Stärke der Ausprägung eines jeden Ansatzes innerhalb des Vorgehensmodells nach den Anforderungen an das Projekt. Wenn die Planungssicherheit eine Anforderung des Projekts ist, aber die Flexibilität, auf Änderungen im Projekt zu reagieren, höher gewichtet wird, wird der agile Ansatz einen viel stärkeren Teil des Charakters des Vorgehens ausmachen. Dies ermöglicht der Vorgehensweise ein sehr individuelles Setting, Projekte zu gestalten und dadurch den größtmöglichen Nutzen aus dem agilen und traditionellen Ansatz zu gewinnen. Es wird daher nicht angewendet in Fällen, bei denen die Anforderungen an das Projekt einen reinen Ansatz zulassen und die Beimischung eines weiteren Ansatzes keinen Mehrwert für das Projekt bringt. Die wichtigen Rollen richten sich danach, welche Methodiken im Projekt verwendet werden. Kommt es etwa im Projekt zur Anwendung der agilen Methodik Scrum, werden Scrum Master und Product Owner zwingend erforderlich. Bei einem Schwerpunkt auf dem traditionellen Ansatz wird der Projektleiter sowie das Projektteam benötigt. Das Vorgehen innerhalb des Projektes richtet sich danach, welcher Ansatz überwiegt und wie diese Ansätze im Projekt verwendet werden. Es wird hier von drei Ausprägungen gesprochen. Dem verteilt, sequenziellen Ansatz, bei dem die Mischung der Ansätze in verschiedenen Projektphasen stattfindet, die in einer zeitlichen Abfolge nacheinander gegliedert sind. Als weitere Ausprägung gibt es die verteilt, parallele Ausprägung, bei der die verschiedenen Methodiken parallel in einer Phase an verschiedenen Arbeitspaketen durchgeführt werden. Als letzte Ausprägung wird die integrierte Ausprägung gesehen. Bei dieser Ausprägung werden im gleichen Arbeitspaket die traditionellen und agilen Methodiken verwendet. Die Vorteile dieser Methodik ergeben sich aus ihrem hohen Maß an Anpassungsfähigkeit an verschiedene Projekte und den Vorteilen der agilen sowie der traditionellen Vorgehensweisen. Es ist möglich, ein gewisses Maß an Planungssicherheit zu erzeugen, ohne dabei die Flexibilität des agilen Ansatzes einzubüßen. Als Nachteil ist dabei zu nennen, dass diese Modelle meist wenig erprobt sind, weil sie immer wieder neu zusammengestellt werden können und es daher zu Akzeptanzproblemen innerhalb der Organisation kommen kann. Des Weiteren wird ein hohes Maß an Methodenkompetenz von den Projektbeteiligten verlangt, da es eine riesige Zahl an Methoden gibt, die

im Projekt verwendet werden können. Der Erfolg wird anhand des Kundenfeedbacks gemessen und bei den Zwischenvalidierungsschritten, die durch den agilen Ansatz eingebracht werden.

Change Projektmanagement ist kein neues Vorgehensmodell in dem Sinne, dass ein neues Vorgehen in Kontrast zu den traditionellen oder agilen Ansätzen erfolgt. Bei diesem Ansatz steht der Projektoutput nicht im Fokus, sondern der Prozess zum Output. Es werden Änderungen elementarer Art innerhalb von Organisationen herbeigeführt, die über Bereichsgrenzen hinweg wirken und von den zu verändernden Organisationsteilen selbst initiiert werden. Der Fokus liegt hier auf den Menschen selbst und auf der Veränderung eines bestehenden Prozesses oder Verfahrens. Der Fokus liegt hier auf den Menschen selbst und auf der Veränderung eines bestehenden Prozesses oder Vorgehensweise. Dies erfolgt häufig über verschiedene Phasen, die sich im Kern aber auf drei idealtypische eingrenzen lassen. Diese Phasen heißen Auftauen, Verändern und Einfrieren, die sich jedoch in anderen Modellen weiter aufgliedern lassen, um eine bessere Anwendbarkeit in der Praxis zu erreichen. Change-Projekte sind im Kontext dieser Arbeit wichtig, da die Einführung von No-Code-/Low-Code-Plattformen vordergründig über die Menschen funktioniert und dort vorhandene Strukturen aufgebrochen werden und bedingten Veränderungen unterliegen. Changemanagement geht von drei Grundannahmen aus, von denen aus es beginnt: Individuen bilden die Grundlage einer Organisation, Organisationen beinhalten Ablauf- und Aufbauorganisationen und die Organisationskultur ist ein starker Treiber für den Wandel. Aufgrund der Tatsache, dass Change von Menschen ausgeht, muss ein funktionierendes Risikomanagement etabliert sein, da eine Veränderung der Organisation selbst herbeigeführt werden soll und es viel Potenzial für Fehler geben kann. Zum Erfolg eines solchen Projektes können daher verschiedene Faktoren beitragen, wie Wandel in der Führung, positive Zielsetzung, Kommunikation in der Organisation, Beteiligung aller Stakeholder, Stärkung des Zusammenhalts, personelle Entwicklung, Organisation des Projektes und gezielter Einsatz externer Berater.

Diese Vorgehensmodelle dienen als Grundlage für das Ziel dieser Arbeit, einen Leitfaden für die Implementierung von NCLC-Plattformen zu erstellen.

## 4 Forschungsfragen und Ablauf

In diesem Kapitel wird die verwendete Methodik aus der qualitativen Sozialforschung im Detail vorgestellt, um den Nachweis zu erbringen, dass die gewählte Datenerhebungsmethode in Form eines Experteninterviews und die darauffolgende Auswertung der Daten ein möglichst optimales Ergebnis in Bezug auf die Beantwortung der Forschungsfragen möglich macht. Der Begriff Methodik beschreibt dabei den Prozess, der zur Erreichung des Forschungsziels durchlaufen wird.<sup>131</sup> Zu diesem Zweck wird das Experteninterview in der Forschung eingeordnet und die theoretischen Hintergründe dazu werden erläutert. Es wird eine Zielgruppe definiert und erläutert, wie sich die Stichprobe der Interviewpartner zusammensetzt. Im Anschluss an die wissenschaftliche Einordnung werden die Stichprobenauswahl und die methodische Erstellung des Interviewleitfadens aufgezeigt.

### 4.1 Das Interview als Teil der Forschung

In der empirischen Forschung wird in qualitative und quantitative Forschung unterschieden.<sup>132</sup> Die quantitative Forschung lässt sich aus den Begriffen herleiten und bedeutet, dass die komplexe Information mithilfe von mathematisch-statistischen Verfahren auf ihre wesentlichen Informationen reduziert wird.<sup>133</sup> Es erfolgt eine Zuordnung zu numerischen Werten, wohingegen, der Definition von Helfferich folgend, die qualitative Forschung den Sinn oder die subjektive Sichtweise rekonstruieren will. Es werden sprachliche oder schriftliche Texte ausgewertet werden, um nicht quantifizierte Ergebnisse zu erhalten und weiterzuarbeiten. Daraus leitet sich der Anspruch ab, einen Textinhalt, der nicht durch Messbarkeit standardisierbar ist, zu standardisieren und dadurch vergleichbare Ergebnisse zu erhalten. In diesem Kontext wird das wissenschaftliche Interview als Datenerhebungsmethode verwendet.<sup>134</sup> Das wissenschaftliche Interview wird verwendet, wenn nach Aspekten des individuellen und subjektiven Erlebnisses gesucht wird, die nicht direkt beobachtbar sind und was es gegenüber einer Beobachtungsmethode abgrenzt. Eine Abgrenzung zu schriftlichen Methodiken wie dem Fragebogen erfolgt dadurch, dass durch das Interview Rückfragen ermöglicht werden und dadurch, auf die Antwort reagierend, flexible Wissensvertiefung erfolgen kann. Aus diesen Abhängigkeiten entwickelt sich die Wichtigkeit des Interviews für diese Arbeit. Es wird nach den individuellen subjektiven Eindrücken und dem Wissen der Experten gesucht und es wird die Möglichkeit von Rückfragen benötigt, da die Antworten nicht vorhersagbar sind. Durch diese tiefgreifende Evaluierung von Expertenwissen können weitgehende Daten erhoben werden, welche in der Inhaltsanalyse nach Mayring weiterverarbeitet werden können.

Experteninterviews können nach drei Strukturmerkmalen klassifiziert werden.<sup>135</sup> Diese sind das vollstrukturierte Interview, das semi- oder teilstrukturierte Interview und das unstrukturierte Interview. Das Strukturmerkmal bezieht sich in diesem Zusammenhang auf die Form des Interviews und die Standardisierbarkeit der verwendeten Interviewtools.

---

<sup>131</sup> vg. St er 1999, S. 1.

<sup>132</sup> vg. He ffer ch 2011, S. 21.

<sup>133</sup> vg. Ra the 2008, S. 7.

<sup>134</sup> vg. Dör ng and Bortz 2016, S. 356.

<sup>135</sup> vg. Dör ng and Bortz 2016, S. 358; vg. He ffer ch 2011, S. 35; vg. St er 1999b, S. 184.



- Ein unstrukturiertes Interview orientiert sich bei der Durchführung an keinem vorher festgelegten Interviewtool. Die Befragten äußern sich völlig frei bzw. es werden spontan offene Fragen gestellt, die sich aus der jeweiligen Aussage des Befragten ergeben. Diese Strukturform von Interviews ist nicht standardisierbar, weil diese stark von der jeweiligen Situation und Aussage abhängig ist.
- Semi- oder teilstrukturierte Interviews basieren auf einem groben Leitfaden, der jedoch kontextbasierte Zwischenfragen zulässt. Der Interviewleitfaden enthält offene Fragen und gibt eine strukturelle Reihenfolge vor, wie diese gestellt werden. Je nach Interviewsituation ist es möglich, diesen Leitfaden individuell anzupassen und auf Ereignisse zu reagieren, indem Fragen übersprungen oder weiter vertieft werden.
- Ein vollstrukturiertes Interview folgt einem stringenten Interviewleitfaden, indem die Fragen und der Wortlaut vorgegeben sind. Es ist nicht möglich, bestimmte Fragen zu überspringen oder weitere Nachfragen zu stellen. Durch diesen Aspekt ist diese Interviewstruktur leicht standardisierbar. Die Fragen sind hierbei nicht offene Fragen, sondern geschlossene, weshalb sich diese Struktur am besten für Fragebögen eignet.

Für die Beantwortung der Forschungsfragen eignet sich in diesem Zusammenhang das semi- oder teilstrukturierte Interview am besten, da es nicht vorhersehbar ist, welche Antworten die Befragten geben werden und dadurch eventuelle Rückfragen oder das Überspringen von Fragen nötig werden können. Zum Zwecke der gezielten Evaluierung von Informationen ist allerdings eine gewisse Grundstruktur nötig, diese wird über einen Interviewleitfaden umgesetzt. Des Weiteren ist die Interviewumgebung zu betrachten, in der das Interview stattfindet. Diese Interviewart wird auch als Leitfadeninterview bezeichnet.<sup>136</sup> Hierbei kann unterschieden werden zwischen einer persönlichen, telefonischen, chatbasierten oder Onlinebefragung. Vor allem in der qualitativen Forschung wird der persönliche Kontakt bevorzugt, weil es eine persönliche und vertrauensvolle Umgebung ermöglicht.<sup>137</sup> Das telefonische Interview ist vorwiegend mit Blick auf die Kostenfrage sehr vorteilhaft, bringt allerdings genauso wie das chatbasierte Interview die Gefahr einer gewissen Entfremdung zwischen den Interviewpartnern mit sich. Die Vorteile des persönlichen Interviews liegen darin, dass es in einer Face-to-Face-Umgebung stattfindet und eine vertrauensvollere Beziehung zwischen den Interviewpartnern zulässt. Dem kann jedoch ein erhöhter Kostenaufwand gegenüberstehen, hauptsächlich bei weiten Reisen zum Interviewpartner und der Buchung für Räumlichkeiten. Das Videochat-basierte Verfahren bietet einen, für diese Arbeit, guten Kompromiss zwischen der Kosteneffizienz des Telefonverfahrens und dem persönlichen Kontakt zwischen den Interviewpartnern, da es ermöglicht, sich gegenseitig zu sehen, ohne zwingend besondere Räumlichkeiten zu buchen oder an einen bestimmten Ort zu reisen. Die Art des Interviews unterscheidet sich in verschiedene Kategorien. Die Einordnung der Interviewform erfolgt in der Literatur verschiedentlich. Döhring & Bortz folgend werden die Interviewpartner dabei nach Laien-, Betroffenen- und Experteninterviews eingeordnet.<sup>138</sup> Die nächste Interviewart wird geführt, um Wissen oder Expertenwissen zu einem spezifischen Problem zu erlangen, weshalb diese Interviewform in dieser Arbeit Anwendung findet. Diese Art des Interviews ist das problemzentrierte Interview, wie es zum Beispiel Witzel beschreibt.<sup>139</sup> Zu berücksichtigen ist in diesem Fall allerdings, dass problemzentrierte Interviews und Experteninterviews sich sehr ähnlich sind, da beide durchaus Expertenwissen erfassen können. Das Experteninterview dient jedoch eher der

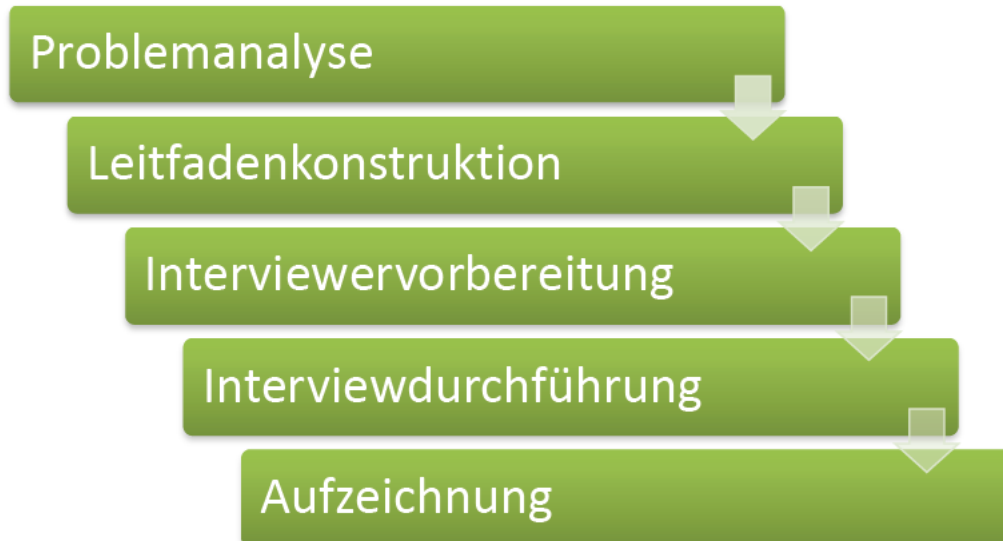
<sup>136</sup> vg . Baur and B as us 2014, S. 559.

<sup>137</sup> vg . Dör ng and Bortz 2016, S. 359.

<sup>138</sup> vg . Dör ng and Bortz 2016, S. 360.

<sup>139</sup> vg . W tze and Re ter 2021, S. 2; vg . Baur and B as us 2014, S. 39; vg . Mayr ng 2016, S. 69.

Überprüfung einer bestehenden Hypothese, wohingegen das problemzentrierte Interview eher die Wahrnehmung und Erfahrung zu einem Problemkomplex herausstellt. Durch die Fokussierung auf ein spezifisches Problem, in dieser Arbeit der Ermittlung eines Vorgehensmodells zur Einführung von Low Code, kann innerhalb der Interviews eine stärkere Eingrenzung vorgenommen werden.



**Abbildung 9 - Ablaufmodell problemzentriertes Interview<sup>140</sup>**

Die Zielgruppe dieser Arbeit sind deutsche Unternehmen, die eine Mitarbeiteranzahl haben von über 250 Arbeitnehmern, da diese nicht als Kleinunternehmen gelten.<sup>141</sup> Bei Großunternehmen ist der finanzielle Hintergrund wesentlich stärker, was für die Einführung von Low Code als begünstigend gewertet wird.<sup>142</sup>

## 4.2 Aufbau Leitfaden für Interview

Der Fragebogen folgt mehreren Frageebenen, die sich auf 3 Phasen verteilen. Diese Phasen sind farblich im Interviewleitfaden hervorgehoben.<sup>143</sup> Die 1. Phase beschäftigt sich mit dem Wissen des Experten zum Thema No Code/Low Code und soll der Beantwortung der Frage dienen, welche Faktoren für Unternehmen wichtig sind, bevor diese NCLC-Plattformen einführen. Diese Phase ist im Interviewleitfaden blau markiert. Die 2. Phase dient der Erlangung von Wissen zu Projektmanagement und Vorgehensmodellen. Durch die Antworten aus diesem Fragenblock soll im späteren Verlauf der Leitfaden für das Einführungsprojekt beschrieben werden. Die 2. Phase ist im Interviewleitfaden gelb markiert. Die 3. Kategorie von Fragen ist der Interviewabschluss und soll Messmöglichkeiten für Qualität und Erfolg des Projektes enthalten. Dieser ist wichtig zur Ermittlung der Nachbereitung des Implementierungsprojektes sowie zur Qualitätsmessung während des Projektes. Der Farbcode im Interviewleitfaden ist rot.

Die Fragen wurden so gestaltet, dass sie eine offene Antwort ermöglichen, dabei folgen die Fragen jedoch einem Leitfaden, der an einigen Stellen Abzweigungen beinhaltet, um je

<sup>140</sup> vgl. Wtze and Reiter 2021, S. 2 ff.

<sup>141</sup> vgl. Europäische Union 2003, S. 4.

<sup>142</sup> vgl. Destatis 2022.

<sup>143</sup> Anlage 1, A-I

nach Antwort des Interviewten reagieren zu können. Sollten während der Beantwortung der Hauptfragen des Interviewleitfadens Unklarheiten auftreten oder der Bedarf nach einer detaillierteren Antwort auftauchen, ist es dem Interviewer möglich, kontextbasiert weitere Fragen zu stellen, die nicht im Leitfaden vorhanden waren. Dadurch soll erreicht werden, dass möglichst viele Informationen von den Experten evaluiert werden können, um eine möglichst große Datengrundlage zu erhalten.

Der Leitfaden des Interviews wurde so aufgebaut, dass eine flexible Reaktion des Interviewers auf die Antwort des Interviewten möglich ist, weshalb diesem eine Semistruktur zugrunde liegt. Dies bedeutet, dass abhängig davon, wie der Befragte eine Frage beantwortet und wie weit er dabei vom Leitfaden abweicht, es dem Interviewer durch Zwischenfragen ermöglicht wird, das Interview in gewissem Maße zu steuern.

Der Leitfaden des Interviews wurde zur Visualisierung wie in einem Baumdiagramm dargestellt, wie es in Anlage 1 zu sehen ist.<sup>144</sup> Der Fokus der Fragen liegt dabei auf den Erfahrungen der Experten. Aufgrund der zeitlichen Beschränkung war es nicht möglich, über ein Trichtermodell induktiv vorzugehen, und so wurden gezielte Fragen sowie Unterfragen gestellt, um das Interview zu steuern, was dieses beeinflusst haben kann.

### 4.3 Auswertung Interviews

In diesem Abschnitt wird kurz die Auswertemethodik erläutert, nach der die Interviews ausgewertet werden. Eine ausführliche Beschreibung findet in Kapitel 5 statt. Bei der Auswertung der Interviews wird auf die computergestützte qualitative Inhaltsanalyse nach Kuckartz zurückgegriffen.<sup>145</sup> Im Speziellen soll hier die inhaltlich strukturierende Analyse vorgenommen werden. Diese Methodik strebt das Ziel an, einen Text inhaltlich so zu strukturieren, dass dadurch die essenziellen Inhalte sichtbar gemacht werden können.<sup>146</sup> Eine qualitative Inhaltsanalyse bietet sich als Methode an, weil diese für das hier durchgeführte Experteninterview als optimale Lösung angesehen wird, wie Bogner in seiner Arbeit herausstellt.<sup>147</sup> Der Zweck des Interviews ist eine Auswertung zu Informationszwecken. Die Inhaltsanalyse fokussiert sich bei dieser Art des Interviews auf die Informationsgewinnung durch Experten. Als Methodik wurde die von Kuckartz vor der ebenfalls weit verbreiteten Methodik nach Mayring gewählt, da die strukturierende Analyse nach Kuckartz praxisorientierter und ausführlicher beschrieben wurde, als dies Mayring tut, der in seiner Schrift den Fokus auf die zusammenfassende qualitative Inhaltsanalyse legt.<sup>148</sup> Der computergestützte Anteil der Analyse wird durch das Programm MAXQDA realisiert.

---

<sup>144</sup> Anlage 1, A-I

<sup>145</sup> vgl. Kuckartz and Rädker 2022, S. 129–156.

<sup>146</sup> vgl. Kuckartz and Rädker 2022, S. 104.

<sup>147</sup> vgl. Bogner, Lüttg, and Menz 2014, S. 72.

<sup>148</sup> vgl. Mayring 2015, S. 69–90.

## 5 Vorstellung Forschungsmethodik

### 5.1 Theoretische Methodik

#### 5.1.1 Qualitative vs. Quantitative Forschung

In der vorliegenden Arbeit wird die Forschungsmethode der qualitativen Inhaltsanalyse nach Kuckartz angewendet.<sup>149</sup> Diese Methodik basiert auf der qualitativen Inhaltsanalyse von Mayring und entwickelt diese weiter.<sup>150</sup> Der Methode nach Kuckartz wurde der Vorzug gegeben, da in seiner Arbeit die inhaltlich strukturierte Analyse detaillierter beschrieben und ausgearbeitet ist, als es bei Mayring der Fall ist.<sup>151</sup> Die Entscheidung für diese Forschungsmethode der qualitativen Inhaltsanalyse basiert auf den Vorteilen, welche die qualitative Forschung gegenüber der quantitativen Forschung besitzt. Die qualitative Forschung unterscheidet sich dadurch von der quantitativen Forschung, dass sie versucht, die subjektive Sichtweise des Untersuchungsgegenstandes, wie gesprochenes Material, zu beleuchten.<sup>152</sup> Es ist in dieser Arbeit nicht möglich, den Forschungsgegenstand über eine Berechnung oder Messung zu erfassen und so durch ein standardisiertes Verfahren wissenschaftlich verwertbare Daten zu evaluieren. Dies betrifft insbesondere subjektiv wahrgenommene Fragen wie: „Finden Sie die Bilder schön, die vom Maler XY gemalt wurden?“ Bei dieser Art der Fragestellung besteht das Problem, dass angenommen wird, dass beide Parteien, der Interviewer und der Interviewte, die gleiche Definition des Wortes ‚schön‘ als Bewertungsgrundlage haben. Das Wort ‚schön‘ ist jedoch ohne vorherige Definition eine subjektive Sichtweise und kann sich, je nach Person und deren persönlichem Hintergrund, unterscheiden.<sup>153</sup> In der qualitativen Forschung wird dieser Sachverhalt berücksichtigt. Im Weiteren wird berücksichtigt, dass Deutung oder Sinn nicht in jedem Fall objektiv sind, sondern in Bezug auf den persönlichen Hintergrund der Menschen entstehen. Bei der Anwendung des qualitativen Forschungsansatzes werden Gegenstände, Zusammenhänge und Prozesse nicht strikt analysiert, sondern der Anspruch ist es, sich in den Forschungsgegenstand hineinzuversetzen und die subjektive Sichtweise nachzuvollziehen.<sup>154</sup> Es ist die Aufgabe der qualitativen Forschung, durch empirische Untersuchungsmethodiken aus Theorien in der Realität allgemeingültige Theorien zu machen.<sup>155</sup> Hierbei werden verschiedene Aufgabenfelder für die qualitative Forschungsanalyse definiert:

- Hypothesenbildung und Theoriebildung,
- Pilotstudien durchführen,
- vorherige Studien und Forschung vertiefen,
- Einzelfallstudien wie Fallanalysen und Prozessanalysen,
- Klassifizierung von Datenmaterial nach empirischen und theoretischen Ordnungsgesichtspunkten sowie Theorie- und Hypothesenprüfung,

---

<sup>149</sup> vg. Kuckartz and Rädker 2022.

<sup>150</sup> vg. Mayring 2015.

<sup>151</sup> vg. Mayring 2015, S. 97–101; vg. Kuckartz and Rädker 2022, S. 130.

<sup>152</sup> vg. Heffner 2011, S. 21.

<sup>153</sup> vg. Gätzer 2013.

<sup>154</sup> vg. Mayring 2015, S. 19.

<sup>155</sup> vg. Mayring 2015, S. 22 ff

- In diesem Zusammenhang ist nicht ausgeschlossen, dass in diesen Aufgabefeldern ergänzend quantitative Methoden zum Einsatz kommen können, um bessere Ergebnisse zu erzielen.

Mayring legt folgende Grundsätze fest, an denen sich qualitative Forschung orientieren sollte.<sup>156</sup>

1. Die qualitative Inhaltsanalyse muss die Vorzüge der quantitativen Techniken einbeziehen, da sonst der Vorwurf, beliebig zu sein, zur Geltung kommen kann.
2. Das Material sollte Teil einer Kommunikationskette sein und sich in ein Kommunikationsmodell einordnen lassen.
3. Die qualitative Inhaltsanalyse sollte sich durch bestimmte Gütekriterien kontrollierbar machen lassen.
4. Jede qualitative Inhaltsanalyse beginnt mit einer Quellenkunde.
5. Material ist niemals ohne Vorbehalte und sollte nicht ohne Bewertung übernommen werden. Hierbei sollte das Vorverständnis des Inhaltsanalytikers, wie theoretischer Hintergrund, Fragestellung und Vorannahmen, dargelegt werden.
6. Die qualitative Inhaltsanalyse ist ein Prozess zum Verstehen vielschichtigen Materials und sollte auf hintergründigen Sinn abzielen.
7. Die qualitative Inhaltsanalyse muss direkt an die alltäglichen Prozesse des Verstehens und der Interpretation sprachlichen Materials anschließen.
8. Es muss die Perspektive des Textproduzenten angenommen werden, um das Duplizieren des eigenen Denkens und Vorwissens zu verhindern.
9. Die Interpretation des sprachlichen Materials ist niemals abgeschlossen, es besteht immer die Möglichkeit, dass es zu einer erneuten Interpretation kommt.
10. Die qualitative Inhaltsanalyse muss die bildhaften Grundbegriffe in das zugrundeliegende Kommunikationsmodell aufnehmen, damit die jeweilige Analyserichtung präzisiert werden kann.
11. Sprachwissenschaftliche Bedeutungseinheiten sind für die Definition von inhaltsanalytischen Analyseseinheiten denkbar.
12. Kontexteinheiten können als Möglichkeit der wissenschaftlichen Erklärung verwendet werden.
13. Die qualitative Inhaltsanalyse muss sich an den pragmatischen Bedeutungstheorien orientieren, die aus dem Bereich des zu untersuchenden Materials stammen.
14. Die Interpretationsregeln der strukturalen Textanalyse müssen nutzbar sein für qualitative Inhaltsanalysetechniken, um dadurch die bedeutenden Bereiche des Materials analysieren zu können.
15. Der umfeldbezogene Aspekt des zu untersuchenden Materials sollte durch die Interpretation von Sprechabschnitten erfassbar gemacht werden.
16. In die qualitative Inhaltsanalyse müssen empirische Analysen der Psychologie von Textverarbeitung und Kategorienbildung einbezogen werden, um zu gewährleisten, dass die Prozesse des Verstehens und Interpretierens von sprachlichem Material einbezogen werden.
17. Als qualitative Inhaltsanalyse kann das systematische Zusammenfassen von sprachlichem Material formuliert werden. Dadurch können aus den

<sup>156</sup> vgl. Mayring 2015, S. 29, 32, 38, 43, 44, 49.

Makrooperatoren, wie Auslassen von Propositionen, Generalisation von zusammengehöriger Proposition etc., Regeln entwickelt werden.<sup>157</sup>

18. Für die Inhaltsanalyse kann es sich als hilfreich erweisen, durch die Psychologie der Kategorienbildung klare Anweisungen zur Definition von Kategorien abzuleiten.

Kuckartz definiert die qualitative Inhaltsanalyse in Abgrenzung zu Mayring so, dass der zentrale Bestandteil einer qualitativen Analyse die Kategorien sein sollten. Er verfolgt dabei einen wesentlich praxisorientierteren Ansatz und kritisiert Mayring für seine zu starke Theoretisierung des Themenbereichs. Mit diesem wird das für die Forschungsfrage inhaltlich wichtige Material codiert. Diese Kategorien können deduktiv, induktiv oder deduktiv-induktiv erstellt werden.<sup>158</sup> Deduktive Kategorienbildung bedeutet in diesem Kontext, dass vom Allgemeinen auf das Spezielle geschlossen wird. Im Kontrast dazu werden bei der induktiven Kategorienbildung vom Speziellen zum Allgemeinen Kategorien gebildet.

Basierend auf diesen Definitionen sollte sich die qualitative Inhaltsanalyse im Vergleich mit anderen Analysemethoden, wie der quantitativen Inhaltsanalyse, messen lassen können. Zur Evaluierung des Materials, welches für die qualitative Inhaltsanalyse verwendet wird, gibt es verschiedene Methoden. Diese Methoden können unter anderem eine Beobachtung, Gruppendiskussionen, Dokumentenanalyse oder Interviews sein.<sup>159</sup> Wie bereits dargestellt, wurde für die Beantwortung der in dieser Masterarbeit gestellten Forschungsfragen für die Datenerhebung das teil-standardisierte Interview gewählt, im Speziellen das problemzentrierte Interview.<sup>160</sup> Diese Erhebungsmethode ist eine mündliche Befragungsmethode und enthält einen Fragenkatalog, den der Interviewer möglichst genau zu beachten hat. Das bedeutet, dass die Fragen des Fragenkatalogs im Verlauf des Interviews gestellt werden sollten, es sei denn, sie wurden bereits durch eine vorhergehende Antwort des Befragten beantwortet. Die Fragen können, weil es sich um ein teil-strukturiertes Interview handelt, in der Reihenfolge variiert werden. Der Fragenkatalog dient als Leitfaden, weshalb diese Interviewform ebenfalls als Leitfadeninterview bezeichnet wird. Leitfadeninterviews sind u. a. für die Befragung kleiner Gruppen oder bei Einzelinterviews geeignet. Ziel dieser Methode ist es, Wissen, Erfahrung oder Sichtweise des Befragten zu erfahren. Dabei sollte der Interviewer keinen Einfluss auf den Interviewten ausüben, was jedoch nicht immer realisierbar ist.<sup>161</sup> Das bedeutet, dass die Reaktionen des Interviewten nicht zwingend auf den Frageinhalt zurückzuführen sind, sondern ebenfalls auf die Beziehung und Interaktion zwischen Interviewer und Interviewten. Aus dieser Tatsache können sich Problemstellungen ergeben, weshalb eine Einflussnahme des Interviewers nötig werden kann.

### 5.1.2 Vorgehen nach Kuckartz

Ein einfaches Modell der qualitativen Inhaltsanalyse nach Kuckartz wird in Abbildung 10 dargestellt. Im Verlauf der qualitativen Inhaltsanalyse wird in einem ersten Schritt ermittelt, welches Material analysiert werden soll.<sup>162</sup> In diesem Schritt werden Fragen über die Entstehungssituation geklärt sowie darüber, welche Informationen die Interviewten im Vorfeld hatten. In diesem Schritt soll ein möglichst umfassendes Bild geschaffen werden über das vorliegende Datenmaterial und dessen Entstehung. Bei der Methodik nach Kuckartz ist bei

---

<sup>157</sup> vg. Mayring 2015, S. 45–47

<sup>158</sup> vg. Kuckartz and Rädker 2022, S. 40.

<sup>159</sup> vg. Hochschule Luzern 2020.

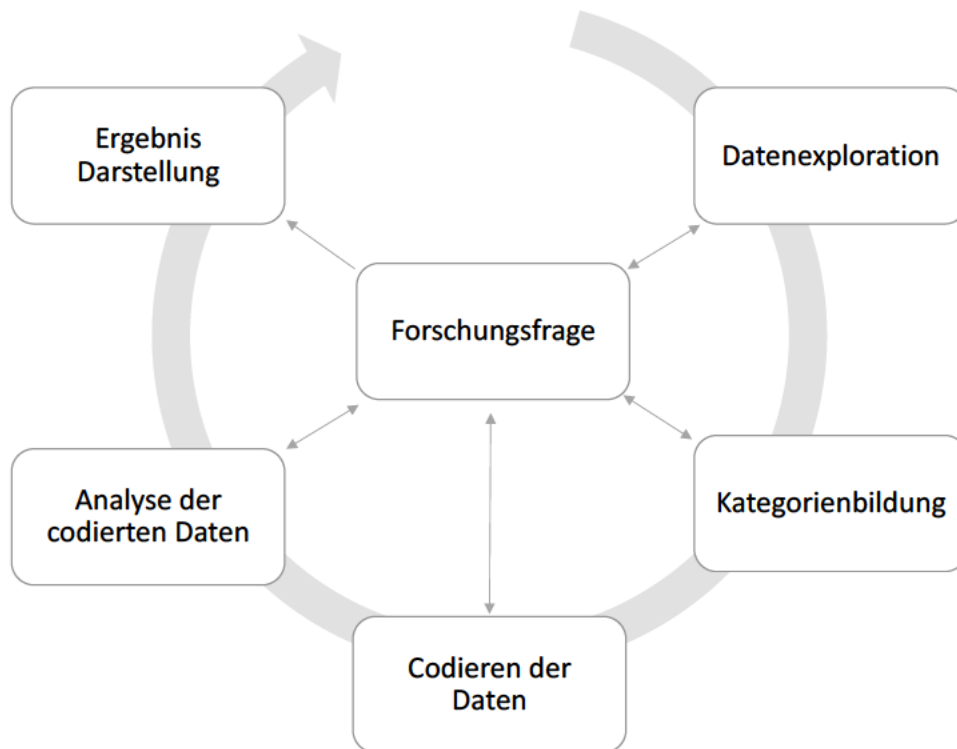
<sup>160</sup> vg. Stier 1999, S. 186; vg. Mayring 2016, S. 67.

<sup>161</sup> vg. Stier 1999, S. 187.

<sup>162</sup> vg. Kuckartz and Rädker 2022, S. 122.

allen Schritten wichtig, dass die Forschungsfrage, Interviewleitfaden sowie die Zielstellung der Arbeit im Vordergrund der Betrachtung stehen. An dieser Stelle kann bei größeren Mengen an Datenmaterial eine erste Auswahl getroffen werden, welches Material verwendet werden soll. Im Fall der vorliegenden Arbeit wurden acht Interviews geführt. Es hat sich im Verlauf der Interviews gezeigt, dass durch eine Erhöhung der Anzahl der Interviews kein weiterer Wissensgewinn zu erwarten ist. Diese Annahme beruht auf den Antworten der Interviewteilnehmer, die sich im Verlauf an vielen Stellen ähnelten. Durch diesen Schritt wird untersucht, unter welchen Bedingungen das Interview erstellt wurde und welche Größen den Ausgang des Interviews beeinflusst haben könnten. Dabei sind die beteiligten Personen wichtig, der Handlungshintergrund des Verfassers, die Zielgruppe des Materials, die konkrete Entstehungssituation und der soziokulturelle Hintergrund der Personen. In Kapitel 4.2 wurden die Hintergründe und Rahmenbedingungen des Interviews dargelegt. Die persönlichen Hintergründe der Interviewpartner werden am Anfang eines jeden Interviews durch die Befragten selbst dargelegt.

Ein weiterer Aspekt in diesem Schritt ist eine intensive Auseinandersetzung mit dem transkribierten Text selbst. Es werden wichtige Abschnitte gekennzeichnet, die Struktur des Textes betrachtet und die Aufmerksamkeit wird auf den Ablauf des Interviews gerichtet.



**Abbildung 10 - Ablaufmodell nach Kuckartz<sup>163</sup>**

### 1. Phase

Die formalen Charakteristika des Materials werden ebenso beschrieben, das heißt, wie das Material vorhanden ist. In der Regel liegt es in Textform vor, wie in dieser Arbeit als Transkript. Durch die theoretische Differenzierung wird der Hintergrund beleuchtet, vor dem die

<sup>163</sup> n An ehnung an Kuckartz and Räd ker 2022, S. 106.

Analyse stattfindet. Dies bezieht sich beispielsweise darauf, welche Forschung der Analyse vorausgeht und welche Erkenntnisse gewonnen werden können. Am Ende dieser Phase soll eine intensive Vertrautheit mit dem Datenmaterial hergestellt werden. Darüber hinaus können die verschiedenen Markierungen als Vorarbeit für den späteren Schritt der Kategorienbildung betrachtet werden.

## 2. Phase

Der zweite Schritt nach der Datenexploration ist die Kategorienbildung. Dabei können zwei Typen von Kategorien unterschieden werden, Haupt- und Subkategorien. Kategorien klassifizieren verschiedene Bestandteile des Textes nach verschiedenen Kriterien.<sup>164</sup> Die Hauptkategorien ermitteln sich meist direkt aus den Forschungsfragen und den damit verbundenen Themenfeldern.<sup>165</sup> Dies trifft in gleichem Maße auf den Interviewleitfaden zu, da dieser die zentralen Themenbereiche behandelt, die innerhalb der Arbeit beantwortet werden sollen. Diese Hauptkategorien können ergänzt werden durch Themenfelder, die sich aus der Literaturrecherche ergeben. Je nachdem, wie komplex das Material ist, kann es sich an dieser Stelle anbieten, Subkategorien zu bilden für eine bessere Übersichtlichkeit des Materials. Diese Kategorien können, wie bereits beschrieben, entweder deduktiv oder induktiv erstellt werden. Bei der induktiven Vorgehensweise sollte jede Auffälligkeit und alles Relevante festgehalten werden. Die induktive Kategorienbildung zeichnet sich dadurch aus, dass sie direkt am zu analysierenden Material erfolgt.<sup>166</sup> Kuckartz schlägt in diesem Zusammenhang ein Rahmenwerk vor, wie vorzugehen ist bei der induktiven Kategorienbildung.

1. Die Kategorien sollten auf Grundlage der vorliegenden Forschungsfrage bestimmt werden. Die Fragestellung, was will man mit der Kategorienbildung erreichen und für welche Fragen sind diese hilfreich.
2. Kategorien und Abstraktionsniveau sollten sich an der Fragestellung und dem zu erreichenden Ziel orientieren. Welches Abstraktionsniveau ist hier am sinnvollsten? Je höher das Abstraktionsniveau, desto allgemeiner ist die Aussage.<sup>167</sup>
3. Material vor der Kategorienbildung durcharbeiten und festlegen, wie der Umfang der zu codierenden Textstellen sein soll.
4. Die Kategorienbildung erfolgt direkt am Text und das, während dieser bearbeitet wird. Dabei wird schrittweise vorgegangen. Die Kategorien werden in diesem Prozess entweder neu erstellt oder die entsprechenden Textstellen werden vorhandenen Kategorien zugeordnet.
5. Wenn im Prozess der Kategorienbildung der Zustand entsteht, dass sehr viele Kategorien gebildet wurden und die Übersichtlichkeit verloren geht oder kaum noch neue Kategorien gebildet werden, sollten die bestehenden Kategorien überprüft werden. Ähnliche Kategorien sollten zusammengefasst oder zu einer allgemeinen Kategorie gebündelt werden.
6. Im letzten Schritt wird das Kategoriensystem finalisiert und es werden Kategoriendefinitionen erstellt. Das passiert, wenn nur sehr wenige oder keine neuen Kategorien mehr erstellt werden. Dies bedeutet nicht, dass keine

---

<sup>164</sup> vg. Kuckartz and Rädker 2022, S. 53.

<sup>165</sup> vg. Kuckartz and Rädker 2022, S. 134.

<sup>166</sup> vg. Kuckartz and Rädker 2022, S. 82.

<sup>167</sup> vg. Strang 2022.



neuen Kategorien mehr hinzukommen können, aber es sollte begonnen werden, die vorhandenen Kategorien zu überprüfen und den Codierungsleitfaden zu erstellen. Zu diesem Zwecke werden die Kategorien definiert und mit Beispielen aus dem Text versehen. Mayring nennt dies Ankerbeispiele.<sup>168</sup>

Bei der deduktiven Vorgehensweise werden die Kategorien, wie bereits erwähnt, unabhängig vom Material erstellt. Dies wird unter anderem als A-Prior-Kategorienbildung bezeichnet.<sup>169</sup> Die Quellen für die deduktive Kategorienbildung können sehr unterschiedlich sein.

- Als ein Beispiel wäre der Leitfaden des Interviews zu nennen. Werden Interviews mit einem Leitfaden durchgeführt, wurde für dessen Entwicklung für gewöhnlich eine theoretische Vorarbeit geleistet, die in den Aufbau des Leitfadens eingeflossen ist. Ein daraus resultierendes Kategoriensystem folgt oftmals den Fragen des Leitfadens, da diese sich am Forschungsinteresse orientieren.
- Eine weitere Quelle für deduktive Kategorienbildung ist eine zugrundeliegende Theorie. Die der Theorie zugrundeliegenden Konzepte und Konstrukte können hierbei in Kategorien übersetzt werden.
- Der Forschungsgegenstand der wissenschaftlichen Arbeit kann ebenfalls zum Zwecke der Kategorienbildung herangezogen werden. Sollten bereits erste Zusammenhänge in der theoretischen Arbeit herausgestellt worden sein, kann dies mit in die Bildung von Kategorien einfließen.
- Verschiedene Prozessmodelle können ebenso eine Grundlage für Kategorien bilden.
- Ist die Zielstellung der Analyse, eine Hypothese zu überprüfen, müssen die erstellten Kategorien diese widerspiegeln. Der Ursprung der Hypothese spielt dabei keine Rolle.

Diese beiden Formen der Kategorienbildung kommen in der Forschung häufig in einer gemischten Form vor. Die Hauptkategorien leiten sich oftmals deduktiv aus dem Forschungsgegenstand oder den Forschungsfragen ab und werden nach diesem Muster gebildet. Weitere Haupt- und Subkategorien können gebildet werden durch die Textrecherche.

Die gebildeten Kategorien werden an etwa 10 – 25 % des Materials getestet und überprüft, je nach Komplexität und Umfang des Kategoriensystems.

### 3. Phase

In dieser Phase wird das gesamte Material nun zum ersten Mal codiert, wenn der Probelauf der 2. Phase erfolgreich war. In einem sequenziellen Vorgang wird das Material durchgearbeitet. Jeder Code, der neu gebildet wird, erhält dabei eine Definition, um diesen von den anderen Codes abzugrenzen. Es werden die Textstellen unterschieden, die inhaltlich als sinnvoll erachtet werden, sowie jene, die als nicht sinnvoll angesehen und deshalb nicht codiert werden. Wichtig ist, dass ein Textabschnitt nicht in einer 1:1-Beziehung zu einer Kategorie steht. Solange das Kategoriensystem nicht bewusst so konstruiert ist, kann ein Textabschnitt ebenfalls zu weiteren Kategorien, deren Definition passt, hinzugefügt werden.

<sup>168</sup> vgl. Mayring 2015, S. 49.

<sup>169</sup> vgl. Kuckartz and Radiker 2022, S. 71.

#### 4. Phase

Ist der 1. Codierungsvorgang abgeschlossen, werden die Hauptkategorien weiter ausdifferenziert und verfeinert. Das Vorgehen lässt sich in diesem Zusammenhang verallgemeinern zu den folgenden Schritten:

5. Auswahl der Hauptkategorie,
6. Selektion der zugehörigen codierten Textstellen,
7. Bilden von Subkategorien nach dem Verfahren der induktiven Kategorienbildung,
8. Organisation der neuen Kategorien und ggf. Zusammenfassung von Subkategorien zu Subkategorien eines höheren Abstraktionsniveaus,
9. Formulieren von Definitionen für die neuen Kategorien.

#### 5. Phase

In dieser Phase findet der 2. Codierungsprozess statt. Die ausdifferenzierten Kategorien werden nun den Textstellen zugeordnet, welche bisher mit der Hauptkategorie verknüpft waren. Dabei wird das gesamte, bereits codierte Material erneut durchlaufen und erneut codiert.

#### 6. Phase

Die 6. Phase nutzt die durch die Codierung gewonnenen Ergebnisse und es findet die Analyse der Daten statt. Dies sind im Fall der inhaltlich strukturierenden Analyse die Forschungsthemen, welche in Abbildung 11 dargestellt sind.

In dieser Arbeit wird die kategorienbasierte Analyse entlang der Hauptkategorie angewandt und es werden verschiedene Visualisierungen verwendet, um besondere Merkmale hervorzuheben. Dabei werden für jede Hauptkategorie die Ergebnisse aufgelistet. Das bestimmende Kriterium ist hier, was thematisch zugehörig und was nicht Teil der Aussage ist. Diese Analyse ist eine beschreibende Technik, weshalb die Kategorien in eine sinnvolle Reihenfolge gebracht werden sollten, um eine bessere Nachvollziehbarkeit zu ermöglichen.

#### 7. Phase

Im letzten Schritt werden die gefundenen Ergebnisse verschriftlicht und in einen Kontext zu den Forschungsfragen gebracht, um diese zu beantworten. Die in dieser Arbeit gewählte Analyseform ist die kategorienbasierte Analyse entlang der Hauptkategorien.<sup>170</sup>

---

<sup>170</sup> vg. Kuckartz and Rädker 2022, S. 148.

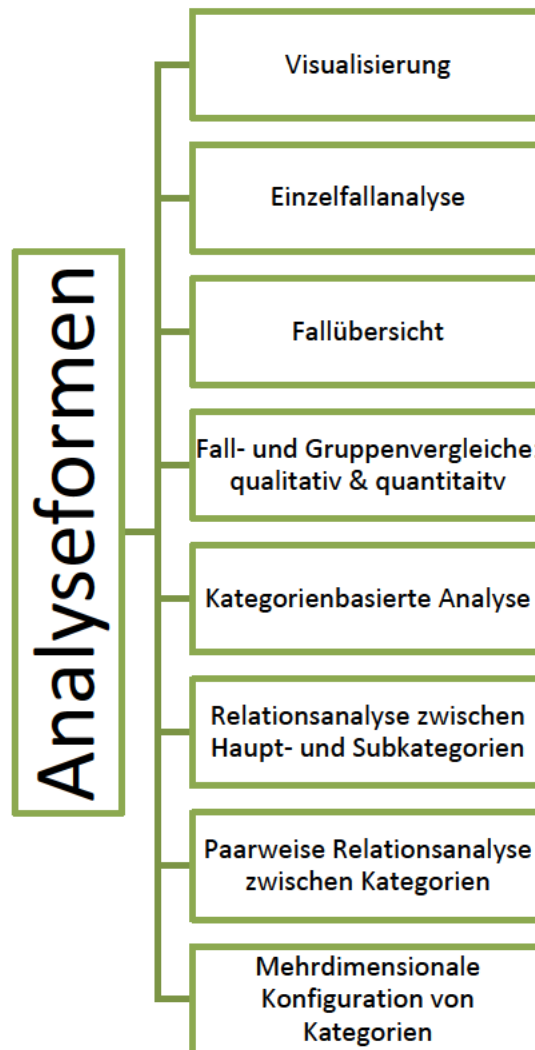


Abbildung 11 - Formen einfacher & komplexer Analyse<sup>171</sup>

<sup>171</sup> in Anlehnung an Kuckartz and Rädiker 2022, S. 147.

## 6 Durchführung der Analyse

### 6.1 Ablaufmodell der Analyse

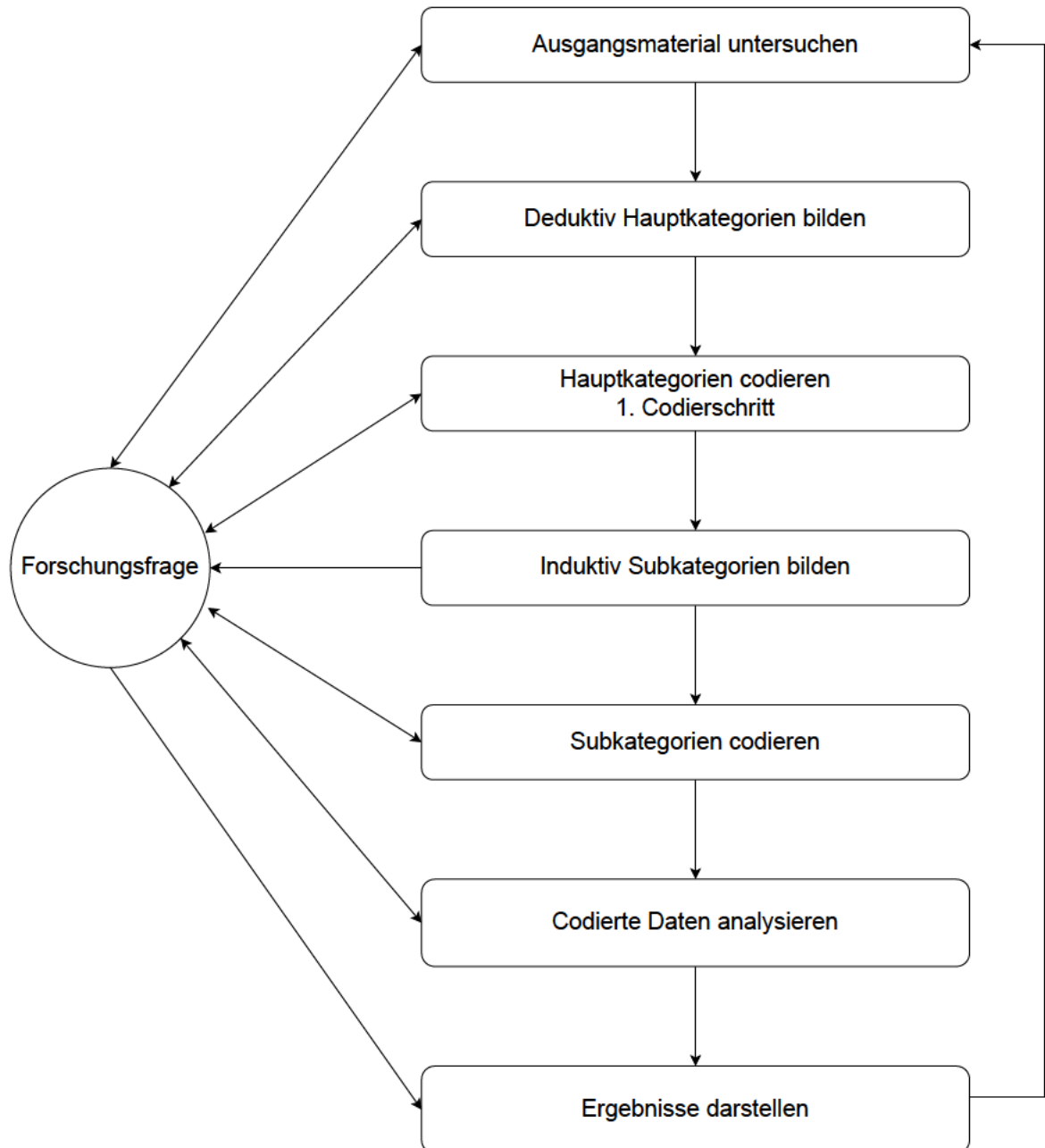


Abbildung 12 - Ablaufmodell der Analyse<sup>172</sup>

<sup>172</sup> in Anlehnung an Kuckartz and Radiker 2022, S. 106.

## 6.2 Charakterisierung Ausgangsmaterial

Das zugrundeliegende Ausgangsmaterial dieser Masterarbeit sind acht Experteninterviews. Die Struktur der Interviews wurde im Kapitel 4 dargelegt. Der Ort des Interviews war bei jedem Interview die Online-Meeting-Plattform Microsoft Teams. Dadurch konnten die Interviewpartner den Ort selbst wählen, von dem aus sie das Interview führen möchten, wodurch ein angenehmes Interviewumfeld angestrebt wurde. Die Aufzeichnung erfolgte dabei durch die integrierte Aufnahmefunktion dieser Plattform. Für die Terminwahl wurde den Interviewpartnern ermöglicht, sich Termine auszusuchen. Der Zeitraum für die Terminauswahl umfasste drei Monate.

Am Anfang eines jeden Interviews wurden die Interviewpartner auf diese Aufnahme hingewiesen und gesondert haben alle Interviewpartner ihr Einverständnis zur Aufnahme des Interviews in Schriftformat erklärt.<sup>173</sup> Der Zeitrahmen der Interviews betrug 40 bis 80 Minuten, je nach Art und Ausführlichkeit der Antworten sowie abhängig von der Sprechgeschwindigkeit der interviewten Person. Der Interviewleitfaden, in Form eines Baumdiagramms, wurde den Befragten im Vorhinein zur Verfügung gestellt, damit diese sich inhaltlich auf das Interview vorbereiten konnten.<sup>174</sup> Durch die Bereitstellung des Interviewleitfadens im Voraus wurde eine höhere Qualität der Antworten angestrebt, da die Interviewpartner sich auf die inhaltlichen Schwerpunkte des Interviews vorbereiten konnten. Es wurde in diesem Zusammenhang offen kommuniziert, dass der Zweck der Interviews eine weitere Verarbeitung in der Masterarbeit ist.

Die Akquirierung der Interviewpartner erfolgte durch direkte Kontaktaufnahme via Priorität, über die Social-Plattform LinkedIn und basierend auf Empfehlungen von anderen Interviewpartnern. Die Kriterien für die Eignung als Experte im Sinne dieser Arbeit waren, dass die Partner eine langfristige Erfahrung im Bereich des Projektmanagements vorweisen konnten sowie Erfahrung in Implementierungsprojekten besitzen. Als weiteres Filterkriterium war es notwendig, dass die Interviewpartner Erfahrungen bei der Implementierung von NCLC-Plattformen haben und dadurch qualitative Aussagen zur Implementierung machen konnten. Diese Experten haben sich ausführlich zu Beginn ihrer Interviews vorgestellt, weshalb in Tabelle 2 eine verkürzte Vorstellung vorliegt.

Das Material wurde, wie bereits erwähnt, über die integrierte Aufnahmefunktion von Microsoft Teams aufgenommen und später mit Hilfe des Programms MAXQDA 2022 transkribiert.<sup>175</sup> Die Transkription erfolgte nach den Regeln der vereinfachten Transkription:

- Füllwörter wie „Ähm“ oder „Öhm“ werden aus dem Transkript weggelassen, weil es hier nicht auf die Semantik der Sprache zur Erreichung des Forschungsziels ankommt.
- Stottern und Wortwiederholungen, die direkt aufeinanderfolgen, werden weggelassen, da diese, genauso wie die vorher genannten Füllwörter, nicht zweckdienlich sind zur Erreichung des Forschungsziels.
- Vorhandene Dialekte werden in hochdeutsche Sprache übersetzt, da die Beibehaltung des Dialektes zu keinem Wissensgewinn im Sinne des Forschungsziels führt.
- Satzabbrüche werden zugunsten einer besseren Lesbarkeit gelöscht.
- Bei unverständlichen Anteilen in der Aufzeichnung wird **(unverständlich)** im Transkript eingefügt.
- Nonverbale Merkmale werden nicht in das Transkript aufgenommen

<sup>173</sup> An age 2 – 9; A-VI

<sup>174</sup> An age 1, A-I

<sup>175</sup> MAXQDA ist eine Software zur computergestützten qualitativen Text- und Datenanalyse

Die Interviewpartner werden nicht anonymisiert innerhalb der Masterarbeit, damit diese als Experten erkennbar sind. Auf diese Tatsache wurden die Interviewpartner in mündlicher und schriftlicher Form hingewiesen.

Der Fokus der Analyse liegt ausschließlich auf der inhaltlichen Ebene, emotionale Faktoren werden nicht berücksichtigt.

Eine weitere intensive Auseinandersetzung mit dem Text, wie Kuckartz sie fordert, erfolgt durch den Prozess des Transkribierens.

Ein wichtiger Schritt ist ein erster Durchlauf der Bearbeitung des Materials. Ergebnis dieses Schrittes sind das Hervorheben von wichtigen inhaltstragenden Stellen und kurze Fallzusammenfassungen. Eine Fallzusammenfassung ist eine stichpunktartige, tatsachen- und textorientierte Zusammenfassung des jeweiligen Interviews.<sup>176</sup> Diese Fallzusammenfassung orientiert sich an den Forschungsfragen. In Anlage 10 sind diese Fallzusammenfassungen zu sehen. Hierbei sollten die Charakteristika der interviewten Person und deren Bezug zur Forschungsfrage herausgestellt werden.

**Tabelle 2 - Kurzsteckbrief Interviewpartner**

Name	Interviewnummer	Hintergrunddaten
Dr. Eckhard Hauenherm	EB01	<ul style="list-style-type: none"> <li>- Promovierter Germanist</li> <li>- Tätigkeit als Projektleiter nach der Promotion in einer IT-Beratung</li> <li>- Selbstständiger Berater seit 7 Jahren</li> <li>- Erfahrung mit No Code/Low Code seit 2003</li> <li>- Seit 2009 beim Project Management Institute aktiv; leitet seit 2 Jahren dort die Community of Practice Citizen Development</li> </ul>
Robin Rosengrün	EB02	<ul style="list-style-type: none"> <li>- Studium Maschinenbau</li> <li>- Tätig als Low Code Enabler bei der EnBW AG</li> <li>- Aufgabenfeld ist Befähigung von MA auf der Microsoft Power Platform</li> <li>- In vorherigen Positionen Tätigkeit als Projektmanager und Projektleitung</li> </ul>
Fabian Gackstätter	EB03	<ul style="list-style-type: none"> <li>- Studium Internationale Wirtschaftsinformatik und Master in Business Administration</li> <li>- Tätig als Solutions Architect für Dynamics 365 und Microsoft Power Platform</li> <li>- Arbeitet seit 20 Jahren mit Dynamics und CRM-Lösungen</li> <li>- Leitet NCLC-Projekte bei Microsoft</li> </ul>
Simone Göttl	EB04	<ul style="list-style-type: none"> <li>- Tätig als Unternehmensberaterin seit 13 Jahren</li> <li>- Aktueller Fokus auf New-Work-Themen, wie Einführung und Rollout von NCLC-Plattformen</li> <li>- Fokusfeld ist die Finanzindustrie</li> </ul>
Christian Rohrbeck	EB05	<ul style="list-style-type: none"> <li>- Studium Wirtschaftsinformatik</li> <li>- Aktuelle Tätigkeit ist IT-Projektmanagement</li> <li>- Ist seit 20 Jahren in dieser Tätigkeit</li> </ul>

<sup>176</sup> vgl. Kuckartz and Rädker 2022, S. 124.

		<ul style="list-style-type: none"> <li>- Kontakt zu Low-Code-Plattformen über eigene Tätigkeit in einem Einführungsprojekt als Entwickler</li> </ul>
Moritz Hesse	EB06	<ul style="list-style-type: none"> <li>- Tätigkeit als Technical Consultant bei der Firma smapOne, einer No-Code-/Low-Code-Plattform</li> <li>- Betreut dort Kunden im Rahmen von Integration und Integrationsvorhaben</li> <li>- Besitzt Projektmanagement-Erfahrung sowohl als Mitglied als auch in leitender Funktion</li> </ul>
Tino Janke	EB07	<ul style="list-style-type: none"> <li>- Studium in Jura</li> <li>- Tätig als IT-Berater in einer Telekom-Tochter, speziell zuständig für Microsoft Power Platform</li> <li>- Weitere Aufgabenfelder sind Projektsteuerung</li> <li>- Ehrenamtlich tätig für das Project Management Institute im Bereich Citizen Development</li> </ul>
Dr. Peter Fey	EB08	<ul style="list-style-type: none"> <li>- Tätigkeit als Projektmanager bei der T-Systems Multi Media Solutions</li> <li>- Ehrenamtliche Tätigkeit für das Project Management Institute, dort zuständig für die Community of Practice Citizen Development</li> </ul>

### 6.3 Deduktive Kategoriendefinition

In diesem Schritt werden die ersten Haupt- und Subkategorien deduktiv gebildet. Diese werden anhand des finalen Interviewleitfadens erstellt. Zielstellung der Interviews war es, die Interviewpartner anzuregen, über ihre Erfahrungen in den Bereiche NCLC und Projektmanagement, mit Fokus auf Implementierungsprojekten, zu berichten. Dabei sollte herausgestellt werden, welche Faktoren sie bei einer Implementierung von solchen Plattformen als wichtig empfinden und welche wichtigen Vorbereitungs- und Nachbereitungsmaßnahmen in einem Unternehmen nötig sind. Ein weiteres Ziel war es, dass die Interviewpartner über Erfahrungen und Problemstellungen sprechen, die ihnen in der Vergangenheit bei solchen Projekten in Verbindung mit dem Einführen einer solchen Plattform und mit den Plattformen selbst begegnet sind.

Für das eigentliche Einführungsprojekt stand im Mittelpunkt der Fragestellung die Klärung, welche Eigenschaften ein Unternehmen besitzen muss und welches Vorgehensmodell sich unter welchen Eigenschaften für die Einführung von NCLC eignet. Hierbei sind die Erfahrungswerte der Interviewpartner von Bedeutung und welches Vorgehen sie als positiv bewerten und welches eher auf Ablehnung stößt. Wichtige Faktoren sind dabei die Gründe für die Entscheidung für ein Vorgehensmodell und welche Tools aus den Vorgehensmodellen dabei verwendet werden. Es ist in diesem Zusammenhang ebenfalls von Interesse, wie die bevorzugte Variante der Interviewpartner mit den Punkten Projektrollen, wichtige Stakeholder, Änderungsanfragen an das Projekt und Befähigung von Mitarbeitern umgeht.

Abschließend steht die Fragestellung der Qualitätssicherung und Erfolgsmessung des Projektes. Hierbei soll herausgefunden werden, ob es spezielle Mechanismen und Handlungsempfehlungen gibt, die zur Qualitätssicherung und Erfolgsmessung angewendet werden. Die letzte Frage des Leitfadens richtet sich darauf, den Interviewpartnern Raum zu geben, wichtige Aspekte zu betonen oder weitere Einflüsse zu nennen, welche eventuell nicht Teil

der Fragen waren. Diese Frage bildet keine eigene Hauptkategorie, da hier nach Inhalten gesucht wurde, die eventuell zu einer der anderen Kategorien gehörten.

Aus dieser theoretischen Differenzierung des Leitfadens ergeben sich die folgenden Kategorien.

**Tabelle 3 - Liste deduktive Hauptkategorien**

Kategoriennummer	Thematische Beschreibung Hauptkategorie
1	Vorbereitung im Vorfeld durch Unternehmen
1.1	Vorteile NCLC-Plattformen
1.2	Nachteile NCLC-Plattformen
2	Eigenschaften Unternehmen
2.1	Unternehmensgröße
2.2	Kapital
2.3	Mindset
3	Risiken und Hürden bei der Einführung NCLC-Plattformen
4	Projektorganisationsform
5	Wahl Vorgehensmodell
6	Vorteile gewähltes Vorgehensmodell
7	Nachteile gewähltes Vorgehensmodell
8	Verwendete Tools Vorgehensmodell
9	Umgang mit Änderungswünschen
10	Rollen im Projekt
11	Wichtige Stakeholder außerhalb des Projekts
12	Skalierung des Pilotprojekts
13	Befähigung Mitarbeiter
14	Qualitätssicherung im Projekt
15	Erfolgsmessung des Projekts

In einem ersten Probedurchlauf werden diese Kategorien am Interview EB01 Eckhard Hauenerm getestet.

## 6.4 1. Codierung des Materials

In dieser Arbeit wird nach dem System der deduktiven-induktiven Kategorienbildung vorgegangen.<sup>177</sup> Bei dem induktiven System entstehen die Kategorien im Verlauf der Bearbeitung des Textes und sind nicht vorgegeben, was bei einer deduktiven Kategorienbildung der Fall wäre. Die deduktiven Kategorien orientieren sich inhaltlich am Interviewleitfaden. Im Verlauf der Analyse wurden die deduktiv gebildeten Kategorien um weitere induktiv gebildete Kategorien ergänzt. Das Kategoriensystem ist als integraler Bestandteil der Analyse zu

<sup>177</sup> vg. Kuckartz and Rädker 2022, S. 134.



verstehen. Dieses trägt zur Nachvollziehbarkeit der Analyse bei und ermöglicht unter anderem eine Reliabilitätsprüfung durch eine weitere forschende Person. Diese Art der Kategorienbildung bietet sich bei der inhaltlich strukturierenden qualitativen Inhaltsanalyse an, weil das Ziel dieser Arbeit ist, die Texte inhaltlich nach spezifischen inhaltlichen Aussagen der Experteninterviews zu strukturieren, um die vorgestellte Fragestellung zu beantworten. Wie in Abbildung 12 gezeigt, ist der nächste Schritt nach der Bildung der deduktiven Hauptkategorien die 1. Codierungsphase. Es werden keine Textstellen codiert, die nicht für die Beantwortung der Forschungsfrage von Nutzen sind. Die Länge der Textstellen, welche codiert werden, ist nicht festgelegt und orientiert sich an dem Inhalt und der Übereinstimmung mit der Kategoriendefinition. Es werden also Sinneinheiten codiert, welche in der Regel mindestens ein Satz sind. Dies schließt jedoch nicht aus, dass auch einzelne Wörter Teil der Kategorie sein können. Gibt es Verständnis- oder Zwischenfragen, werden diese in diesem Fall mitcodiert. Im Verlauf des Codierungsprozesses kann es dazu kommen, dass Textstellen die Definitionsanforderungen mehrerer Kategorien erfüllen. Nachdem die ersten Codes gebildet wurden, werden diese konkretisiert, indem sie definiert werden. Durch diesen Schritt soll die Zuordnung einzelner Textstellen zu einem Code oder mehreren Codes aufgezeigt werden.

## 6.5 Erstellen von Subkategorien und 2. Codierungsvorgang

Dieser Schritt erfolgt, wenn der erste Kategorisierungsvorgang vorgenommen worden ist.<sup>178</sup> Es werden nun induktive Subkategorien gebildet. Diese Subkategorien sind als weitere Differenzierung der deduktiv gebildeten allgemeinen Hauptkategorien zu betrachten. Diese Subkategorien werden direkt am Material gebildet, an den Textstellen, die der Hauptkategorie im 1. Codierungsvorgang zugeordnet wurden. Ist dies geschehen, werden inhaltsgleiche oder ähnliche Subkategorien zusammengefasst. Wie die Hauptkategorien erhalten die Subkategorien Definitionen, unter welchen Bedingungen diese verwendet werden. Aufgrund der praktischen Nähe und der Anforderung, eine entsprechend große Datenmenge zum Bilden der Subkategorien zu verwenden, kann in diesem Schritt gleich die nächste Phase des 2. Codierungsprozesses mitbearbeitet werden.<sup>179</sup> Zum Bilden der Subkategorien werden die Interviews EB01 und EB02 verwendet und daran anschließend findet die Überprüfung der induktiv gebildeten Codes statt. Direkt danach werden die restlichen Interviews anhand dieser Codes bearbeitet. Sollte im Verlauf des 2. Codierungsvorgangs eine erhebliche Anzahl neuer Subcodes entstehen, ist das Codierungssystem erneut zu überprüfen und das codierte Material abermals zu bearbeiten. Die Subkategorien werden so definiert, dass nicht jeder neue Einzelfall eine eigene Subkategorie bildet, sondern dass die Definitionen so gestaltet sind, dass viele Einzelfälle unter eine Subkategorie fallen können. Dies ist auf den geringen Umfang des zugrundeliegenden Materials zurückzuführen, das aus acht Interviews besteht. Die komplette Herleitung des Kategoriensystems, mit den entsprechenden Codedefinitionen, kann im Codebuch in Anlage 11 eingesehen werden.

---

<sup>178</sup> vgl. Kuckartz and Rädker 2022, S. 138.

<sup>179</sup> vgl. Kuckartz and Rädker 2022, S. 142.

## 6.6 Kategorisierung und Analyse

In der 6. Phase, nachdem das finale Codesystem gebildet und den entwickelten Subkategorien das Datenmaterial zugeordnet worden ist, werden die Kategorien analysiert.<sup>180</sup> Das Codesystem ist in Tabelle 4 dargestellt. Die gewählte Analyseform ist die kategorienbasierte Analyse entlang der Hauptkategorien. Das Ziel der Arbeit ist, die inhaltlichen Aussagen auszuwerten und weiterzuverarbeiten. Dafür wurde ein Summary Grid erstellt, in dem die Aussagen der Experten zusammengefasst wurden.<sup>181</sup> Ein solches Summary Grid ist hilfreich, wenn Aussagen von Experten über das Interview verteilt sind, was im vorliegenden Fall so ist. Inhalt des Summary Grids ist dabei eine thematische Zuordnung von zusammengefassten Aussagen der Experten zu einem Themenkomplex. Dies ermöglicht eine hohe Transparenz und ist ein Vorschrift für die letzte Phase.

Die letzte Phase passiert überschneidend mit der 6. Phase. Inhalt dieser Phase sind das Verschriftlichen der Ergebnisse und die Dokumentation des Vorgehens. Dies wird im Ergebnisbericht festgehalten. Dabei werden die inhaltlich für die Forschungsfragen relevanten Aspekte der codierten Textstellen mit eigenen Worten festgehalten und mit Zitaten ergänzt. Das Ergebnis der letzten beiden Schritte wird ausführlich in Kapitel 7 aufgezeigt und dargestellt.

---

<sup>180</sup> An age 11, A-CXX X

<sup>181</sup> vg . Kuckartz and Räd ker 2022, S. 143.; An age 12, A-CLCI

## 7 Ergebnispräsentation

Im Folgenden werden die Ergebnisse der durch die Analyse gewonnenen Daten deskriptiv wiedergegeben. Auf Grundlage der in diesem Kapitel vorgestellten Ergebnisse und der Ergebnisse der Literaturrecherche wird in Kapitel 8 das Vorgehensmodell aufgebaut. Zu Beginn dieses Kapitels kommt es zu einer Vorstellung des vollständigen Kategoriensystems, inklusive aller Subcodes. Die Vorstellung der Ergebnisse erfolgt entlang der Hauptkategorien und den damit verbundenen Subkategorien in Kapitel 0.

### 7.1 Kategoriensystem mit Subcodes

**Tabelle 4 - Codesystem inklusive Subcodes**

Hauptcodes	Subcodes	Sub-Subcodes	
1 - Vorbereitung Unternehmen	1.1 - Auswahl Citizen Developer		
	1.2 - Motivation Stakeholder		
	1.3 - Projektbewusstsein		
	1.4 - Organisation Unternehmen Vorbereitung		
	1.5 - Umgang mit MA, Unternehmenskultur		
	1.6 - Vorteile NCLC-Plattformen	1.6.1 - Lösung Kommunikationsproblem	
		1.6.2 - Prozessverbesserung	
		1.6.3 - Verbesserung der Arbeitsabläufe MA	
		1.6.4 - Steigerung der Qualität von Lösungen	
		1.6.5 - Höhere Entwicklungsgeschwindigkeit	
	1.7 - Nachteile NCLC-Plattformen	1.7.1 - schlechte Qualität der Programme	
		1.7.2 - Falsche Erwartungen	
	1.8 - Governance Einfluss	1.8.1- Fehlende Governance	
		1.8.2 - Governancestrategie	
1.9 – IT-Abteilung und NCLC	1.9.1 - Negative Auswirkungen für IT		
	1.9.2 - Entlastung IT-Situation		
	1.9.3 - Fachkräftemangel IT		
2 - EP: Eigenschaften Unternehmen	2.1 - Mindset	2.1.1 - Fehlerkultur	
	2.2- Kapital		
	2.3 - Unternehmensgröße		
	2.4 - Technische Planung		
	2.5 - Erfahrenes Projektmitglied		
	3.1 – Schatten-IT		

3 - EP: Risiken und Hürden vor der Einführung	3.2 - Cybersecurity und Datenschutz	
	3.3 - Thema unterschätzt	
	3.4 - Weitere Risiken	
4 - EP: Projektorganisationsform		
5 - Wahl Vorgehensmodell	5.1 - Agiles Vorgehensmodell	5.1.1 - Option auf Wasserfallelemente
		5.1.2 - Begründung agiles Vorgehen
	5.2 - Hybrides Vorgehensmodell	5.2.1 - Ausprägung hybrides Modell
		5.2.2 - Begründung hybrides Vorgehen
6 - Vorteile gewähltes Vorgehensmodell	6.1 - Vorteile agiles Vorgehen	6.1.1 - Vorteil agil: schnelles Feedback
		6.1.2 - Vorteil agil: Reaktionsgeschwindigkeit
		6.1.3 - Vorteil agil: Motivation
	6.2 - Vorteile traditionelles Vorgehensmodell	6.2.1 - Vorteil trad.: Planungssicherheit
	6.3 - Vorteile hybrides Vorgehensmodell	
7 - Nachteile gewähltes Vorgehensmodell	7.1 - Nachteile agiles Vorgehensmodell	7.1.1 - Nachteil agil: fehlende Planungssicherheit
		7.1.2 – Nachteil agil: Projektcommitment
		7.1.3 - Nachteil agil: fehlende Erfahrung
		7.1.4 - Nachteil agil: schlechte Dokumentation
	7.2 - Nachteile traditionelles Vorgehensmodell	7.2.1 - Nachteil trad.: zu starr
		7.2.2 - Nachteil trad.: Dauer
	7.3 - Nachteile hybrides Vorgehensmodell	7.3.1 - Nachteil hybrid: zu starr
		7.3.2 - Nachteil hybrid: Kosten
		7.3.3 - Nachteil hyb.: fehlende Erfahrung mit einem der Modelle
		7.3.4 - Nachteil hybrid: kombi.-Probleme reine Vorgehensweisen
8 - Tools Vorgehensmodell	8.1 - Tools agiles Vorgehen	
	8.2 - Tools trad. Vorgehen	
9 - Änderungswünsche	9.1 - Changemanagement agiles Vorgehen	
	9.2 - Changemanagement hybrides Vorgehen	
10 - Rollen und Stakeholder im Projekt	10.1 - Projektverantwortlicher	
	10.2 - technische Rollen	
	10.3 - andere Rollen	
	10.4 - Enabler	
	10.5 – Governance-Rolle	
	10.6 - Projektorganisationsrollen	
	10.7 – Changemanagement-Rolle	

	10.8 - Scrum Master	10.8.1 - Scrum Master sollte im Projekt sein
		10.8.2 - Vorbehalte reine Scrum Master-Rolle
	10.9 - Citizen Developer/ Fachbereichsmitarbeiter	10.8.1 - Vorbehalte Citizen Developer
11 - Wichtige Stakeholder außerhalb des Projekts	11.1 - Organisatorische Vorgesetzte	
	11.2 - Sonstige Beteiligte	
	11.3 - Geschäftsleitung	
	11.4 - Abteilungen	
	11.5 - Mitarbeitervertretung	
	11.6 - Alle Mitarbeiter im Unternehmen	
	11.7 - Support Abteilungen	
12 - Skalierung Pilotprojekt	12.1 - Sequenzielles Vorgehen	
	12.2 - Paralleles Vorgehen	
	12.3 - Sequenziell-Paralleles Modell	
	12.4 - Vorbereitende Maßnahmen	
13 - Befähigung Mitarbeiter	13.1 - Befähigung durch Schulung/ Workshop	
	13.2 - Befähigung durch Community	
	13.3 - Befähigung durch Multiplikatoren	
	13.4 - Befähigung zum Selbstlernen	
14 - Qualitätssicherung Projekt	14.1 - Qualitätssicherung durch emotionale Aspekte	
	14.2 - Qualitätssicherung durch Monitoring	
	14.3 - Qualitätssicherung durch Feedback	
15 - Erfolgsmessung Projekt	15.1 - Quantitative Größen	
	15.2 - Qualitative Größen	

## 7.2 Kategorienzusammenfassung

In den folgenden Unterabschnitten werden die Ergebnisse der Analyse deskriptiv wiedergegeben. Ziel dieses Kapitels ist es, einen Gesamtüberblick über die Ergebnisse der Analyse zu erhalten und diese im Zusammenhang mit den deduktiv gebildeten Hauptkategorien sowie Subkategorien darzustellen. Dabei werden die im Summary Grid erstellten Zusammenfassungen der codierten Textstellen pro Interview verwendet, um ein Gesamtbild dahingehend aufzuzeigen, wie die forschungsrelevanten Meinungen der Experten zu den einzelnen Themen sind. Die für die Zusammenfassung herangezogenen einzelnen Experten-Zusammenfassungen können im Summary Grid nachgelesen werden.<sup>182</sup> Es werden teilweise Zitate der Experten verwendet, um Ergebnisse in der Kategorienpräsentation hervorzuheben. Einige Kategorien überschneiden sich inhaltlich mit anderen. Da das Thema in viele Bereiche eines Unternehmens eingreift, konnte dies nicht vermieden werden. Es dient

<sup>182</sup> An age 12, A-CLVI

dem Hervorheben von besonders wichtigen Aspekten, die durch die Experten angesprochen wurden.

### **Auswahl der Citizen Developer**

Die Auswahl der Citizen Developer ist ein wichtiger Faktor, der bereits im Vorfeld des Einführungsprojektes geplant werden sollte. Dabei sehen die Experten verschiedene Faktoren, die zu beachten sind. Einer dieser Faktoren ist, dass nicht jeder Mitarbeiter in einer Fachabteilung ein Citizen Developer ist, sondern ein oder wenige ausgewählte Mitarbeiter, die verschiedene Kriterien erfüllen.

*„[E]s werden nicht alle Mitarbeiter auf die No-Code-/Low-Code-Plattform befähigt. Also das hieße, man würde einer Fachabteilung noch mal eine zusätzliche Aufgabe aufoktroieren. Das ist zu viel. Es muss ja auch nicht jeder in der Fachabteilung ein Excelspezialist sein.“*

EB08, Pos. 51

Dies begründet sich auf der einen Seite darin, dass die Befähigung von allen Mitarbeitern einer Fachabteilung einen sehr großen Aufwand bedeuten würde und die Gefahr besteht, dass die originären Fachbereichsaufgaben vernachlässigt werden. Ein weiterer Aspekt, weshalb nicht jeder Mitarbeiter ein Citizen Developer sein sollte, ist, dass bestimmte Voraussetzungen an den einzelnen Citizen Developer gestellt werden sollten. Diese Voraussetzungen oder Kriterien sind nicht zwingend alle zu erfüllen, aber es wirkt begünstigend für die Tätigkeit. Citizen Developer sind Mitarbeiter in den Fachabteilungen, die ein starkes technisches Interesse aufweisen. Es ist wichtig, dass der Mitarbeiter ein prozesslogisches Denken besitzt und Zusammenhänge gut nachvollziehen kann. Explizit ist dabei zu sagen, dass der Mitarbeiter ein gewisses Verständnis für IT-Wissen haben oder sich dieses aneignen sollte. IT-Kenntnisse, die dabei erforderlich sind, bewegen sich im Bereich der Datenstrukturen und dahingehend, wie Daten organisiert sind. Ein weiterer Aspekt, der durch die Experten hervorgehoben wurde, ist, dass Citizen Developer meist Mitarbeiter sind, die sich bereits mit anderen Anwendungsprogrammen tiefgreifend beschäftigen und dort die jeweiligen Funktionalitäten ausreizen, um ein bestmögliches Ergebnis zu erzielen.

*„Leute, die sich mit PowerPoint auskennen, Leute, die sich mit Excel auskennen, die dann tatsächlich mit den entsprechenden Editoren oder Wizards sich einfache Applikationen, einfache Automationen zusammenklicken können, ohne eine Zeile Code schreiben zu müssen.“*

EB03, Pos. 7

### **Motivation der Stakeholder**

Eine Herausforderung, der sich Unternehmen im Vorfeld stellen müssen, ist es, die relevanten Stakeholder für das initiale Einführungsprojekt zu motivieren. In Bezug auf diesen Themenkomplex sehen die Experten verschiedene wichtige Aspekte, die berücksichtigt werden sollten. Es wird durch den Experten EB07, Tino Janke, darauf hingewiesen, dass alle Projektmitglieder eine intrinsische Motivation für das Projekt haben sollten. Diese intrinsische Motivation entsteht, wenn die Projektmitglieder im Vorfeld bereits negative Erfahrungen gesammelt haben, indem sie auf Lösungen durch die IT-Abteilungen warten mussten und wissen, was sich verbessern sollte. Dabei wird hervorgehoben, dass das Einführungsprojekt nicht aufgezwungen wirkt und für die Stakeholder des Projektes ein Mehrwert erkennbar ist. Eine Hürde kann es sein, die Mitarbeiter dafür zu begeistern, sich an dem

Projekt zu beteiligen. Eine fehlende oder schlechte Motivation kann sich negativ auf den Projektoutput auswirken sowie auf die nachgelagerten Folgeprozesse während des aktiven Betriebs der Plattform. Ein weiterer Grund für fehlende Motivation bestimmter Stakeholder kann sein, dass diese sich nicht zuständig fühlen und das Thema eher als zur IT-Abteilung zugehörig gesehen wird.

*„Das hat wahrscheinlich auch was mit einem Rollenverständnis zu tun, denn das klassische Rollenverständnis ist so, für Applikationen ist die IT zuständig und deswegen machen die das auch.“*

EB01, Pos. 17

Falsche Erwartungen werden von einem weiteren Experten angeführt als Problemstellungen für die Motivation im Zusammenhang mit dem Einführungsprojekt. Die Kriterien dieses Faktors werden erfüllt, wenn das Unternehmen eine unterschiedliche Erwartungshaltung hat zwischen dem, was NCLC können sollte, und dem, was die Plattform tatsächlich kann. Daraus kann für Mitarbeiter, die sich mit diesem Thema initial beschäftigen, eine Hürde entstehen, welche sich negativ auf ihre Motivation auswirkt.

Es wird die Motivation der Citizen Developer als Stakeholder hervorgehoben. Dabei ist es wichtig, den Citizen Developern ein gewisses Maß an Sicherheit zu bieten, da diese sich ohne die benötigte Bildung in ein neues Tätigkeitsfeld vorwagen. Das bedeutet nicht, dass es keine Kritik geben darf, aber eine respektvolle Unternehmenskultur wird als wichtiger positiver Faktor für die Motivation der Citizen Developer angesehen.

### **Projektbewusstsein**

*„Also, wirklich kann ich nur raten, daraus wirklich ein Einführungsprojekt zu machen.“*

EB01, Pos. 79

Mehrere der Experten weisen auf diesen Aspekt explizit hin, dass für das Projekt ein Bewusstsein existieren sollte. Eine Plattformsentscheidung ist eine weitreichende Entscheidung, die bei falscher Umsetzung zu weitreichenden negativen Konsequenzen führen kann, für das ganze Unternehmen. Es wird explizit darauf hingewiesen, dass Praxiserfahrungen zeigen, dass viele Unternehmen eine NCLC-Einführung oftmals nur als Softwareprojekt sehen, mit einem überschaubaren Personenkreis, der involviert werden muss. Diese Betrachtungsweise kann schon zu Beginn des Projektes größere Probleme aufwerfen, insbesondere dahingehend, dass wichtige personelle und finanzielle Ressourcen nicht zur Verfügung stehen. Zusätzlich müssen gewisse organisatorische Maßnahmen vorher erfüllt sein, die oftmals noch nicht umgesetzt sind. Zu diesen organisatorischen Maßnahmen gehören unter anderem wichtige personelle Ressourcen, die im Vorfeld für die Administration der Plattform geklärt werden sollten. Weitere Ressourcen, die Klärungsbedarf haben, sind diejenigen für die IT-Infrastruktur, die für den Support der Plattform nötig ist. Als wichtig wird ebenfalls angesehen, dass eine Changemanagement-Organisationseinheit existiert, welche mit dem Thema NCLC umgehen kann.

*„Wo die meisten Kunden sagen: ‚Ich dachte, das wäre ein IT-Thema.‘ Und das ist so eine Plattformsentscheidung halt nicht. Die betrifft im Zweifelsfall das ganze Unternehmen und genau so muss ich die auch aufhängen. Da es eben nicht mit den drei Admins getan ist, die das jetzt noch nebenhermachen, sondern es ist ja ein organisationsweiter Change, den ich da einführe und den muss ich auch mit der entsprechenden Manpower unterfüttern.“*

## EB03, Pos. 23

Diese Changemanagement-Organisationseinheit wird wichtig aufgrund der Tatsache, dass die Einführung einer Plattform eine grundlegende Entscheidung für ein Unternehmen ist, es sollten daher weitere wichtige Fragen geklärt werden. Diese Fragen sind für einen optimalen Verlauf wichtig und beziehen sich darauf, wie die Befähigung der Mitarbeiter funktionieren sollte. Wie wird die Motivation zur Entwicklung auf der Plattform kreiert und erhalten? Welche Maßnahmen müssen dafür neben der Plattform implementiert werden?

**Vorbereitung der Organisation im Unternehmen**

Mit der Einführung einer Plattform entstehen viele Risiken. Vorbereitende Maßnahmen, welche die Organisationen in diesem Zusammenhang vor der Einführung klären sollten, wurden von nahezu allen Experten genannt. Eine solche Plattformscheidung ist keine Entscheidung nur über neue Technik, sondern hat weitreichende Konsequenzen für die ganze Organisation. Aus diesem Grund sollte eine weitreichende strategische Planung durchgeführt werden. Zu diesem Zweck ist eine Empfehlung durch den Experten EB02, Robin Rosengrün, das Thema NCLC in die Digitalisierungsstrategie des Unternehmens zu übernehmen und dort zu verankern. Diese klare Art des Commitments hinsichtlich der Thematik wird als positiv für das Projekt gesehen. Weiterführend muss in der strategischen Planung berücksichtigt werden, dass neue Strukturen innerhalb der Organisation entstehen werden, als Beispiel wird eine Supportstruktur genannt. Die Einführung von NCLC führt ebenfalls dazu, dass innerhalb des Unternehmens neue Regeln für Prozesse entstehen werden. Als vorbeugende Maßnahme sollte sich im Vorfeld des Projektes überlegt werden, wie diese neuen Regeln und Prozesse allen relevanten Stakeholdern der Plattform zugänglich gemacht werden. Es wurde im Experteninterview EB05 hervorgehoben, dass viele Unternehmen teilweise Probleme haben, ihre eigenen Prozesse zu verstehen und aus diesem Grund sollten letztere im Vorfeld überprüft werden, damit während des eigentlichen Einführungsprojektes keine unerwarteten Verzögerungen aufgrund des mangelhaften Prozesswissens entstehen. Dies kann sonst zu Verzögerungen im Projekt führen. Als Teil der strategischen Planung sollten Unternehmen sich Gedanken machen, wie weit die Plattform in das Unternehmen skaliert werden soll und bereits die entsprechenden Abteilungen frühzeitig in die Planung einbinden. Das zeigt den Bedarf nach einem umfassenden Informationsmanagement auf.

Neben den Prozessen und strukturellen Vorbereitungen muss den Unternehmen bewusst sein, dass die Einführung oftmals auch einen Wandel in der Unternehmenskultur voraussetzt. Es muss innerhalb der Organisation im Vorfeld begonnen werden, sich für neue Vorgehensweisen und Prozesse zu öffnen. Dadurch soll die Problemstellung eines Innovationsstaus vermieden werden. Nur weil ein Prozess vorher bereits funktioniert hat, heißt das nicht, dass dieser nicht auch verändert und verbessert werden kann. Mit dieser kulturellen Veränderung sollte in der Organisation, wenn nicht bereits vollzogen, ebenfalls ein Verständnis für Daten und Datenstrukturen erzeugt werden. Dabei geht es unter anderem um Anwendungsfälle, bei denen Informationen oftmals weitergeschickt werden, aber es wesentlich sinnvoller wäre, diese in einer Datenbank zentral zugänglich für alle abzulegen. Aufgrund dieses fehlenden Verständnisses ist oftmals für viele MA nicht ersichtlich, wie NCLC positiv in die Arbeitsabläufe integriert werden kann und welche Anwendungsfelder im Unternehmen verbessert werden können. Dieser Aspekt überschneidet sich mit dem Thema, dass ein Mehrwert in der Plattform gesehen werden sollte, um die MA und das Projektteam für das Thema zu motivieren. Ein weiterer wichtiger Aspekt ist die eigentliche



Wahl der Plattform. Aufgrund der Vielzahl der unterschiedlichen NCLC-Plattformanbieter sollten sich Unternehmen bei der Auswahl der Plattform Zeit lassen und die für ihre Prozesse passende auswählen. Während der Auswahl sollten Fragen geklärt werden wie:

- Welcher Anbieter passt für das eigene Unternehmen?
- Ist es sinnvoll, mehrere Plattformen zu verwenden?
- Wie funktioniert die Integration und Anbindung zu bereits vorhandenen Systemen?
- Wie könnte eine Exitstrategie aussehen, wenn eine Plattform einmal nicht mehr für das eigene Unternehmen passt?

Weiterführende organisatorische Fragestellungen sind die Themen Governance und Cybersecurity. Diese werden ausführlich in der Ergebnispräsentation zur Kategorie Governance-Einfluss besprochen.

### **Umgang mit MA als Teil der Unternehmenskultur**

Ein Aspekt, der von einigen Experten explizit hervorgehoben wird, ist die Veränderung der Unternehmenskultur. Wie bereits erwähnt, wird keine neue Office-Anwendung eingeführt, sondern eine Plattform, die bei voller Nutzung das Potenzial hat, das gesamte Unternehmen zu verändern und dadurch die Mitarbeiter stark beeinflusst. Im Falle, dass die Mitarbeiter nicht ausreichend eingebunden werden und die Befähigung unzureichend erfolgt, kann das viele negative Auswirkungen auf das Unternehmen haben, wie ablehnendes Verhalten der Mitarbeiter bezüglich der Nutzung der Plattform oder die Nichtbeachtung von Richtlinien. Als Teil dieses Problems wird gesehen, dass der traditionelle Blick auf Mitarbeiter sich verändern muss. Im Experteninterview EB08 wird in diesem Zusammenhang Steve Jobs zitiert, der Unternehmer und ehemalige CEO von Apple.

*„Es macht keinen Sinn, kluge Leute einzustellen und ihnen zu sagen, was zu tun ist. Wir stellen kluge Leute ein, damit sie uns sagen können, was zu tun ist.“*

Steve Jobs

Die Schaffung eines angenehmen Arbeitsklimas wird als wichtig angesehen, um Innovationen zu fördern sowie ein qualitativ gutes Feedback zu ermöglichen. Sollten Mitarbeiter sich bei verschiedenen NCLC-Fragestellungen einbringen, sollten diese keine Angst haben müssen, direkt mit negativen Konsequenzen und Äußerungen herabgewürdigt zu werden, wenn sie sich äußern und dies nicht dem regulären Vorgehen in IT-Strukturen entspricht. Dieser Umgang mit Mitarbeitern und Citizen Developern kann es ermöglichen, dass völlig neue innovative Ideen in den Problemlösungsprozess hineingebracht werden, welche vom normalen IT-Lösungsprozess abweichen. Mit diesem Aspekt einhergehend müssen Unternehmen ihren Mitarbeitern mehr Vertrauen entgegenbringen und die Freiräume lassen, dort eigene Ideen zu verwirklichen. Dies bedeutet nicht, dass kein Monitoring erfolgen kann oder eine Endabnahme durch die IT-Abteilung nicht sinnvoll ist, aber es sollten im Vorfeld nicht zu viele Grenzen gesetzt werden, um das Potenzial von NCLC voll ausnutzen zu können.

*„[U]nd auch das Vertrauen seinen Mitarbeitern gegenüber, weil, ansonsten lasse ich halt viel Potenzial von so einer No-Code-/Low-Code-Plattform liegen. Wir haben Kunden, die es tatsächlich einführen, aber dann nur mit tatsächlichen Entwicklern bestücken. Das heißt, die ganzen Citizen Developer bleiben nach wie vor außen vor. Was natürlich dann viel Potenzial von so einer Plattform wieder verschenkt. Zum einen dem Misstrauen geschuldet, dass man sagt: ‚Ich traue das meinen Mitarbeitern einfach nicht zu.‘ Oder: ‚Wir können die*

*nicht in den entsprechenden Kontrollmechanismus hineinbringen, wie ich es mit meiner zentralen IT machen kann.“*

EB03, Pos. 23

### **Vorteile NCLC-Plattformen**

Im Vorfeld der Einführung sollten sich Unternehmen darüber bewusst sein, welche Vorteile sie aus so einer Plattform generieren können, um der Problematik von falschen Erwartungen entgegenzuwirken. Nahezu alle Experten haben hervorgehoben, dass einer der großen Vorteile einer solchen Plattform die Lösung des Kommunikationsproblems ist. Das Kommunikationsproblem entsteht beim Transfer von Anforderungen der Fachabteilung an die IT-Anforderungen. Dies ist unter anderem der Fall, wenn die speziellen Anforderungen der Fachabteilungen durch die IT-Abteilungen in IT-Anforderungen übertragen werden. Es kommt oft zu Problemen in der Kommunikation zwischen den Anforderern und den IT-Entwicklern aufgrund des unterschiedlichen fachlichen Backgrounds. Oftmals können IT-Mitarbeiter nicht die spezifischen Anforderungen, die ein Fachabteilungsmitarbeiter hat, voll erfassen oder es kommt zu Missverständnissen. Dieses Problem herrscht jedoch auch in der Beziehung von IT-Mitarbeiter und Fachmitarbeiter. Fachmitarbeiter können oftmals spezifische umsetzungsrelevante Inhalte nicht nachvollziehen und sind dann unzufrieden mit der Lösung. Diese Problemstellung entfällt, wenn der Entwickler gleichzeitig der Fachmitarbeiter ist, genauer gesagt eine Person aus der Fachabteilung und damit der Anforderer. Durch die Vereinigung beider Rollen in einer Person wird ein Verständnis für beide Seiten erzeugt. Lösungen, die auf dieser Grundlage entwickelt werden, sind bestmöglich auf die Bedarfe der Anforderer angepasst und können bei komplizierteren Anforderungen auch in Zusammenarbeit mit der IT-Abteilung erstellt werden, um ein optimales Ergebnis zu erzielen.

*„Dass der Fachbereich eine Vorstellung davon hat, was er will. Aber eine Sprachblockade quasi zum IT-Bereich besteht und dann in den Köpfen verschiedene Vorstellungen davon entstehen, was man eigentlich braucht und das fliegt dann erst auf, oder merkt man dann erst, wenn es dann den ersten Prototyp gibt.“*

EB02, Pos 13

Das führt zu einem weiteren Vorteil, der durch die Experten angesprochen wurde. Die Geschwindigkeit von Entwicklungen kann signifikant erhöht werden. Wie bereits erwähnt, entfallen die Zeitverluste, die entstehen zwischen dem Missverstehen von Anforderungen und dem Entdecken des Missverständnisses im Prototypen-Review oder im Review der fertigen Anwendung. In diesem Falle müsste die Anwendung erneut geändert und danach erneut durch die Fachabteilung neu bewertet werden. Dieser Prozess würde sich so oft wiederholen, bis eine akzeptable Applikation entsteht, bei der entweder alle Anforderungen vollumfänglich umgesetzt wurden oder bei der ein erneutes Durchlaufen des Prozesses aus Aufwandssicht nicht mehr gerechtfertigt ist. Eine Erhöhung der Entwicklungsgeschwindigkeit wird gleichfalls dadurch erreicht, dass die Hürde zur Entwicklung einer Applikation sinkt. Ein langwieriger Prozess von der Anforderung einer neuen Applikation und dem Zeitunterschied, bis die Applikation durch die IT-Abteilung realisiert wird, entfällt durch das Zusammenlegen der Rollen Entwickler/ Anwender. Im Falle, dass in einer Fachabteilung der Bedarf für eine solche Applikation auftaucht, kann diese durch den Citizen Developer schnell realisiert werden. Dies wird dadurch ermöglicht, dass die Plattformen einfach gestaltet sind und eine große Anzahl von Anwendungsfällen, ohne größeren Aufwand, abgedeckt werden

kann. Das Bottleneck IT-Abteilung wird also umgangen und eine größere Anzahl von Applikationen wird entwickelt, was zu einem Innovationssprung führen kann. Applikationen, die aufgrund einer niedrigen Priorität für die IT-Abteilung oder wegen des zu großen Aufgaben-Backlogs der IT-Abteilung nicht entstanden wären, können durch NCLC umgesetzt werden und erhöhen dadurch die Produktivität im Unternehmen.

Die Möglichkeit, schnell und mit präzisen Anforderungen eigene Lösungen umsetzen zu können, sorgt dafür, dass die Qualität von Lösungen steigt. Dies zeigt sich in Szenarien, wo früher eine Excel-Tabelle durch den Fachbereichsmitarbeiter erstellt worden wäre. In dieser Excel-Tabelle wären in irgendeiner Form Daten gespeichert oder ausgewertet und dann weitergeschickt oder in andere Unternehmenssysteme eingegeben worden.

*„da Sachen, die jetzt bei uns nicht im ERP-System abgebildet werden, wäre ich immer sofort dabei, macht das bitte, bitte, bevor ihr dort anfangt, es über Excel-Dateien etc. zu machen, nutzt da ein Low-Code-Tool, eine Low-Code-Plattform dafür. Man hat immer ein besseres Ergebnis, als wenn man das über Kalkulationstabellen macht oder so was.“*

EB02, Pos. 17

Ein weiterer Qualitätssprung bei Lösungen kann dadurch erreicht werden, dass eine Kollaboration zwischen IT-Abteilung und Citizen Developern erfolgt. Dies kann sich in verschiedenen Aspekten begründen. Einerseits, dass eine gewisse Datenstruktur eingehalten werden soll, verschiedene andere unternehmensinterne Systeme angebunden werden müssen oder die Anwendung einen so hohen Grad an Komplexität besitzt, dass diese durch den Citizen Developer aufgrund des fehlenden IT-Wissens nicht mehr alleine realisiert werden kann. Für diese Szenarien kann eine IT-Abteilung unterstützend wirken und dem Citizen Developer bei komplexen Lösungen helfen. Andererseits könnte die IT-Abteilung die Basis der Applikation selbst erstellen, wohingegen das spezifische User Interface, kurz UI, durch den Citizen Developer auf die Bedürfnisse der Abteilungen realisiert und angepasst werden könnte.

Durch den hohen Grad an Individualisierbarkeit entsteht auch eine Steigerung der persönlichen Produktivität und der Arbeitsabläufe. Das können kleine Prozesse sein, wie der Eingang einer Rechnung und das automatische Ablegen in einem Ordner bis dahin, dass Daten aus einem System heruntergeladen und dann durch die App weiterverarbeitet werden können. Dies kann Arbeitsabläufe beschleunigen und Prozesse verbessern.

### **Nachteile NCLC-Plattformen**

Wie bereits ausgeführt, besitzen NCLC verschiedene Vorteile. Alle Experten weisen jedoch ebenfalls auf verschiedene Nachteile der Plattformen hin, auf die Unternehmen und Anwender achten sollten. Dabei wird durch zwei Experten herausgestellt, dass die Qualität der Anwendungen von den Fähigkeiten der entwickelnden Personen abhängt. Hierdurch werden Schnittmengen aufgezeigt mit der Kategorie „Auswahl Citizen Developer“. Die Qualität ist unter anderem davon abhängig, wie gut das technische und logische Verständnis des Entwicklers der Anwendung ist. Je nach Befähigungsstand des Entwicklers und dessen IT-Wissen kann es zu schlecht erstellten Anwendungen kommen, die für den Beginn funktional sind, aber in der weiterführenden Nutzung und Entwicklung verschiedene Problemstellungen aufwerfen können. Dieses Qualitätsproblem überschneidet sich mit dem Bereich der falschen Erwartungen, die durch Unternehmen an eine NCLC-Plattform gestellt werden. Aufgrund dessen, dass Citizen Developer wesentlich weniger IT-Wissen besitzen, ist die Erwartungshaltung von Unternehmen falsch, dass die entwickelten Anwendungen

vollumfänglich für alle Bereiche des Unternehmens genutzt werden. Komplexe Unternehmensanwendungen mit einem hohen Datenvolumen, welche geschäftskritisch sind, können nicht durch eine Low-Code-Plattform ersetzt werden. Darin stimmen die Experten EB01-EB05 anhand ihrer Aussagen überein. Es wird nicht möglich sein, komplette IT-Prozesse oder sogar die gesamte IT-Abteilung damit zu ersetzen. Im Experteninterview EB04 wird auch darauf hingewiesen, dass ein Vorteil der Plattform zwar ist, dass schnell Anwendungen entwickelt werden können, aber dies nicht für jeden Anwendungsfall gilt. Es sollte in diesem Zusammenhang genau analysiert werden, was realistische Entwicklungszeiten für die im Unternehmen vorhandenen Anwendungsfälle sind. Es sollte nicht durch falsche Erwartungen Druck auf die Mitarbeiter ausgeübt und dadurch ein Scheitern der Plattform riskiert werden.

Zu den Nachteilen zählen auch die Themen Entlastung der IT-Abteilung und Governance/Cybersecurity. Da diese Themen ausführlich in eigenen Kategorien dargelegt werden, wird an dieser Stelle darauf verzichtet.

### **Governance-Einfluss**

Governance wird von fast allen Experten als ein wesentlicher Faktor für NCLC-Plattformen gesehen. Um eventuell später auftretenden Problemen vorzubeugen, sollten Unternehmen diesbezüglich bereits im Vorfeld Überlegungen treffen, um während des Einführungsprojektes keine Verzögerungen oder Probleme zu riskieren. Dies ist nötig, weil die Plattformen ein weitreichendes Tool-Set besitzen und dadurch sehr viele Freiheiten ermöglichen, wenn diese nicht durch Rahmenbedingungen eingeschränkt werden.

*„Die dritte Wissensbasis, die ich brauche, ist tatsächlich so ein Gefühl für Risiken und Governance, was den Umgang mit Daten angeht.“*

EB01, Pos.23

Die Governance in Unternehmen sollte im Idealfall bereits im Vorfeld geklärt werden oder ein Unternehmen sollte sich darüber zumindest zuvor erste Gedanken gemacht haben. Governance in Bezug auf NCLC bezieht sich auf den Umgang mit den Daten, den Rahmenbedingungen für Anwendungsentwickler, den Rahmenbedingungen der Anwendungen (Application Life Cycle), den Austausch von Daten zwischen Anwendungen und anderer im Unternehmen vorhandener Software über Konnektoren sowie den Umgang mit Zugriffsberechtigungen. Es müssen ebenso Fragen zur Dokumentation geklärt werden, um kritische Situationen zu vermeiden, wenn aus irgendwelchen Gründen der Anwendungsentwickler dem Unternehmen nicht zur Verfügung steht. Das kann zu der Situation führen, dass niemand die Applikation mehr verändern kann oder sich jemand mit sehr hohem zeitlichen Aufwand hineinarbeiten muss. Eine Klärung sollte auch dahingehend erfolgen, welche Bestandteile vom Tool Set der Plattform benutzt werden dürfen. Werden die Citizen Developer nur auf Canvas Apps beschränkt, also die Anwendungsoberfläche, mit der interagiert wird, oder erhalten diese weiterreichende Rechte, um zum Beispiel Business Intelligence Tools verwenden zu können? Wird ein dynamisches Modell verfolgt, bei dem diese Freigaben an bestimmte Bedingungen geknüpft sind, dann muss geklärt werden, wer unter welchen Bedingungen von wem für weitere Anwendungen freigeschaltet werden kann. Weitergehend sollten auch Überlegungen erfolgen, wie das Monitoring der Plattform vonstatten gehen soll. Es sollte auch überlegt werden, ob eine Anwendung aus einer Fachabteilung für das Unternehmen nützlich ist, wie sie skaliert werden kann und wer solche unternehmensweiten Anwendungen von Seiten der IT-Abteilung betreut.

Diese Fragestellungen sind ebenso wichtige Aspekte bei der Auswahl der Plattform. Hat die Plattform eventuell sogar eigene Tools, die den Prozess der Governance erleichtern, kann dies ein wichtiger Entscheidungspunkt sein. Entscheidet sich ein Unternehmen zur Befähigung der Citizen Developer, eine Community zu gründen, sollten ebenfalls dafür entsprechende Governance-Richtlinien erstellt werden. Als weiterer Teil einer Governance-Strategie sollte auch geklärt sein, welche Anwendungen überhaupt entwickelt werden. Wie im Interview EB07, Pos. 15, durch den Experten herausgestellt wird, müssen die Citizen Developer in diese Governance-Überlegung einbezogen werden, um zu vermeiden, dass Richtlinien erstellt werden, die für diese völlig unpraktikabel sind und in der Folge die Plattform nicht genutzt wird.

*„Also mir bringt die beste Governance nichts, wenn ich die Menschen nicht mit an Bord hole in dem Projekt, wenn es also niemand nutzt. Und wie gesagt, mir bringen die begeisterten Nutzer der Plattform nichts, wenn ich das nicht schaffe, in sinnvolle Leitplanken zu geben, diese Projekte.“*

EB07, Pos. 15

Eine notwendige Fragestellung für Unternehmen ist das Thema Sicherheit. Dabei sollten Aspekte betrachtet werden wie Datensicherheit, Datenkonsistenz und generelle IT-Sicherheit des Unternehmens. IT-Sicherheit und Datensicherheit sind keine originären Teile der Governance, sie werden allerdings von den Experten in diesem Zusammenhang genannt. Eine fehlende Governance kann weitreichende Problemstellungen verursachen, welche die nachgelagerten Prozesse beeinflussen. Ist die Governance nicht definiert und wird diese nicht spätestens während des Einführungsprojektes implementiert, besteht die Gefahr, dass eine Schatten-IT oder ein sogenannter Wildwuchs während der aktiven Nutzung der Plattform entsteht. Dies ist ursächlich darin, dass die Plattformen oftmals sehr viele und sehr weitreichende Tools mitbringen, die ohne Rahmenbedingungen eine Gefahr für die IT-Sicherheit bedeuten und ein unkontrolliertes Entwickeln ermöglichen. Es können verschiedene Datenstände entstehen, welche innerhalb eines Unternehmens weiterverarbeitet werden. Das würde bedeuten, unterschiedliche Datenbankstände würden weiterverarbeitet. Das kann im schlimmsten Fall dazu führen, dass Daten verwendet werden, die veraltet sind oder Daten unkontrolliert durch das Unternehmen geschickt werden oder, was wesentlich schlimmer wäre, an Stellen geschickt werden, wo diese nicht sein sollten. Wie bereits weiter oben erwähnt, ist durch Dokumentation eine gewisse Nachvollziehbarkeit erreichbar, allerdings besteht trotzdem die Gefahr, dass ohne entsprechende Vorgaben Applikationen völlig ohne jede Struktur entwickelt werden. Das kann sich einerseits negativ für die Nutzer auswirken, wenn jede Applikation ein anderes User Interface besitzt. Des Weiteren könnten die Citizen Developer mit der Vielfalt an Tools, die solche Plattformen bereitstellen, überfordert werden. Dies betrifft auch das Themenfeld des Application Life Cycles. Tritt der Fall ein, dass ohne Anwendung einer entsprechenden Governance Anwendungen entwickelt werden, welche nicht überwacht werden, kann es dazu führen, dass Applikationen nicht an neue Standards angepasst werden, solange sie ihren originären Zweck immer noch erfüllen. Ein weiterer Punkt ist, dass unfertige Entwicklungsstände in der IT-Struktur liegen bleiben und dadurch eventuelle Sicherheitsrisiken entstehen aufgrund veränderter Entwicklungen, welche nicht angepasst werden. Eine weitere Gefahr entsteht im Kundenkontakt, sollten die Anwendungen ebenfalls dafür verwendet werden, um auf Kundenseite bestimmte Aufgaben zu erfüllen.

## **IT-Abteilung und NCLC**

Es herrscht die dominante Meinung unter den Experten, dass der Fachkräftemangel in der IT-Abteilung durch NCLC-Plattformen nicht vollständig behoben werden kann. Im Experteninterview EB05, Pos. 23 – 25, wird die weiterführende Meinung vertreten, dass die Einführung nicht zwingend zu einer Entlastung der IT-Abteilung beiträgt, sondern es lediglich zu einer Verschiebung der Aufgaben kommt. Dies begründet sich in der Problemstellung der Supportstruktur für die Citizen Developer sowie im benötigten Monitoring der neuen Applikationen.

Durch andere Experten wird hingegen dargelegt, dass eine Entlastung sehr wohl erreicht werden kann, wenn verschiedene Voraussetzungen erfüllt sind. Zu diesen Voraussetzungen zählt eine ausführliche Befähigung der Citizen Developer im Vorfeld der Nutzung für die Entwicklertools der Plattformen sowie bei entsprechender Vorauswahl der Citizen Developer. Die Entlastung der IT-Abteilung würde dadurch entstehen, dass der Aufgaben-Backlog sich verringert, weil die kleinen Anwendungsanfragen durch die Fachabteilungen entfallen würden. Ein weiterer Punkt, der für eine Entlastung angeführt wird, ist, dass die IT-Abteilung selbst die Plattformen nutzen kann. Bei der Nutzung durch die IT-Abteilung und dem damit verbundenen höheren Grad an IT-Wissen können kleine Applikationen wesentlich schneller entwickelt werden, da ein initialer Programmieraufwand weitestgehend entfällt. Im Weiteren wird darauf hingewiesen, dass für die Anwendungen, die durch den Fachbereich erstellt werden, der Review-Prozess entfällt. Wie bereits oben beschrieben, gehört dazu, dass ein Prototyp gezeigt wird und ein Vergleich des Soll-Zustandes mit dem Ist-Zustand vorgenommen wird und die damit verbundenen Korrekturschleifen durchgeführt werden.

Ein weiterer Grund, weshalb der Fachkräftemangel nicht durch NCLC-Plattformen vollständig behoben werden kann, liegt darin, dass die IT-Abteilung nicht allein Entwickleraufgaben innehat. Es wird darauf verwiesen, dass weitere Aufgabenfelder im Bereich der Administration von Datenbanken und Unternehmenssoftware sowie der Instandhaltung von Bestandssystemen liegen.

## **Eigenschaften von Unternehmen**

Als eine wichtige Bedingung, die Unternehmen erfüllen sollten für die Einführung von NCLC-Plattformen, wird von allen Experten die Auffassung vertreten, dass ein bestimmtes Mindset eine Voraussetzung ist oder dieses als stark begünstigend bewertet wird. Dies wird von einigen ebenfalls als agiles Mindset bezeichnet und wird in Kontrast zu einem starren Denken gesehen, welches in veralteten hierarchisch organisierten Strukturen verankert ist. Dieses starre Denken wird als negativer Einfluss für das Einführungsprojekt bewertet. Dieser Einfluss kann ebenfalls über das Einführungsprojekt weitergehend auf die nachfolgende Umsetzung des Konzepts Citizen Development und Nutzung der NCLC-Plattform einwirken. Angesichts dessen wird durch die Experten das agile Mindset bevorzugt. Es wird in diesem Zusammenhang von einigen Experten in zwei Merkmalsbereiche aufgeteilt. Der eine Merkmalsbereich beschäftigt sich mit den Mitarbeitern, die später entweder als Nutzer oder als Citizen Developer mit der Plattform arbeiten sollen. Es herrscht bei einigen Experten die Erwartungshaltung, dass diese Mitarbeiter die Bereitschaft zeigen, Verantwortung zu übernehmen und sich offen zu zeigen für neue Entwicklungen im Unternehmen. Es darf nicht davor zurückgeschreckt werden, unbekannte Aufgabenfelder zu bearbeiten. Ein Wille zur ständigen Weiterbildung wird dabei für Citizen Developer vorausgesetzt, da die Plattformen einem ständigen Wandel unterliegen. Als übergeordnete Instanz sollten die

Vorgesetzten der im Projekt involvierten Mitarbeiter ein Commitment gegenüber dem Projekt und der Nutzung der Plattform empfinden und diesem unterstützend gegenüberstehen. Dadurch soll vermieden werden, dass Mitarbeiter sich zwischen ihrer originären Aufgabe in der Fachabteilung und ihrer Aufgabe im Projekt oder als Citizen Developer entscheiden müssen. Der zweite Merkmalsbereich beschäftigt sich mit der Unternehmensseite. Wie bereits im Abschnitt Motivation Stakeholder dargelegt wurde, ist ein Commitment durch das Unternehmen zum Projekt eine wichtige Voraussetzung. Dies sollte spätestens zu Beginn des Projekts vorhanden sein. Dieses Umdenken von Unternehmensseite sollte sich auf den Umgang mit den Mitarbeitern auswirken. Zu Beginn des Projekts sollte eine Vertrauensbasis geschaffen sein und Unternehmen sollten bereit sein, eine gewisse Fehlertoleranz zuzulassen, vor allem zu Beginn der Implementierung. Es sollte ebenso eine Veränderung in der Wahrnehmung von IT-Themen stattfinden. Bei neuen Aufgaben, welche im Kontext der IT-Aufgaben anfallen, sollte darüber nachgedacht werden, ob dies durch die IT-Abteilung realisiert werden muss oder ob diese Aufgabe nicht durch die Citizen Developer erfüllt werden kann, bei gleicher oder ähnlicher Qualität. Dieses Mindset darf den Führungskräften und Abteilungen nicht aufgezwungen werden, sondern sie müssen die Motivation für die Plattform selbst mitbringen und das sollte durch das Unternehmen gefördert werden. Dafür sind eine generelle Willkommenskultur und Offenheit durch das Unternehmen vorzuleben. Eine weitere Voraussetzung, die jedoch unter den Experten unterschiedlich bewertet wird, ist das Thema zugrundeliegendes Kapital. Dies wird von Experten als weniger wichtig eingestuft. Dabei hebt der Experte EB01 hervor, dass im Grunde nur die Lizenzen bezahlt werden müssen als Minimumaufwand.

*„Also, ich muss die Lizenzen bezahlen können. Das würde ich sagen, das reicht.“*

EB01, Pos. 33

Zwei andere Experten weisen aber auf die Notwendigkeit von finanziellen Mitteln auch für personelle Ressourcen und Investitionen hin. Der Betrieb einer solchen Plattform erfordert gewisse personelle Aufwände, die durch die IT-Abteilungen oder externe Dienstleister erfüllt werden müssen, sowie eine gewisse technische Infrastruktur. Dies bezieht sich ebenso auf das Einführungsprojekt, welches mit entsprechenden finanziellen und personellen Ressourcen ausgestattet werden sollte. Ebenfalls werden die technischen Voraussetzungen durch die Experten benannt und diese werden vom Experten EB06, Moritz Hesse, als einzige wirklich essenzielle Voraussetzung eingestuft. Dabei wird im Experteninterview EB03 genauer darauf eingegangen, welche technischen Voraussetzungen darunter verstanden werden können. Das betrifft auf der einen Seite personelle IT-Ressourcen zur Umsetzung der Richtlinien und des Monitorings auf technischer Ebene sowie die Erstellung einer Infrastruktur-Grundlage zur Skalierung des Projekts und dessen aktivem Betrieb. Die technischen Fragestellungen, die in diesem Zusammenhang wichtig sind, werden als Beispiele unter anderem wie folgt aufgeführt:

*„Welche Szenarien habe ich denn? Was sind klassische Entwickler-Applikationen? Was sind klassische Citizen Developer? Wie sieht die Zone dazwischen aus? Wie sieht auch eine Progression aus von einer Applikation, die ursprünglich mal ein User für sich gebaut hat? Die dann aber so gut war, dass sie irgendwann zur Enterprise-Applikation geworden ist. Wie sieht dieser Lebenszyklus aus?“*

EB03, Pos. 21

Die Unternehmensgröße wird von keinem der Experten als wichtiger Faktor aufgefasst. Es wird lediglich darauf hingewiesen, dass je nach Größe des Unternehmens ein gewisser erhöhter Aufwand für personelle und finanzielle Ressourcen nötig werden kann.

### **Risiken und Hürden vor der Einführung**

Im Folgenden werden weitere Hürden und Risiken für das Einführungsprojekt beschrieben, die durch die Experten geäußert wurden. Dieser Code ist als Ergänzung zu den vorangegangenen Themen zu sehen, wo bereits Hürden in Form von fehlender Governance geäußert und durch die Plattformen selbst verursacht wurden. Die in diesem Abschnitt genannten Hürden ergeben sich teilweise aus den Nachteilen der NCLC-Plattformen und sollten bei der Risikobewertung berücksichtigt werden.

### **Schatten-IT**

Eine Hürde im Projekt wird durch verschiedene Experten darüber definiert, dass die Entstehung einer Schatten-IT, ebenfalls als Wildwuchs benannt, verhindert wird. Dieses Problem kann entstehen, wenn im Vorfeld keine Governance-Strategie festgelegt wurde. Die fehlenden Rahmenbedingungen müssten dann spätestens zu Beginn des Projekts festgelegt werden, um dieses Risiko zu vermindern. Zu diesem Zwecke sollten verschiedene personelle Ressourcen aus der IT-Abteilung vorhanden sein, welche die gebildeten Maßnahmen sowie Richtlinien überwachen und steuern. Eine klare Dokumentation sollte durchgeführt werden, um zu vermeiden, dass Probleme bei der Nachvollziehbarkeit von Aufgaben entstehen, wenn Änderungen im Projekt durchgeführt werden, vor allem wenn Projektmitglieder wechseln. Es sollte zu einer Klärung der Aufgaben kommen und dahingehend, welche Tätigkeiten durch welche Abteilung durchgeführt werden. Werden diese Punkte nicht beachtet, kann es zu negativen Konsequenzen für den Projektverlauf, wie Terminproblemen, kommen.

### **Cybersecurity und Datenschutz**

Eine weitere Hürde, die die Experten im Vorfeld des Projekts sehen, sind die unternehmensinterne IT-Sicherheit und der Datenschutz. Ähnlich wie bei den Governance-Themen sollten Unternehmen im Vorfeld Datenschutz- und Sicherheitsrichtlinien für den Umgang mit den NCLC-Plattformen erstellen. Diese Richtlinien sollten verschiedene Fragen beantworten: Was passiert mit den Daten? Was darf mit den Daten passieren? Was sind Regularien, nach denen sich das Unternehmen in Bezug auf Datenschutz richten muss? Welche Funktionen können freigeschaltet werden, ohne die unternehmensinterne IT-Sicherheit zu gefährden? Lässt ein Unternehmen zu, dass der Hersteller der Plattform Zugriff auf die eigenen Daten erhält? Welche Folgen hat es, wenn kein Zugriff gewährt wird? Welche Daten dürfen durch die Plattform generell verarbeitet werden? Wie schütze ich meine Daten? Wie sichere ich die Plattform ab in Übereinstimmung mit meinen Sicherheitsrichtlinien? Diese Fragestellungen werden als Hürde betrachtet und sollten bedacht werden, vor oder spätestens während der Einführung.

*„Also häufiges Thema ist natürlich, was darf ich in dieser Plattform? Das heißt eigentlich auch, welche Funktionen schalte ich frei an dem Punkt und wie löse ich diesen Sicherheitsaspekt.“*

EB04, Pos. 27



## Thema unterschätzt

Dieser Punkt überschneidet sich mit dem Thema falsche Erwartungen. Die Unterscheidung ist, dass hier durch Experten darauf hingewiesen wird, dass aufgrund eines Mangels an Informationen der Projektumfang unterschätzt wird. Dieses Unterschätzen des Projektumfangs kann Auswirkungen auf das Einführungsprojekt in Bezug auf inhaltliche, zeitliche und Kostenplanung haben. Vor allem zu Beginn des Projekts wird ein hoher personeller Initialaufwand nötig, um einen ersten Wissenstransfer durchzuführen. Es wird durch den Experten EB03, Fabian Gackstatter, herausgestellt, dass die dafür wichtigen Ressourcen vonseiten des Unternehmens teilweise fehlen oder nicht verfügbar sind. Das führt in der Folge zur Verzögerung des Projekts. Eine weitere Hürde ist, dass Unternehmen ihre eigenen Anforderungen nicht oder schlecht definiert haben. Dies begründet sich dadurch, dass Unternehmen erwarten, dass ihnen die Plattformbetreiber bei der Implementierung vorgeben, wie es zu funktionieren hat. Dies ist jedoch eine unzureichende Erwartungshaltung. Aufgrund der hohen Individualität von Unternehmen kann ein Plattformbetreiber so etwas meist nicht leisten. Die Plattform sollte sich an die Anforderungen, die ein Unternehmen hat, anpassen und aus diesem Grund sollten Unternehmen ihre Anforderungen selbst definieren und diese von Anfang an mit dem Plattformbetreiber kommunizieren.

*„Und es ist eben auch wichtig, dass die Kunden verstehen, dass wir ihnen nicht vorkauen können: ‚So müsst ihr eine No-Code/Low-Code-Plattform aufsetzen.‘ Wir haben zwar entsprechende Best Practices, aber jedes Unternehmen tickt eben anders und unser Job ist es dann, die Plattform so anzupassen und den Kunden durch den richtigen Entscheidungsprozess durchzuführen, dass am Ende die Plattform dabei rauskommt.“*

EB03, Pos.25

Als weiterer Aspekt dieses Themenfeldes ist anzuführen, dass aufgrund des unterschätzten Projektumfangs das Thema in der Unternehmenshierarchie zu niedrig angesiedelt ist. Die Problemstellung entsteht, dass eine Veränderung in der Priorisierung einen Mehraufwand bedeutet, den es zu vermeiden gilt, weil jeder Mehraufwand zur Veränderung von Projektumfang, Projektkosten oder Projektzeitplan führen kann. Als Beispiel zur Vermeidung dieser Hürde wird angeführt, dass die Plattform ursprünglich geplant nur in einer Abteilung eingeführt werden sollte, aber es von der Priorität und dem Umfang eingeplant werden sollte, dass eine Erweiterung auf das ganze Unternehmen möglich ist. Dadurch soll der Fall vermieden werden, dass im Verlauf der Einführung und Skalierung doch mehr Abteilungen hinzukommen und an diesen Stellen wichtige Ressourcen fehlen.

Ein unterschätzter Projektumfang und die Frage, wie weit die NCLC-Plattformen in das Unternehmen hineinreichen, werden auch in Bezug auf die Stakeholder angesprochen. Der Einsatz der Plattformen und die Umsetzung des Citizen-Developer-Konzepts hat weitreichende Auswirkungen auf die Mitarbeiter des Betriebs. Es sollte also berücksichtigt werden, dass Mitarbeitervertretungen von Beginn an in die Planung des Projekts einbezogen werden, um eine Blockade durch diese zu vermeiden.

## Weitere Risiken

Weitere Risiken, die durch Experten vereinzelt erwähnt wurden, aber keiner der anderen Kategorien zugeordnet werden konnten, fallen in diese Kategorie. Dabei wurde im Interview EB03 auf die Problemstellung einer zu starren IT-Bürokratie verwiesen. Aufgrund langer Genehmigungsprozesse kann sich ein zeitliches Risiko für das Projekt ergeben. Die

Einführung einer solchen Plattform bricht starre Strukturen auf und das sollte durch ein Unternehmen angenommen und gefördert werden.

Eine weitere Hürde, die im Experteninterview EB04 angesprochen wurde, ist die Plattformauswahl. Diese sollte sehr gründlich durchgeführt werden und es sollten sich Exitstrategien für den Fall überlegt werden, dass während des Einführungsprojektes oder später im aktiven Betrieb festgestellt wird, dass die gewählte Plattform nicht oder nicht mehr zum Unternehmen passt.

Im Experteninterview EB05 wird die Problemstellung hervorgehoben, dass viele Unternehmen ihre eigenen Prozesse nur unzureichend kennen. Dies ist ursächlich in einem Mangel in der Dokumentation oder darin, dass Prozesse sich im Verlauf der Zeit weiterentwickelt haben, initiiert durch Mitarbeiter selbst und das ohne Wissen des Unternehmens. Er weist auf die Wichtigkeit einer Dokumentation der tatsächlichen Prozesse hin. Das Aufbrechen von solchen veralteten Prozesse durch eine Standardisierung oder während der Implementierung der NCLC-Plattform kann zu weitreichenden negativen Auswirkungen für das Einführungsprojekt führen. Diese Auswirkungen können einen negativen Effekt auf Projektumfang, Kosten oder Zeitplan haben.

*„Ich habe es sehr oft erlebt und fast in jedem Unternehmen. Das ist halt auch das, was mich sehr überrascht hat, weil ich das von einem theoretischen Ansatz nie gedacht habe. Die meisten Unternehmen kennen ihre eigenen Prozesse nicht.“*

EB05, Pos. 33

### **Projektorganisationsform**

Die meisten Experten favorisieren eine Matrixorganisation oder eine Organisation ähnlich der Struktur einer Matrixorganisation. Dabei wird durch einen Experten hervorgehoben, dass wahrscheinlich der beste Projektoutput durch eine völlige Herauslösung der Projektteammitglieder aus der Organisationsstruktur erreicht werden würde. Dies wird allerdings durch den Experten selbst als ein unrealistisches Szenario angesehen. Dem Argument, dass bei 100 % Freistellung die besten Ergebnisse erzielt werden können, stehen jedoch verschiedene andere Experten entgegen, da dadurch die Möglichkeit verloren geht, Fachwissen und Feedback aus der Abteilung zu erhalten, was als wichtig für den Projekterfolg gewertet wird.

Allgemein wird eine Matrixorganisation angestrebt, bei der die Projektmitglieder zu einem gewissen Teil aus ihren originären Abteilungen herausgelöst werden. Es kann aber die Notwendigkeit entstehen, einige Projektmitglieder von Anfang an komplett für das Projekt herauszulösen, zum Beispiel für funktionale Aufgaben, wie Plattform-Enablement und Administration.

*„Aber in der Praxis, würde ich schon fast sagen, ist das utopisch, man wird vermutlich nie über eine Matrixorganisation hinauskommen, weil einfach die Unternehmen, in der Regel, sich dem Produktionsbetrieb, also den Operations, zugewandt haben und ein Projekt ist immer on top.“*

EB05, Pos. 35

Diese zu 100 % freigestellten Mitglieder könnten ebenfalls in einem eigenen Digitalisierungsteam zusammengefasst werden, das für die Kommunikation mit der IT-Abteilung zuständig ist und für eine Integration in Übereinstimmung mit den im Unternehmen vorhandenen IT-Strukturen.

In Bezug auf die Fachbereichsentwickler wird es, wie bereits erwähnt, als wichtig angesehen, dass sie in ihren Abteilungen verbleiben und dort das Fachwissen der Fachabteilung sammeln und die Fortschritte des Projektes in die Fachabteilung zurückspeiegeln. Dadurch kann ein Verständnis für das Projekt und dessen Zweck in den entsprechenden Abteilungen erzeugt werden, was für eine bessere Akzeptanz sorgen soll. Die Experteninterviews EB01 und EB08 heben dabei eine spezielle Form der Projektorganisation hervor, bei der sich die Projektmitglieder fast ausschließlich in ihren originären Abteilungen befinden und sich in spezifischen Zeitabschnitten in Workshops zusammenfinden, wo darüber gesprochen wird, wie der aktuelle Fortschritt der einzelnen Mitglieder ist, welchen Hürden sie gegenüberstehen und wie sie zusammen das Projekt weiterentwickeln. Mit den Projektergebnissen und entsprechenden Aufgabenstellungen für jedes Projektmitglied gehen diese zurück in ihre originären Abteilungen und bearbeiten die individuellen Aufgabenstellungen für das Projekt bis zum nächsten Workshop.

Ein wichtiger Faktor, auf den durch verschiedene Experten hingewiesen wurde, ist das Commitment. Es darf nicht die Situation entstehen, dass Projektmitarbeiter sich zwischen ihren Fachabteilungsaufgaben und den Projektaufgaben entscheiden müssen. Dieses Commitment gilt in gleichem Maße für die Projektmitarbeiter. Die Aufgaben für das Projekt sollten zu 100% erfüllt werden und nicht den Aufgaben der Abteilung untergeordnet werden. Dies wird unter anderem auch als Hürde für das gesamte Projekt betrachtet.

### **Wahl Vorgehensmodell**

Keiner der Experten hat einen reinen traditionellen Projektmanagementansatz für die Einführung von NCLC-Plattformen als favorisierte Form gewählt. In einigen Interviews wurde der Einsatz sogar als nachteilig für das Projekt aufgrund der starren und unflexiblen Eigenschaften des Ansatzes gesehen. Es werden ein hybrides Modell mit beiden Ansätzen oder ein rein agiles Vorgehen als bevorzugte Varianten genannt. Im Experteninterview EB07 wird darauf verwiesen, dass die Vorgehensweise jedoch zum Unternehmen passen sollte. Unternehmen sollten ein Vorgehen wählen, welches in der Organisation bereits bekannt ist, um dadurch zu vermeiden, dass eine doppelte Unsicherheit auftritt. Damit sind Unsicherheiten gemeint in Bezug auf die neue Plattform, die eingeführt werden soll und eine zusätzliche Unsicherheit in Bezug auf das neue Vorgehen.

### **Hybrides Modell mit Schwerpunkt traditionelles Vorgehen**

Herausgestellt werden sollte, dass im Experteninterview EB05 zwar ein hybrider Ansatz bevorzugt wird, allerdings mit einem Schwerpunkt auf einer Wasserfalllogik. Dies ist allerdings der einzige Experte, der diese Auffassung so explizit vertritt. Dieses Vorgehen begründet der Experte damit, dass die meisten Unternehmen, die so eine Plattform einführen, oft noch nicht oder nur in geringem Umfang mit agilen Methodiken gearbeitet haben. Dies stellt er als Hindernis heraus, weshalb er eine Wasserfalllogik als primäre Eigenschaft des von ihm bevorzugten Modells sieht. Dies ist allerdings, wie bereits erwähnt, eine Einzelmeinung und wurde so nicht von anderen Experten geteilt.

### **Agiles Vorgehensmodell**

Das agile Vorgehen wird von den meisten Experten vor allem in der aktiven Einführungsphase bevorzugt. Damit ist die Phase gemeint, wenn die Vorplanungen abgeschlossen sind und die Plattform in das Unternehmen implementiert wird, aufgrund der Tatsache, dass die Plattformen selbst sehr agile Eigenschaften besitzen und diese sich ständig

weiterentwickeln, auch während des Einführungsprojektes. Dadurch entsteht der Bedarf, dass ebenfalls im Projekt agil vorgegangen wird. Selbst Experten, die eher den hybriden Ansatz als das bevorzugte Modell angeben, benennen dabei einen Schwerpunkt im agilen.

*„Man hat ja quasi diese klassischen Sprints, im Scrum-Verfahren, im Scrum-Modell, und die ermöglichen einfach ein viel schnelleres Reagieren auf einen Wechsel der Anforderungen und ich habe selten eine so hohe Volatilität der Anforderungen im Verlauf der Implementierung erfahren, wie ich das mit Kunden, dieser Lösung hier, erfahre.“*

EB06, Pos. 41

Die Agilität der Plattform selbst sorgt dafür, dass sich die Anforderungen in einem ständigen Veränderungsprozess befinden. Es werden neue Features implementiert für die Plattform durch den Plattformbetreiber oder durch das Unternehmen werden Möglichkeiten der Plattformen entdeckt, welche im Vorfeld nicht klar waren. In einem traditionellen Ansatz würden an diesem Punkt starke Probleme entstehen, weil dort eine gewisse Planbarkeit des Projektumfangs vorausgesetzt wird. Es wird durch den Experten aus EB06 hervorgehoben, dass ebenfalls in einer agilen Methodik eine Art Phasenmodell durchaus denkbar ist, weshalb er ein Wasserfallmodell als nicht zielführend ansieht. Diese Phasen werden innerhalb jeder Iteration durchlaufen. Die Phasen Planung, Durchführung, Testing, Release und Problembehebung werden dabei immer wieder durchlaufen, bis die Plattform eingeführt ist und die Anforderungen erfüllt wurden. Ein ähnlicher Ansatz wird durch den Experten im Interview EB03 beschrieben. In diesem Fall erfolgt die Einführung ebenfalls agil und das Einführungsprojekt wird in verschiedene Themenkomplexe aufgeteilt, wie Changemanagement, technische Implementierung, Befähigung von MA. In diesen Thementeams, die analog zu Scrum-Teams verstanden werden können, werden die entsprechenden Projektmitglieder ihrer Aufgabe entsprechend zugeordnet und es erfolgt eine Bearbeitung der Themenschwerpunkte nach einem Scrum-ähnlichen Modell. Als initialer Einführungsschritt wird ein Sprint 0 durchgeführt, wo die Befähigung der Projektteilnehmer auf die agile Methodik und die Befähigung auf das Vorgehen erfolgt. Durch diesen Sprint 0 wird der Problemstellung entgegengewirkt, dass ein Unternehmen nicht mit einer agilen Methodik vertraut ist und Unsicherheiten soll so vorgebeugt werden. Ein weiterer Grund, weshalb der Experte dieses Vorgehen anwendet, ist, dass keine Abhängigkeit zu anderen Thementeams besteht. Alle Schritte können parallel durchgeführt werden, wodurch eine schnellere Einführung erfolgen kann. Der Experte EB01 stellt ebenfalls eine rein agile Methodik nach einem Sprint-System als bevorzugte Variante heraus, aus den vorher genannten Gründen. Einen weiteren Vorteil sieht er darin, dass eine schnellere Exit-Strategie realisiert werden kann, wenn während des Einführungsprojekts eine alternative Plattform gewählt wird. Der Experte beschreibt jedoch, dass gewisse Vorteile in einer hybriden Arbeitsweise liegen können, wenn Elemente aus den traditionellen Vorgehensweisen mitgenutzt werden. Dies kann sich vorteilhaft auf die nötigen Vorarbeiten wie Governance und Risikosteuerung auswirken, allerdings hebt der Experte ebenfalls hervor, dass dies ebenfalls in einer agilen Methodik umsetzbar ist. Angesichts dessen gibt es aus seiner Sicht keinen realen Mehrwert für das Projekt bei einem hybriden Vorgehen.

### **Hybrides Vorgehen Schwerpunkt agil**

Ein ähnliches Vorgehen, wie es in EB03 beschrieben wird, führt auch die Expertin aus EB04 an. Es wird von ihr hervorgehoben, dass es nicht sinnvoll ist, die Projektmitglieder zu 100 % aus ihren Abteilungen zu lösen, um den Fachbezug nicht zu verlieren. Eine

Voraussetzung für die bevorzugte Scrum-Methodik ist jedoch, dass alle Projektmitglieder zu 100 % committed sein müssen, weshalb eine Workshop-Arbeitsweise vorgeschlagen wird. Die Projektteilnehmer werden während der Workshop-Phasen, diese können als Sprint-Phasen gesehen werden, komplett aus ihren Abteilungen gelöst. Diese Arbeitsweise ermöglicht es, dass Zwischenlösungen regelmäßig präsentiert werden können und eine schnelle Rückkopplung über Feedback in das Projekt aufgenommen wird. Dies ermöglicht es, auf veränderte Anforderungen zu reagieren, was als vorteilhaft für die Qualität der Umsetzung angesehen wird. Diese Methodik wird durch die Expertin allerdings als Mischform bezeichnet, da in diesem Vorgehen trotzdem einige Anteile aus den traditionellen Ansätzen verwendet werden. Diese Anteile werden nötig in größeren Unternehmen, wo eine gewissen Planbarkeit aufgrund der Organisationsgröße nötig wird, für Zeitraum, Budget und Projektumfang. Als weiteren Vorteil, der generiert wird durch die Einbeziehung von traditionellen Projektmanagement-Bestandteilen, wird eine bessere Dokumentation hervorgehoben. Die Problematik einer mangelhaften Dokumentation wurde auch durch die anderen Experten mitaufgenommen.

*„Niemand, der ein Programm schreibt, liebt es, zu dokumentieren. Also es werden Dokumentationen gemacht, aber ich habe bisher noch keinen getroffen, der außer Lippenbekenntnissen zur Dokumentation wirklich dann auch dafür brennt, das, was er tut, da gut zu dokumentieren.“*

EB08, Pos. 35

Diese Methodik wird ebenfalls durch den Experten EB08 vorgeschlagen. Es wird kritisch betrachtet, ob zur Realisierung der Ziele Methodiken aus dem traditionellen Projektmanagement eingesetzt werden müssen oder ob diese nicht auch mit Methodiken aus den agilen Ansätzen, die den traditionellen Methodiken gleichwertig sind, umgesetzt werden können. In Ergänzung zu den anderen Experten wird im Experteninterview EB02 auf ein hybrides Modell verwiesen, mit klarem agilem Schwerpunkt. Es wird als begünstigend wahrgenommen, wenn im Vorfeld der aktiven Einführungsphase in einem phasenorientierten Vorgehensmodell eine Governance und andere planbare Aspekte aufgebaut werden. Der Zeitraum dafür sollte jedoch nur sehr kurz bemessen werden. Die tatsächliche Einführung sollte agil erfolgen. Um Unklarheiten bei einem agilen Vorgehen auszuräumen und die Machbarkeit zu testen, schlägt der Experte EB05 vor, agil zu beginnen und zu evaluieren, ob dieses Vorgehen für das Unternehmen funktioniert. Wird die agile Methodik nicht akzeptiert oder ist die Qualität nicht wie erwartet, kann schrittweise auf traditionelle Projektmanagement-Ansätze umgestellt werden, bis ein Maß erreicht ist, bei dem die gewünschte Akzeptanz und Qualität erreicht ist. Dieses Vorgehen macht der Experte stark vom Unternehmen abhängig und favorisiert, wie bereits beschrieben, eher einen Wasserfallsschwerpunkt.

### **Vorteile gewählte Vorgehensmodelle**

Durch die Experten werden verschiedene Vorteile bei den jeweils bevorzugten Vorgehensmodellen genannt. Vor allem beim agilen Vorgehen werden im Zusammenhang mit NCLC-Plattformen viele Vorteile gesehen. Dabei wird durch nahezu alle Experten die Möglichkeit hervorgehoben, schnell auf geänderte Anforderungen reagieren zu können sowie die Fähigkeit, ein schnelles Feedback zu erhalten. Wie bereits erwähnt wurde, werden die NCLC-Plattformen durch die Hersteller kontinuierlich weiterentwickelt. Mit einem agilen Vorgehen in Iteration, wie zum Beispiel in Sprints, ist es möglich, diese kontinuierlichen Weiterentwicklungen schnell im Prozess der Einführung zu berücksichtigen und diese zu testen. Im

Anschluss können diese durch die Projektteilnehmer über einen Feedback-Mechanismus bewertet werden. Dadurch kann es schnell zu einer Evaluierung kommen, wie diese neuen Entwicklungen der Plattform für das Unternehmen integriert werden können und welche Änderungen im Bereich Prozesse, Governance und Cybersecurity- Richtlinien nötig werden. Bei agilem Vorgehen wird ebenso die Möglichkeit der Parallelisierung von Aufgaben besonders hervorgehoben, welche sich bei der Vermeidung von Bottlenecks vorteilhaft auswirken kann. Im Gegensatz zu den meisten traditionellen Modellen ist es bei einem agilen Vorgehen möglich, verschiedene Teams mit Aufgaben zu betrauen, die diese individuell bearbeiten, ohne dabei auf kritische Arbeitspakete durch vorgelagerte Phasen warten zu müssen. Durch die im Verhältnis zu traditionellen Vorgehensweisen kurzen Sprints kann schnell ein MVP vorgestellt werden, welches wiederum schnell über Feedback bewertet werden kann. Dieses Feedback kann in die nächste Sprint-Phase direkt einbezogen werden. Diese Eigenschaft wird ebenfalls als ein motivatorischer Faktor bewertet, da Erfolge schnell sichtbar gemacht werden und die Projektteilnehmer den Mehrwert schnell erkennen können.

*„Das sind Kunden, die wollen ja Flexibilität, die wollen Schnelligkeit und es ist selten so, dass ich am Ende des Projekts die Anforderungen noch so habe, wie sie ursprünglich genannt wurden.“*

EB06, Pos. 41

Aus diesem flexiblen Vorgehen ergibt sich die Möglichkeit einer kostengünstigen Exit-Strategie, wodurch ein bereits benanntes Risiko bei den NCLC-Plattformen vermieden werden kann. Die praktische Eignung der Plattform kann frühzeitig evaluiert werden und ob sie den gewünschten Mehrwert für das Unternehmen bietet. Ein weiterer Vorteil, der für das agile Vorgehen im Einführungsprojekt genannt wird, ist die Möglichkeit, die gebildeten Strukturen des Einführungsprojektes teilweise für den aktiven Betrieb der Plattform zu übernehmen und diese agil weiterzuentwickeln. Die Projektteammitglieder haben am Ende des Einführungsprojektes bereits Erfahrung mit den agilen Vorgehensweisen gesammelt und sie können diese Erfahrung während des aktiven Betriebes der Plattform mit in die spätere Struktur übernehmen. Die agile Vorgehensweise bietet sich vor allem an, wenn die genaue Zielstellung nicht klar ist, was oft der Fall ist bei dieser Art von Projekten.

Im Gegensatz zum agilen Vorgehen wird beim traditionellen Vorgehen hauptsächlich der Aspekt der Planbarkeit als Vorteil für solch ein Projekt gesehen. Vor allem in den frühen Phasen des Projekts, bei dem die Prozess-, Governance-, Enabling- und Cybersecurity-Konzepte aufgebaut werden, wird ein Vorgehen in Phasen als förderlich angesehen. Der Experte EB01, Eckhard Hauenherm, hebt jedoch hervor, dass dies auch bei agilem Vorgehen möglich ist, mit gleicher Qualität des Outputs.

Als weiterer positiver Aspekt von klassischem Projektmanagement wird gesehen, dass bei der Anwendung eine stärkere Kontrolle und ein besseres Monitoring des Gesamtfortschritts möglich sind, wohingegen es bei agilen Vorgehensweisen passieren kann, dass dies durch die iterative kleinteilige Vorgehensweise verloren geht. Die Bedeutung dieses Vorteils wird vor allem für größere Organisationen als wichtig angesehen, wo Kennzahlen eine wesentlich größere Rolle spielen. Ein Vorteil, der durch die Experten genannt wurde, ist der Umstand, dass in vielen Unternehmen agile Vorgehensmodelle wesentlich unbekannter sind und deshalb ein höherer Befähigungsaufwand zu Beginn des Projekts notwendig sein kann, um diese erfolgreich zu nutzen. Dieser Befähigungsaufwand entfällt bei der Anwendung eines traditionellen Modells.

Die Vorteile von traditionellem Vorgehen werden durch die Experten hauptsächlich im Zusammenhang mit einem hybriden Vorgehen genannt. Die Kombination von beiden Vorgehensmodellen ermöglicht es, ein hohes Maß an Planbarkeit sowie die Fähigkeit eines ausführlichen Monitorings zu erhalten. Durch diese Planbarkeit bleibt der Gesamtüberblick im Fokus, was für große Einführungsprojekte als sehr wichtig angesehen wird. Demgegenüber werden beim Hinzufügen von einem agilen Anteil die Möglichkeit für schnelles Feedback und der motivatorische Faktor sowie die hohe Reaktionsgeschwindigkeit auf neue Anforderungen als Gewinn für das Projekt betrachtet. Die Auswirkungen der jeweiligen Vorteile sind jedoch stark davon abhängig, welcher Vorgehensansatz dominiert im Projekt und wie stark die jeweiligen Nachteile diesem entgegenwirken.

### **Nachteile gewähltes Vorgehensmodell**

Im Kontext der Anwendung der verschiedenen Vorgehensmodelle haben Experten auf bestimmte Hürden und Nachteile hingewiesen, auf die Unternehmen bei der Wahl des Vorgehensmodells achten sollten. Bei einem agilen Vorgehen wurde von den Experten die Problematik der Planungsunsicherheit genannt. Vor allem in großen Projekten kommt es zu der Situation, dass ein starker Projektdruck vorherrscht. Große Organisationen haben einen gewissen Zeit- und Kostenrahmen, den sie einhalten müssen. Dabei wird eine Relation hergestellt zwischen einer Verzögerung im Zeitplan und steigenden Kosten für das Projekt. Dies kann unter anderem ursächlich in der kleinteiligen Arbeitsweise sein, die durch ein agiles Vorgehen praktiziert wird. Es ist ebenfalls während der Einführung eine weiterführende Planung für den aktiven Betrieb der Plattform vorzunehmen. Dabei sollten Handlungsweisen geplant werden, zum Beispiel ein Releasemanagement. Eine weitere Hürde, die beim agilen Vorgehen zutage treten kann, entsteht, wenn das Projektteam einer hohen Volatilität der Projektteilnehmer unterliegt. Wechseln diese oft und fehlt die Konstanz im Team, kann es dazu kommen, dass bestimmte Aspekte der entwickelten Lösungen, zum Beispiel aus verschiedenen Sprints, nicht den Status eines Minimal Viable Products überschreiten und somit nicht das volle Potenzial erreichen, welches realisiert werden könnte. Dies kann in Situationen passieren, bei denen das MVP bereits einen Grad an Funktionalität erreicht hat, wo es für die Organisation nutzbringend eingesetzt werden kann. Es ist jedoch trotzdem nur ein MVP und erfüllt eventuell nicht die Qualitätskriterien. Eine Empfehlung der Experten ist, dass bei dieser Vorgehensweise darauf geachtet wird, personenunabhängig ein Ergebnis so lange zu entwickeln, bis es den bestmöglichen Nutzen für die Organisation erbringt. Es wird darauf hingewiesen, dass diese Probleme und weitere unter anderem aus einer fehlenden Erfahrung mit der Methodik hervorgehen können. Aufgrund der Tatsache, dass traditionelle Ansätze wesentlich älter und bekannter sind, kann es sein, dass ein Unternehmen noch keine Erfahrung mit dem agilen Vorgehen hat. Dies kann zu methodischen Fehlern führen, die negativ für den Projektverlauf sein können. Beispiele für Fehler, die durch mangelhaftes methodisches Vorgehen entstehen können, sind, dass ein Product Backlog nicht nach der Priorisierung der Projektgruppe gestaltet ist, sondern nach der Priorisierung von Einzelpersonen, die besonders dominant Einfluss auf den Product Backlog nehmen. Um dieses Problem zu vermeiden, sollte in Erwägung gezogen werden, eine Befähigungsphase zu Beginn des Projekts durchzuführen, wenn das Projektszenario darin besteht, dass das Verfahren einem Großteil der Projektteilnehmer noch unbekannt ist. Eine weitere Gefahr, die durch das schnelle und kleinteilige Arbeiten entstehen kann, ist, dass es zu einer mangelhaften Dokumentation kommt. Dies kann sich unter anderem negativ auswirken, wenn Aufgaben zwischen Personen übergeben werden und die neue Person einen erheblichen Mehraufwand hat, sich in die Arbeitsergebnisse der vorherigen Person

einzuarbeiten. Außerdem kann es zu Schwierigkeiten beim Projektmonitoring führen, wenn Vorgehensweisen unbekannt sind.

Die Nachteile, die in Bezug auf die traditionelle Vorgehensweise genannt wurden, grenzen sich auf ein zu starres Vorgehensmodell und eine zu lange Dauer ein. Vor allem bei NCLC-Plattformen ist das zu starre Vorgehen als nachteilig bewertet worden, da die Plattformen sich selbst schnell verändern. Demzufolge ist es schwer, eine längerfristige Planung für Ziele zu erstellen, da ein Mangel an Wissen über die zukünftigen Entwicklungen der Plattform vorherrscht. Es muss immer wieder neu evaluiert werden, wie sich Veränderungen auf die Plattform sowie auf die Organisation auswirken und wie diese dann entsprechend eingebunden werden. Dieses starre Vorgehen und der plangetriebene Charakter des Modells führen dazu, dass der initiale Kostenaufwand wesentlich höher sein kann, als das bei agilen Methoden der Fall ist. Gleichfalls wird auf die schlechtere Parallelisierbarkeit von Aufgabepaketen eingegangen, was sich ebenfalls nachteilig auf den zeitlichen Faktor des Projektes auswirken kann.

*„Ich glaube, bei der Einführung und der nachhaltigen Nutzung von Low-Code-Lösungen tun sich Unternehmen keinen Gefallen, an einem starren Wasserfallmodell festzuhalten. Lassen Sie mich ausführen, warum. Eine wesentliche Stärke von Low-Code-Lösungen zeichnet sich durch diese Flexibilität und Schnelligkeit aus.“*

EB06, Pos. 41

Die verschiedenen Nachteile der Einzelmodelle werden auch bei hybriden Vorgehensmodellen durch Experten benannt. Diese wirken sich, je nach Schwerpunkt des hybriden Vorgehens, entsprechend intensiv aus. Hier wird durch die Experten auf die Problematik des starren Vorgehens und der erhöhten Kosten aus dem Bereich der traditionellen Modelle sowie auf einen erhöhten Befähigungsaufwand eingegangen. Im Falle eines hybriden Modells müssen die Projektteilnehmer auf beide Methoden in gewissem Maße befähigt werden, abhängig davon, welche Bestandteile der verschiedenen Vorgehensmodelle eingesetzt werden. Dies kann sich ebenfalls negativ auf den zeitlichen Rahmen und den Kostenrahmen auswirken.

### **Tools Vorgehensmodelle**

Die verschiedenen Vorgehensmodelle bestehen aus sehr unterschiedlichen Tools und Methodiken, welche verwendet werden. Die Experten sehen bei den traditionellen Vorgehensmodellen eher die Phasen und den damit verbundenen Vorteil von Planungssicherheit sowie die Möglichkeit, sehr genau den aktuellen Fortschritt des Projektes zu messen, im Fokus. Dies kann vorwiegend in größeren Unternehmen wichtig sein, wo ein starres Controlling vorhanden ist und ein sehr genaues Monitoring erfolgen muss. Für diese Unternehmen, die eine solche Unternehmensorganisation aufweisen, bietet sich ein Wasserfallansatz an, allerdings wird hierbei durch die Experten explizit hervorgehoben, nicht in reiner Form, sondern als hybrider Ansatz. Vor allem zu Beginn des Projektes sehen einige Experten Vorteile in der Verwendung von Phasen und der dadurch gegebenen klaren Struktur. Durch diese frühen Phasen sollen die Rahmenbedingungen des Projektes, zum Beispiel die Governance, geklärt werden, bevor es in die aktive Umsetzung geht.

Im Bereich der agilen Methodiken und Tools wird mehrheitlich auf die Scrum-Methodiken verwiesen oder Methodiken, die diesen ähnlich sind. Vor allem das Arbeiten in Sprints, als Teil der Scrum-Methodik, wird als sehr positiv wahrgenommen. Auch die Möglichkeit von täglichen oder wöchentlichen kurzen Treffen, sogenannten Dailys, wird durch die Experten



als vorteilhaft gesehen, weil sie eine schnelle Möglichkeit bieten, Feedback einzuholen, Anforderungen neu zu planen oder Hürden, die existieren, anzusprechen. Als weiteres Tool werden die Möglichkeit von Retrospektiven und der Vorteil eines Wissensgewinns für spätere Sprints oder die Skalierung im Unternehmen genannt. Durch einige Experten wird auch die Möglichkeit, Kanban-Boards einzusetzen, als förderlicher Faktor für das Projekt gesehen, da dies eine vielversprechende Möglichkeit bietet, den Entwicklungs- und Applikationsprozess sichtbar und nachvollziehbar zu gestalten. Allerdings wird Kanban nicht als empfehlenswertes Vorgehensmodell für die Grundlage des Projektes bewertet.

Der Experte Peter Fey vertritt die Auffassung, dass die Tools in ihrer Anwendung nicht zwingend nur in dem einen oder anderen Modell anzutreffen sind. Als Beispiel skizziert er ein Phasenmodell, das in mehreren Iterationen durchgeführt wird.

### **Änderungswünsche**

Wie bereits in der vorangegangenen Ergebnispräsentation durch die Experten erklärt wurde, unterliegt ein solches Projekt Veränderungen. Diese Änderungen können von außerhalb des Projektes, durch die Organisation oder von den Projektmitgliedern selbst kommen. Die Quelle des Änderungswunsches sehen die Experten als wenig wichtig an.

*„Ich überlege gerade, ob es einen Unterschied macht, von wem eine solche Änderung her angetragen wird. Also ich sage mal, ob es einen Unterschied macht, ob es von intern kommt oder von extern. Ich glaube, es kommt gar nicht so sehr auf die Quelle drauf an, sondern auf die Art der Veränderung.“*

EB05, Pos. 55

Innerhalb der Analyse haben sich dabei zwei Vorgehen herausgebildet, die abhängig davon sind, ob der Wasserfallanteil der Vorgehensweise zugrunde liegt oder ob es teilweise oder komplett agil durchgeführt wird. Bei einem hohen Wasserfallanteil in einer hybriden Strategie sollten die Art und Intensität der Veränderung bewertet werden. Dieses Vorgehen wird im Interview EB05 herausgestellt. Ist die Änderung von geringem Umfang oder hat sie wenig Einfluss auf das Gesamtprojekt, sollte diese in das Product Backlog aufgenommen werden und entsprechend durch das Projektteam priorisiert werden. Ist es jedoch eine Änderung mit erheblichem Einfluss auf das Projekt, sei es eine regulatorische Maßnahme oder der Ausfall eines Projektteammitglieds, sollte klassisch ein Change Request erstellt und eine neue Planung mit den geänderten Anforderungen durchgeführt werden.

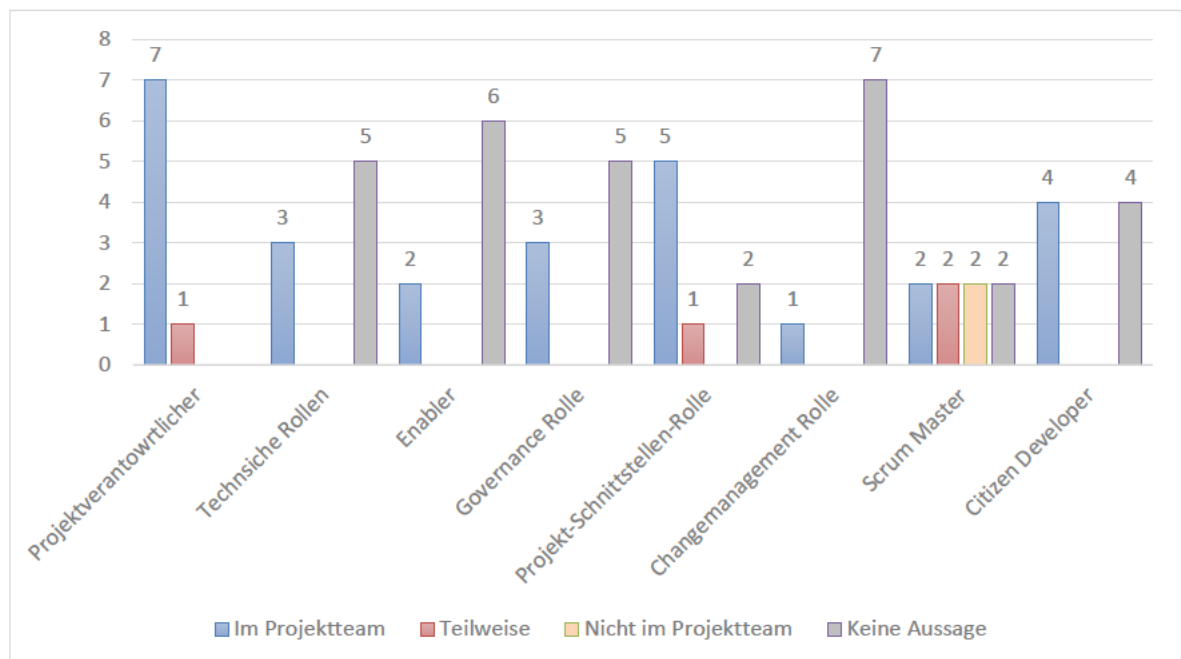
Die Experten, die ein agiles Modell oder ein hybrides Modell mit Schwerpunkt auf agilem Vorgehen bevorzugen, favorisieren eine reine Implementation in den Product Backlog. Bei diesem Vorgehen würde der Änderungswunsch in das Product Backlog übernommen und entsprechend durch das Projektteam priorisiert, unabhängig von der Größe der Änderung. Im weiteren Verlauf würde es, abhängig von der Priorität, in die Sprintplanung aufgenommen werden. Als ein Tool zur Visualisierung wird hier ein Kanban-Board durch den Experten EB08, Peter Fey, vorgeschlagen. Bei kleineren Projektteams, wo ein klassisches Sprint-Modell nicht immer angewendet werden kann, sollte über ein Ticketsystem nachgedacht werden, bei dem sich die einzelnen, mit der Umsetzung von Aufgaben verantwortlichen Personen diese Tickets einzeln zur Bearbeitung nehmen können. Bei großen Scrum-Teams sollte zusätzlich beachtet werden, dass größere Änderungsanforderungen, welche mehrere Teams betreffen, einen erhöhten Koordinationsaufwand nötig machen können. Diese Koordinationsaufgabe kann zum Beispiel durch eine übergeordnete Instanz ausgeübt werden, die die einzelnen Scrum-Teams organisiert. Durch die Expertin EB04, Simone Göttl, wird

ebenfalls darauf hingewiesen, dass es nicht passieren darf, dass der Backlog nur durch eine Person gestaltet wird, sondern durch das gesamte Team. Zu diesem Zweck ist es wichtig, eine Person im Team zu haben, welche die Ausführung der Methodik überwacht. Das wäre die Rolle des Scrum Masters.

Ein weiterer Punkt, der hervorgehoben wird durch den Experten Tino Janke, ist, dass Änderungswünsche durch das Team bewertet werden sollten. Das Projektteam muss diese Änderung umsetzen und nach außen vertreten, weshalb die Bewertung durch dieses durchgeführt werden sollte. Dafür ist es wichtig, dass die Teammitglieder die Änderung als Mehrwert empfinden. Ein weiterer Faktor bei der Bewertung von Änderungswünschen ist ein Soll-Ist-Abgleich mit den Projektzielen. Hat die Änderung negative Auswirkungen auf die Projektzielstellung, sollte ebenfalls darüber nachgedacht werden, ob diese in den Product Backlog übernommen wird. Wie die Änderungsanforderung Einfluss auf das Projekt hat, sollte ebenso klar mit dem Anforderer kommuniziert werden.

**Rollen und Stakeholder im Projekt**

In Abbildung 13 wurde dargestellt, welcher Experte das Vorhandensein von bestimmten Rollen im Projekt bewertet. In Tabelle 5 werden die Rollen beschrieben, wie sie durch die verschiedenen Experten dargestellt wurden. Diese sind nicht als generell relevant für alle Projektformen und Vorgehensmodelle zu verstehen. Diese Rollen können auch als wichtige Stakeholder außerhalb des Projekts vorhanden sein.



**Abbildung 13 - Aussagen der Experten zum Vorhandensein von Rollen im Projekt**

**Tabelle 5 - Ergebnispräsentation: Rollen und Stakeholder im Projekt**

Rollenname	Beschreibung
Projektverantwortlicher	Die Rolle eines Projektverantwortlichen in der Organisation wird als wichtig eingestuft. Dabei verwenden verschiedene Experten die Namen Projekt-Sponsor oder Product Owner, teilweise auch synonym, je nach Projektvorgehen. Diese Rolle sollte die Zielstellung des

	<p>Projekt definieren und die Anwendung der Plattformen im Unternehmen vorantreiben. Eine weitere wichtige Aufgabe dieser Rolle ist das Bereitstellen der nötigen finanziellen Ressourcen und zur Entscheidungsfindung. Der Projektverantwortliche sollte in höheren Hierarchieebenen das Projekt vertreten und Hindernisse für das Projekt beseitigen. Wichtig ist, dass sich der Projektverantwortliche für das Projekt verantwortlich fühlt und am Vorankommen des Projektes interessiert ist.</p>
Technische Rollen	<p>Durch einige Experten wird hervorgehoben, dass technische Rollen als Mitglieder des Projektteams wichtig sind. Es werden in diesem Zusammenhang technische Rollen genannt für die Themen Datenschutz, Datensicherheit und IT-Security. Weitere technische Rollen werden eingesetzt für die Support-Struktur der NCLC-Plattform und den Citizen Developer. Die IT-Abteilung sollte auch eingebunden werden, um eine Anbindung der Plattform an die Unternehmenssysteme durchzuführen.</p>
Enabler	<p>Die Funktion eines Enablers wird durch die Experten ebenfalls aufgeführt. Die Aufgabenbereiche liegen dabei in der Beratung und Befähigung der Citizen Developer und als Unterstützung für diese beim Einarbeiten in die Plattform. Ein weiteres Aufgabenfeld, das durch den Experten EB03, Fabian Gackstatter, gesehen wird für den Bereich Enablement, ist die Organisation der Plattform. Dabei wird durch die Teammitglieder evaluiert, welche Rollen im aktiven Betrieb benötigt werden, welche technischen Beschränkungen vorhanden sind und welche Rahmenbedingungen etabliert werden müssen.</p>
Governance-Rolle	<p>Wie bereits hervorgehoben wurde, wird durch die Experten eine funktionierende Governancestruktur als ein Erfolgskriterium für den Betrieb der NCLC-Plattform definiert. Zu diesem Zweck wird eine Governance-Rolle von einigen Experten als wichtig im Projekt erachtet. Diese Rolle sollte die Strukturen innerhalb des Unternehmens aufbauen und die Governance-Rahmenbedingungen bekannt machen.</p>
Rolle als Projektschnittstelle	<p>Eine weitere Rolle, die durch die Experten beschrieben wird, ist ein Verantwortlicher für die Kommunikation mit dem Unternehmen und als Schnittstelle zwischen den einzelnen im Projekt vorhandenen Teams. Durch die Funktion als Schnittstelle dient diese Rolle auch dazu, Entscheidungen des Teams in das Unternehmen zu tragen und diesen dort eine entsprechende Wichtigkeit zu geben. Diese Rolle sollte auch bei Problemen innerhalb des Projektes zwischen Projektmitgliedern vermitteln. Für größere Projekte, bei dem viele Abteilungen und Personen involviert sind, kann diese Rolle durch ein PMO, Project Management Office, erfüllt werden.</p> <p>Diese Rolle wird nicht zwingend vom Projektleiter ausgeführt, sondern kann auch durch den Product Owner durchgeführt werden.</p>
Changemanagement-Rolle	<p>Das Thema Changemanagement wird als Rolle im Projektteam durch den Experten Fabian Gackstatter hervorgehoben. Diese Projektmitglieder kümmern sich darum, dass die Organisation auf die Veränderung vorbereitet wird, die durch die Einführung der NCLC-Plattform entsteht. Dies betrifft unter anderem die Themenfelder Vorbereitung</p>

	von Trainings, Veränderung der Supportorganisation auf technischer Ebene, Verknüpfung der Plattform mit dem Unternehmen und der Etablierung einer Support-Struktur für die Citizen Developer.
Scrum Master	Der Scrum Master wird bei agilen Vorgehensweisen nach dem Scrum-Modell eingesetzt. Einige Experten sehen jedoch dieses Tätigkeitsfeld abhängig von der Größe des Projektes und gehen davon aus, dass sie bei kleinen bis mittleren Projekten durch den Projektleiter mitabgedeckt werden kann. Bei großen Projekten wird eine eigene Scrum-Master-Rolle im Projekt als nötig erachtet. Bei kleinen Projekten kann die Aufgabe des Scrum Masters auf den Projektleiter aufgeteilt werden, weil dort viele Probleme, die im Projekt entstehen, durch das Projektteam selbst gelöst werden können, weshalb eine dedizierte Scrum-Master-Rolle als nicht zwingend sinnvoll erachtet wird.
Citizen Developer	Es wird durch einige Experten die Notwendigkeit erläutert, Citizen Developer in das Projektteam zu integrieren. Dies kann förderlich sein, damit die Citizen Developer ein Verständnis für die Notwendigkeit einer Governance entwickeln können, welche normalerweise eher als Hürde durch diese wahrgenommen wird. Eine weitere Aufgabe dieser Rolle ist es, die für das Unternehmen relevanten Anwendungsfälle aus den Fachabteilungen mit in das Projekt zu bringen und dadurch die Anpassung der Plattform auf die Unternehmensbedürfnisse zu verbessern.
Berater	Die Berater-Rolle wurde durch einen Experten genannt und kann nicht als allgemein gesehen werden. Die Berater-Rolle ist ein Projektmitglied, das bereits Erfahrung mit der Thematik oder dem Vorgehen hat, und die bei der Administrierung und dem Aufbau der Governance unterstützt.
Compliance	Die Rolle eines Compliance-Verantwortlichen wurde als zusätzliche Rolle ebenfalls nur durch die Expertin Simone Göttl hervorgehoben. Die Beschreibung der Aufgaben wird allerdings bei den technischen Rollen ebenfalls teilweise beschrieben. Die Rolle ist verantwortlich für Datenschutz und Informationssicherheit.
Mitarbeitervertretung	Es wird hervorgehoben, dass die Rolle der Personalvertretung bei solchen Einführungsprojekten nicht unterschätzt werden darf. Aufgrund der Tatsache, dass NCLC-Plattformen weitreichenden Einfluss auf die Mitarbeiter haben, sollte die Mitarbeitervertretung involviert werden. Es besteht ein Mitspracherecht für diese und dadurch können diese das Projekt verzögern.
Citizen Developer Strategist	Der Citizen Developer Strategist wurde durch den Experten Peter Fey benannt und erfüllt die Aufgabe, zu planen, welche Applikationen später im Betrieb entwickelt werden sollen. Im Gegensatz zum Citizen Developer, der dies aus der Fachabteilung heraus tut, plant der Strategist dies vonseiten des Unternehmens.

### Wichtige Stakeholder außerhalb des Projektes

Die Experten haben im Verlauf des Interviews unterschiedliche Stakeholder benannt, die als wichtige Stakeholder außerhalb des Projektes definiert wurden. Diese Stakeholder können aber ebenfalls Teil des Projektteams sein, weshalb es zu Überschneidungen mit den

Rollen und Stakeholdern innerhalb des Projektes kommen kann. Die Experten haben unter anderem auf die wichtigen organisatorischen Vorgesetzten hingewiesen. Die IT-Leitung wurde dabei explizit benannt. Diese muss Ressourcen bereitstellen und bei der strategischen Platzierung des Themas unterstützen. Zu diesem Zweck ist es wichtig, diese eng einzubinden, da sie ein Erfolgsfaktor sein kann. Ebenfalls als wichtige Vorgesetzte wurden die Bereichsleiter und Linienvorgesetzten der am Projekt beteiligten Mitarbeiter benannt. Diese müssen personelle Ressourcen bereitstellen sowie diesen Ressourcen die Kapazitäten zur Verfügung stellen, um die Aufgaben, die sie aus dem Projekt erhalten, bearbeiten zu können. Die Geschäftsleitung wird in einer ähnlichen Funktion benötigt. Die Einführung von NCLC-Plattformen kann Einfluss auf die gesamte Arbeitsweise des Unternehmens haben, weshalb die Geschäftsleitung hier als ein wichtiger Stakeholder angesehen werden muss. Hierbei sollte die Geschäftsleitung die Thematik mit in die strategische Planung des Unternehmens aufnehmen, entsprechende Ressourcen für das Projekt freigeben und ebenfalls als Unterstützung im Unternehmen zur Motivation für das Thema wirken.

Aufgrund der unternehmensweiten Auswirkungen werden verschiedene weitere dedizierte Abteilungen von Auswirkungen betroffen sein, weshalb diese im Stakeholderkreis berücksichtigt werden sollten. Im Experteninterview EB01 wird in diesem Zusammenhang die Personalabteilung benannt. Die Einführung von Citizen Development und alle zugehörigen Aufgaben beeinflussen unter anderem die Personalorganisation, in Form von Arbeitsplatzbeschreibungen sowie arbeitsvertraglichen Details, wie Aufgabenbeschreibungen. Diese sollten in den Informationsfluss eingebunden sein und es sollte das Feedback dieser Abteilung einbezogen werden. Dadurch soll eventuellen Problemstellungen vorgebeugt werden, die dort entstehen können. In den Experteninterviews EB04 und EB05 wird auf die Notwendigkeit hingewiesen, Abteilungen einzubinden, welche für den Datenschutz und die IT-Security zuständig sind. Dies kann alternativ ebenfalls dadurch erfolgen, dass ein Mitarbeiter der Abteilung ein Teil des Projektteams ist. Die IT-Abteilung ist ebenfalls ein wichtiger Stakeholder, weil die Einführung einen großen Einfluss auf die Arbeitsweise der Abteilung hat. Die Fachbereiche, in denen die NCLC-Plattform angewendet werden soll, werden ebenfalls als Stakeholder gesehen, die außerhalb des Projektes stehen, allerdings eingebunden werden sollten.

Ein weiterer Stakeholderkreis, der eine essenzielle Bedeutung für das Projekt hat, wird in den Experteninterviews EB03 und EB04 angeführt. Die Einführung einer NCLC-Plattform hat weitreichende Auswirkungen auf die Mitarbeiter, weshalb Mitarbeitervertretungen in gewissen Bereichen ein Mitspracherecht haben. Werden diese nicht entsprechend eingebunden als wichtige Stakeholder, können diese das Projekt verzögern oder sogar stoppen. Dies kann zum Beispiel darin begründet sein, dass neue Anwendungen, die im Unternehmen verwendet werden, initial durch den Betriebsrat genehmigt werden müssen. Dies ist völlig unpraktikabel aufgrund der großen Anzahl der Anwendungen, die Citizen Developer entwickeln können. Hier müsste eine Veränderung von Prozessen und eventuell von Betriebsvereinbarungen vorgenommen werden.

Die Supportorganisation, welche die technische Struktur der Plattform gewährleistet und im Fehlerfall als Ansprechpartner dient, sollte in jedem Fall eingebunden werden. Durch die Einführung der Plattform entsteht eine völlig neue Entwicklergruppe mit völlig neuen Anforderungen und Fragen. Diese müssen durch die Supportorganisation bearbeitet werden können. Hier sollte eine Entscheidung getroffen werden, ob eine dedizierte Supportorganisation geschaffen wird, die dediziert für die Citizen Developer verantwortlich ist oder ob die bestehende Supportorganisation mit neuen Ressourcen erweitert wird.

Es wird durch den Experten EB01, Eckhard Hauenherm, die Möglichkeit aufgeworfen, dass auch externe Dienstleister relevant sein können, wenn die Dienstleistung für das

Unternehmen durch die Plattform beeinflusst wird. Der Prozessverantwortliche im Unternehmen ist ebenfalls ein Stakeholder, der berücksichtigt werden sollte, da es seine Aufgabe ist, die bestehenden Prozesse an die neuen Anforderungen anzupassen, die durch die Plattform entstehen. Der Experte EB08, Peter Fey, sieht die Notwendigkeit eines externen Testers. Dieser sollte die im Einführungsprojekt entwickelten ersten Applikationen dahingehend testen, ob sie sich für den Einsatz im Unternehmen eignen oder ob Veränderungen vorgenommen werden müssen, was eine Anpassung der Plattform zur Folge haben kann. Es sollte bei einer Skalierung in das ganze Unternehmen ebenfalls über eine Informationspolitik nachgedacht werden, die es allen Mitarbeitern möglich macht, sich über dieses Thema zu informieren und somit frühzeitig Motivation für das Thema zu schaffen.

### **Skalierung des Pilotprojekts**

Die Experten nennen verschiedene Maßnahmen zur Vorbereitung der Skalierung, welche nötig sind oder begünstigend wirken können. Ein wichtiger Punkt, der durch die Experten genannt wird, ist eine erfolgreiche Kommunikation aus dem Einführungsprojekt im Vorfeld der Skalierung. Aufgrund der weitreichenden Veränderungen, die durch die Plattform in den Arbeitsweisen entstehen können, sollte das Projektteam früh im Projekt einen Informationsfluss in die Abteilungen etablieren. Es sollten ebenfalls, sobald es möglich ist, erste Ergebnisse präsentiert werden. Zu diesem Zweck können, je nach Größe des Unternehmens, Lösungen für 1 – 3 Anwendungsfälle des Unternehmens entwickelt werden und diese werden zu Demonstrationszwecken im Vorfeld den Abteilungen zugänglich gemacht. Eine weitere Möglichkeit, die Akzeptanz innerhalb der Abteilung zu fördern, besteht darin, Multiplikatoren auszubilden, die in den Abteilungen vor Ort ihr erlangtes Wissen weitergeben und die Vorstellung der Anwendungsfälle übernehmen. Diese Multiplikatoren haben einen weiteren Vorteil für das Skalierungsprojekt. Sie können während der Skalierung in die Abteilungen ebenfalls die Rolle von Trainern übernehmen. Dadurch wird der initiale Wissenstransfer für die Nutzer unabhängig von den Plattformbetreibern und die Ansprechpartner sind vor Ort im Unternehmen erreichbar. Zur Organisation der Skalierung empfiehlt der Experte EB03, Fabian Gackstatter, die Implementierung einer dedizierten Einheit, welche sich mit den für die Skalierung nötigen Prozessen beschäftigt. Diese Prozesse müssen definiert und entwickelt werden, weil sie eine Grundlage für die Skalierung bilden. Dabei sollten die Prozesse die Organisation auf eventuelle Szenarien vorbereiten, die während der Skalierung auftreten können. Als Beispiele werden das Lizenzmanagement, das Monitoring der Plattform und wie mit Situationen umgegangen wird, wenn unberechtigte Datenzugriffe erfolgen, aufgeführt. Solche Szenarien sollten im Vorfeld mitberücksichtigt werden und Teil der Vorbereitung sein, um im schlimmsten Fall, dass diese eintreten, schnell reagieren zu können. Berücksichtigt werden sollten ebenfalls in der Vorbereitung die relevanten Schnittstellen zu anderen Abteilungen.

Während der aktiven Skalierungsphase werden durch die Experten verschiedene Möglichkeiten gesehen, wie dies geschehen kann. Eine Vorgehensweise ist eine sequenzielle Skalierung, das heißt Abteilung für Abteilung. Bei dieser Vorgehensweise werden die Abteilungen einzeln an die Plattform angeschlossen und die Mitarbeiter befähigt. Dadurch ist es möglich, aus den Erfahrungen zu lernen, die bei der Skalierung in vorhergehenden Abteilungen gesammelt werden konnten. Als Nachteil dieses Vorgehens wird gesehen, dass die Skalierungsphase längere Zeit benötigt als bei einem parallelen Vorgehen.

Bei einem parallelen Vorgehen erfolgt der Roll-out der Plattform in mehreren Abteilungen gleichzeitig. Vorteil dieser Methodik ist, dass die Skalierung wesentlich schneller erfolgt als in einem sequenziellen Vorgehen. Eine Problemstellung, die jedoch dadurch entstehen

kann, ist, dass somit eine hohe Ressourcen-Abfrage für Schulungsangebote und technische Betreuung entsteht. Als Lösungsansatz für diese Problematik wird die Möglichkeit der Automatisierung von Prozessen vorgeschlagen. Als Beispiel wird im Experteninterview EB02 die Möglichkeit aufgezeigt, Schulungen nicht als Präsenzveranstaltungen mit einem Trainer auszurichten, sondern diese als Videokurs anzubieten. Durch diese Möglichkeit können die Nutzer sich selbst befähigen, zu dem Zeitpunkt, wo sie es benötigen.

Als weiteres Vorgehen, welches die Vorteile des sequenziellen und des parallelen Skalierens vereint, wird ein sequenziell-paralleler Ansatz angeführt. Bei diesem Ansatz wird von der Prototypenabteilung beginnend erst schrittweise in ein paar ausgewählte andere Abteilungen skaliert. Während dieser ersten Skalierung wird Feedback gesammelt und die bestehenden Prozesse werden überprüft. Danach werden von diesen Abteilungen ausgehend weitere Abteilungen an die Plattform angeschlossen. Diese Abteilungen können parallel skaliert werden aus Abteilungen, wo der Roll-out bereits stattgefunden hat. In diesem Zusammenhang kann ein Multiplikatoren-Konzept nützlich sein, in dem diese aus den vorherigen Abteilungen die Erfahrungen in die neuen Abteilungen einbringen. Dadurch kann die höhere Skalierungsgeschwindigkeit der parallelen Skalierung mit den Vorteilen des sequenziellen Skalierens verbunden werden. Parallel zur Skalierung sollte ein kontinuierlicher Verbesserungsprozess implementiert werden, um die Veränderungen der Plattformen durch Updates mit in den Skalierungsprozess einzubringen und neue unternehmensrelevante Anwendungsfälle zu evaluieren.

### **Befähigung von Mitarbeitern**

Es werden durch die Experten vier wesentliche Aspekte genannt, welche für die Befähigung von Mitarbeitern geeignet sind. Eine Befähigungsart ist die Wissensvermittlung über Schulungen. Es werden dabei durch die Experten verschiedene Arten von Schulungen genannt, unter anderem eine Einführungsschulung wird im Experteninterview EB01 angeführt. Diese Schulungsmaßnahme dient dazu, eine erste Befähigung für die Plattform bereitzustellen und den Mitarbeitern die Unternehmensrichtlinien in Bezug auf Governance-, IT-Security- und Datenrichtlinien zu vermitteln. Diese Schulung kann als Einstiegshürde definiert werden, damit nur befähigte Mitarbeiter mit der Plattform arbeiten, die die Plattform und die Richtlinien kennen. Der Experte sieht es als vorteilhaft an, wenn Schulungen in einer Form durchgeführt werden, in welcher der Trainer als Ansprechpartner verfügbar ist, um Fragen zu beantworten. Schulungen sind nicht zwingend als starrer Frontalunterricht zu verstehen, sondern können durchaus einen Workshop-Charakter haben. Ein Beispiel wird im Experteninterview EB03 beschrieben. Dieser Workshop sieht vor, dass die Mitarbeiter mit Problemstellungen, die sie in ihrer Arbeit haben, in den Workshop kommen und dann in Gruppen mit anderen Mitarbeitern eine Lösung entwickeln. Ein Trainer ist als Ansprechpartner vor Ort, um unterstützend zu helfen. Eine Schulung reicht jedoch aus Sicht des Experten EB05, Christian Rohrbeck, nicht aus, da die meisten Lernerfahrungen während des tatsächlichen Arbeitens mit der Plattform erlangt werden. Zur Lösung dieser Problematik können Multiplikatoren herangezogen werden. Durch die Experten wird herausgestellt, dass es ein erweitertes Schulungsangebot zu verschiedenen Themen geben sollte, eventuell mit Zertifizierungsprozess und Prüfung.

Eine weitere Form der Befähigung kann über Multiplikatoren erfolgen. Diese Multiplikatoren sind entweder erfahrene Nutzer mit einem hohen Maß an Wissen für die Plattform oder Nutzer, welche bereits in den Abteilungen ein hohes Verständnis für IT-Logiken bewiesen haben. Nutzer aus den Abteilungen, die keine Erfahrung mit der Plattform haben, sich aber als Multiplikatoren eignen, werden in einem initialen intensiven Training auf die Plattform

befähigt. Sind die Multiplikatoren fertig ausgebildet, können diese verschiedene Aufgaben erfüllen, zum Beispiel andere Mitarbeiter im Unternehmen auf die Plattform befähigen oder in Workshops und beim Aufbau sowie Betrieb von Communitys unterstützen. Ein positiver Aspekt, der dadurch generiert wird, ist, dass Unternehmen unabhängig von der Wissensvermittlung durch den Plattformbetreiber oder externe Trainer werden. Multiplikatoren können zur Verbesserung der Motivation beitragen, indem sie die positiven Aspekte der Plattform vorführen und zeigen, welche Mehrwerte dies für die einzelnen Abteilungen haben kann.

*„In jeder Abteilung gibt es, manche nennen es Champions, manche nennen es Leuchttürme, ein oder zwei oder auch drei, je nachdem, wie groß die Abteilung oder der Bereich ist, Leuchttürme identifizieren, die auch die frohe Botschaft vorantragen und auch wissen, was diese Tools können und dort sozusagen Botschafter sind, das ist glaube ich eins.“*

#### EB04, Pos. 51

Die Multiplikatoren sind besonders wichtig in der Anfangszeit nach dem Roll-out in der Abteilung, um als Ansprechpartner bei Problemen zu helfen und die Nutzer in der Abteilung zu unterstützen. Als wichtige Voraussetzung sieht der Experte EB03, Fabian Gackstatter, die Unternehmenskultur. Ist diese primär darauf ausgerichtet, dass die Mitarbeiter eher als Konsumenten fungieren und die IT-Abteilung alle sonstigen Aufgaben übernimmt, wird es schwer, dieses Konzept zu etablieren. Es sollte eher eine Unternehmenskultur angestrebt werden, bei der die Mitarbeiter sich gegenseitig unterstützen und aktiv an der Plattform mitwirken wollen. Dies ist zu favorisieren, da mit einer auf die IT-Abteilungen ausgerichteten Struktur diese erneut zu einem Bottleneck werden kann.

Communitys sind eine weitere Form der Befähigung. Communitys sind Strukturen, in denen die Nutzer der Plattform zusammen kommunizieren können. Diese Communitys können intern im Unternehmen etabliert werden oder extern zum Beispiel in sozialen Netzwerken. Hier können verschiedene Fragestellungen platziert werden, vom Einsteigerlevel bis zu Fragen für eine Lösungsentwicklung. Communitys sollten bei Problemstellungen die erste Instanz sein, wo Nutzer nach Hilfe suchen können. Diese Communitys können im Idealfall mithilfe der Multiplikatoren aufgebaut werden, welche ebenfalls dort administrativ mitwirken können. Die IT-Abteilung kann dabei beim Aufbau der technischen Struktur helfen. Communitys sollen jedoch nicht nur der Problemlösung dienen, sondern zusätzlich für einen Wissenstransfer zur Verfügung stehen.

Einige Experten weisen explizit auf die Wichtigkeit von Selbstbefähigung hin. Der Experte EB02, Robin Rosengrün, stellt heraus, dass Mitarbeiter auf unterschiedliche Art lernen und für jeden eine andere Art zu lernen erfolgreich sein kann. Eine dieser Arten, sich selbst zu befähigen, ist das Lernen mit Videokursen. Diese Form der Wissensvermittlung hat den Vorteil, dass die wichtigen Informationen dann zur Verfügung stehen, wenn sie gebraucht werden. Der Nutzer kann selbst wählen, wann er diese abrufen. Das kann dazu führen, dass eine höhere Motivation besteht, sich selbst weiterzubilden. Steht ein Nutzer vor einer Problemstellung, kann er gegebenenfalls den Videokurs aufrufen und sich das entsprechende Video mit den wichtigen Informationen ansehen, wenn er diese benötigt. Diese Form der Problemlösung ist der Trial-and-Error-Methode vorzuziehen, weil diese eine wesentlich langsamere Lösungsbereitstellung bietet. Bei Trial-and-Error wird so lange versucht, das Problem zu lösen, bis der erreichte Ist-Zustand den Soll-Zustand erfüllt. Dies kann, je nach Level des Anwendungsentwicklers, einen hohen Zeitaufwand bedeuten. Die Form der Befähigung über Videokurse kann mit der Befähigung über Workshops kombiniert werden.



Die Nutzer eignen sich das Wissen in ihrem Tempo an und in gemeinsamen Workshops mit einem Trainer werden dann Unternehmensanwendungsfälle gemeinsam gelöst. Dort würde für Rückfragen, die entstanden sind, ein Trainer zur Verfügung stehen, der diese beantworten kann.

### **Qualitätssicherung Projekt**

Durch die Experten werden verschiedene Faktoren benannt, die für eine gute Qualität in Projekten nötig sind. Dabei wird im Experteninterview EB01 auf den Faktor Motivation eingegangen. Zur Gewährleistung der Motivation während des Projekts sollten Erfolge gefeiert und regelmäßig ein Zustrom an neuen Anforderungen generiert werden. Ein Mangel an Motivation kann dazu führen, dass Mitarbeiter nach Abschluss der zugewiesenen Aufgabe das Projekt verlassen und danach nicht mehr an der Weiterentwicklung mitwirken. Dadurch kann bereits erlangtes Projektwissen verloren gehen. Ein weiterer Faktor ist, dass akzeptiert werden muss, dass nach der Einführung eine gewisse Lernkurve erforderlich ist, bis das Projekt voll funktionsfähig ist. Aus diesem Grund sollte vorrangig in der Phase, wenn die ersten Anwendungen produktiv eingesetzt werden, ein erhöhtes Monitoring betrieben werden. Zu diesem Zweck können Qualitätschecks eingeführt werden, welche zu definierten Zeitpunkten Soll-Ist-Abgleiche durchführen. Teil dieser Qualitätschecks sollten Rückkopplungsprozesse sein, die aufzeigen, ob vermitteltes Wissen richtig aufgenommen wurde und ob dieses Wissen entsprechend angewendet wird. Mit diesem Mittel sollte ebenfalls überprüft werden, ob die definierten Governance-, Sicherheits- und Datenrichtlinien eingehalten werden. Die Überprüfung der Qualität der Anwendungen, in Bezug auf die Funktionalität, sollte durch die Nutzer selbst erfolgen. Danach sollte als zusätzliche Testinstanz überprüft werden, ob alle Unternehmensrichtlinien eingehalten wurden. Als Teil der Qualitätssicherung sollten regelmäßige Feedbacks durch Nutzer und Entwickler eingeholt werden, um festzustellen, dass die Umsetzung der Plattform den Anforderungen des Unternehmens und der Nutzer entspricht.

### **Erfolgsmessung des Projekts**

Ein wichtiger Faktor, der mit Abschluss eines Projekts bedeutend wird, ist die Erfolgsmessung für das Projekt. Bei der Erfolgsmessung kann in zwei Kategorien unterschieden werden, und zwar in quantitative Erfolgsmessung sowie qualitative Erfolgsmessung.

Bei der quantitativen Erfolgsmessung kann auf der einen Seite aus Projektsicht ermittelt werden, ob dieses in Time, in Scope und in Budget ist. Für die Plattform selbst können verschiedene KPIs gebildet werden, um festzustellen wie gut die Plattform funktioniert und ob die Zielstellung erreicht wird. In der folgenden Tabelle 6 sollen mögliche KPIs vorgestellt werden, welche durch die Experten genannt wurden. Dabei erfüllt die Tabelle nicht den Anspruch auf Vollständigkeit aller möglichen KPIs, sondern sie soll nur einen Einblick ermöglichen.

**Tabelle 6 - Mögliche KPIs zur Erfolgsmessung**

Kategorie Kennzahlen	Kennzahlen
Anwendungskennzahlen	<ul style="list-style-type: none"> <li>- Anzahl der erstellten Anwendungen</li> <li>- Anzahl produktiv genutzter Anwendungen</li> <li>- Anzahl ungenutzter Anwendungen</li> <li>- Anzahl unfertiger Anwendungen</li> <li>- Durchschnittliche Entwicklungsdauer</li> <li>- Kosten pro Anwendung</li> <li>- Gesamtaufwand zur Entwicklung einer Anwendung</li> </ul>
Nutzerkennzahlen	<ul style="list-style-type: none"> <li>- Anzahl aktiver Nutzer</li> <li>- Anzahl Nutzer, die bestimmte Anwendungen nutzen</li> <li>- Anzahl Nutzungsaufrufe der Anwendung</li> </ul>
Supportstrukturkennzahlen	<ul style="list-style-type: none"> <li>- Anzahl Mitglieder Communitys</li> <li>- Anzahl Beiträge in den Communitys</li> <li>- Antwortfrequenz auf Beiträge in Communitys</li> <li>- Anzahl Supporttickets</li> <li>- Anzahl Teilnehmer in Workshops</li> <li>- Anzahl Teilnehmer in freiwilligen Workshops</li> </ul>
Für den Fall, dass ein Anwendungskatalog, analog zu einem App Store, eingeführt wurde:	<ul style="list-style-type: none"> <li>- Anzahl Anwendungen im Anwendungskatalog</li> <li>- Messung der Frequenz des Bereitstellens von Anwendungen</li> </ul>

Als qualitative Größen für die Messung des Erfolgs wurden durch die Experten verschiedene Methoden hervorgehoben. Qualitative Größen können über Retrospektiven, strukturierte Interviews oder Umfragen erlangt werden. Durch diese Evaluierungstools sollte ermittelt werden, ob Wissen im Unternehmen aufgebaut werden konnte und ob der Nutzungsumfang der Plattform demjenigen der Zielstellung gleicht, zudem: wie der Einfluss auf die unternehmensinternen Prozesse ist sowie ob und wie die Plattform im produktiven Einsatz genutzt wird. Eine weitere qualitative Größe, die durch die Experten benannt wurde, ist die Fragestellung, ob durch die Automatisierung dieser Prozesse weniger Fehler in den Prozessen entstehen. Im Weiteren sollte ebenfalls evaluiert werden, ob Risiken durch mangelhafte Rahmenbedingungen entstanden sind. Ein weiterer Aspekt ist, wie mit Problemen beim Umgang mit der Plattform vorgegangen wird. Welche Problemstellungen sind während des aktiven Betriebs der Plattform entstanden und werden die bereitgestellten Supportstrukturen, wie Communitys und User Helpdesk, genutzt? Werden die Supportstrukturen in der vorgesehenen Weise genutzt oder entfällt zum Beispiel ein Großteil der Supportanfragen auf die IT-Abteilungen? Es sollten ebenfalls Nutzerzufriedenheitsanalysen durchgeführt werden, um zu verhindern, dass die Nutzer aufhören, die Plattform zu nutzen.

## 8 Diskussion

In diesem Kapitel sollen Handlungsempfehlungen für die Einführung einer NCLC-Plattform entwickelt werden. Zu diesem Zweck werden die im Kapitel 7 vorgestellten Ergebnisse der Interviews, unter Einbeziehung der theoretischen Vorarbeit aus dem Kapitel 2, zusammengefasst und Handlungsempfehlungen abgeleitet. Dadurch sollen Auswirkungen auf das Unternehmen und ein mögliches Vorgehen für die Einführung aufgezeigt werden. Zu diesem Zweck werden die anhand der Experteninterviews präsentierten Vorgehensmodelle vorgestellt und miteinander verglichen. Aus dieser Vorstellung soll eine Handlungsempfehlung für ein mögliches Vorgehen erfolgen. Referenzen zu den entsprechenden Literaturkapiteln und den Codes werden aus Gründen der besseren Lesbarkeit in der Fußzeile verwendet.

### 8.1 Auswirkungen von NCLC-Plattformen auf Unternehmen

#### 8.1.1 Veränderte Arbeitsweisen

Die Einführung von NCLC-Plattformen hat einen bedeutenden Einfluss auf viele Bereiche von Unternehmen. Die Veränderungen betreffen dabei die Prozesse, Richtlinien, Arbeitsabläufe in Unternehmen und verändern die Aufgabenfelder von Abteilungen. Ursächlich für die Veränderungen ist die Funktionsweise einer solchen Plattform. In Abgrenzung zu anderen Nicht-NCLC-Plattformen in Unternehmen verändern NCLC-Plattformen nicht die Arbeitsweise an sich, sondern ermöglichen es Mitarbeitern, selbst ihre Arbeitsweise zu beeinflussen. Diese Auswirkungen werden spürbar in einer erheblich erhöhten Anzahl an Entwicklern, welche selbstständig und in kurzen Zeitabständen eigenständige Lösungen mithilfe der Plattform entwickeln können.<sup>183</sup> Diese Lösungen sind aufgrund der Tatsache, dass sie von den anfordernden Mitarbeitern selbst gestaltet werden, wesentlich besser auf die individuellen Anforderungen angepasst und es kommt nicht zu Zeitverlusten in der Übertragung der Anforderung von der Fachabteilung in IT-Anforderungen. Dies hat Auswirkungen für die Fachabteilungsmitarbeiter, da dort neue Aufgaben entstehen, zusätzlich zu den originären Aufgaben der Fachabteilung. Die Experten stellen heraus, dass nicht jeder Mitarbeiter einer Fachabteilung als Citizen Developer tätig sein muss. Allerdings unterliegen die Mitarbeiter, die als solche tätig sind, einer Veränderung ihres Aufgabenumfanges. Dieser Einflussfaktor wird ebenfalls in der IT-Abteilung sichtbar. Im Falle, dass die IT-Abteilung die Plattform selbst einsetzt, ist es möglich, dass diese schneller angeforderte Anwendungen bereitstellen kann und der Aufgaben-Backlog der IT-Abteilung dadurch entlastet wird. Dies wird ermöglicht durch die Einfachheit der Anwendungsentwicklung, weil keine grundlegende Programmierung nötig ist bzw. letztere in einem überschaubaren Arbeitsumfang durchgeführt wird. Das Aufgabenfeld einer IT-Abteilung liegt jedoch nicht ausschließlich in der Erstellung neuer Anwendungen für das Unternehmen: Wesentliche Faktoren sind der Betrieb der bestehenden IT-Infrastruktur und die Administrierung von anderer Software im Unternehmen. Für den Einsatz von NCLC-Plattformen bietet es sich an, dedizierte Entwickler innerhalb der IT-Abteilungen zu haben, welche sich auf die Plattformen spezialisieren.

---

<sup>183</sup> Subcodes: 1.6.2,3,5

Dies zieht eine Veränderung von Tätigkeiten der Entwickler nach sich. Wie bereits aufgezeigt, gibt es Veränderungen in beiden Fällen in der Arbeitsweise der Mitarbeiter, die die originären Aufgaben beeinflussen. Ein weiteres Szenario, was durch die Experten aufgezeigt wurde, ist eine neue Art von kollaborativem Arbeiten zwischen IT-Abteilung und Fachabteilung. Der Umstand, dass Fachabteilungsmitarbeiter häufig keine vertiefenden IT-Kenntnisse besitzen, kann sich negativ auf die Qualität der Anwendungen selbst auswirken. Diese Erfahrung der Experten deckt sich mit den in Kapitel 2.2.1 genannten Nachteilen.<sup>184</sup> Zur Vermeidung dieser Problemstellung wird durch die Experten eine veränderte kollaborative Zusammenarbeit aufgezeigt. Die grundlegende Entwicklung einer komplexen Anwendung mit entsprechender Anbindung an verschiedene Unternehmenssysteme erfolgt durch die IT-Abteilung, um die Qualität der zugrundeliegenden Struktur zu sichern. Die weitere Anpassung an die Bedürfnisse der Fachabteilung erfolgt durch die Citizen Developer selbst, um die Qualität der Anforderungen zu sichern. Diese veränderte Arbeitsweise führt zu einer teilweisen Entlastung der IT-Abteilung als Bottleneck und fördert die Zufriedenheit mit der Anwendung, da das User Interface durch die Fachabteilung selbst gestaltet wurde.

## 8.1.2 Prozessveränderungen

Die Veränderungen, die in Unternehmen entstehen, beziehen sich ebenfalls auf eine Neuordnung von Prozessen. Prozesse, die vor der Einführung durch Mitarbeiter durchgeführt wurden, können jetzt in digitaler Form umgesetzt werden und entweder teilweise oder vollautomatisiert realisiert werden.<sup>185</sup> Dies verursacht die Notwendigkeit für Unternehmen, die eigenen Prozesse neu zu evaluieren und anzupassen. Beispielsweise werden Szenarien überflüssig, in denen ursprünglich Daten aus einer Unternehmensanwendung bezogen wurden, um diese in einer Exceltabelle weiterzuverarbeiten und dann per E-Mail weiterzuschicken. Eine Realisierung über NCLC-Anwendung in diesem Zusammenhang verringert nicht nur den zeitlichen Aufwand, den Mitarbeiter dafür aufbringen müssen, sondern verringert die Fehleranfälligkeit des Prozesses bei der Übertragung der Daten zwischen den verschiedenen Arbeitsstationen. Ein weiterer Aspekt, der durch die Automatisierung der Prozesse und die Einbindung in NCLC-Anwendung erfolgt, ist die Verbesserung des Prozesses zum Sammeln von Daten.<sup>186</sup> Diese werden schneller, in höherer Qualität und größerem Volumen durch die Plattform bereitgestellt, da die Möglichkeiten dieser wesentlich weitreichender sind. Dadurch werden in Organisationen völlig neue Möglichkeiten der Datenanalyse kreiert.

Es werden des Weiteren Prozesse beeinflusst, die sich mit der unternehmensinternen IT-Security und dem Datenmanagement befassen.<sup>187</sup> Weil die Plattform den Rahmen vorgibt, in dem der Entwickler Anwendungen erstellen kann, ist es möglich, eine Verbesserung der Datensicherheit und IT-Security zu erreichen. Dies wird durch das Vorgeben bestimmter Richtlinien und durch technische Beschränkungen, wie die Anwendungen mit Daten arbeiten können, erreicht. Eine Überwachung durch die IT-Abteilung ist jedoch weiterhin nötig, sie kann aber vereinfacht werden.

Wie bereits beschrieben, entsteht eine Veränderung in der Schnittstelle IT-Abteilung und Fachabteilung. Dies wirkt sich nicht nur auf die Arbeitsabläufe dieser Abteilungen aus,

---

<sup>184</sup> Subcode 1.7.1 sch echte Qua tät der Anwendungen;

<sup>185</sup> Subcode 1.6.2 Prozessverbesserung; Kap te 2.2.1

<sup>186</sup> Kap te 2.2.1 Vorte e Punkt 5

<sup>187</sup> Subcode 3.2 Cybersecur ty und Datenschutz; Kap te 2.2.1 Vorte e Punkt 8; Kap te 2.2.1 Nachte e Punkt 2, 3

sondern ebenfalls auf den generellen Prozess für Anwendungsanfragen an die IT-Abteilung. Dadurch, dass Citizen Developer selbst die Anwendungen teilweise oder ganz entwickeln, müssen neue Mechanismen etabliert werden, zum Beispiel wie eine Anwendungsanfrage gestellt wird und wie im Weiteren diese durch die IT-Abteilung überwacht werden kann. Ansonsten besteht die Gefahr einer Schatten-IT, die sich durch das unkontrollierte Entwickeln von Anwendungen durch Fachabteilungen ergibt.<sup>188</sup> Dies wirkt sich ebenso auf die Supportstruktur durch die IT-Abteilung aus, da schnell die Situation entstehen kann, dass die IT-Abteilung hauptsächlich damit beschäftigt ist, herauszufinden, welche Anwendung gerade wo und wie eingesetzt wird und auf welche Daten zugegriffen wird. Dies würde die IT-Abteilung erneut zum Bottleneck machen und die Gefahr von IT-Security-Problemen aufwerfen, weil die originären Aufgaben der IT-Abteilung eventuell vernachlässigt werden. Hier wird die Notwendigkeit deutlich, neue Prozesse zu etablieren, welche den Application-Life-Cycle der Anwendungen organisieren, überwachen und Vorgehensweisen festlegen. Ein gewisser Einfluss wird ebenfalls bei Freigabeprozessen gesehen. Durch die Anwesenheit einer neuen Entwicklergruppe müssen die im Unternehmen vorhandenen Freigabeprozesse überprüft und eventuell neu evaluiert werden. Diese legen fest, nach welchen Kriterien Citizen Developer ausgewählt werden und welche Freigaben diese für Unternehmensanwendungen sowie auf der Plattform selbst erhalten.<sup>189</sup>

### 8.1.3 Kultur des Unternehmens

Ein wichtiger Einflussfaktor, der sich in der Analyse der Experteninterviews herausgestellt hat, ist eine Veränderung in der Unternehmenskultur.<sup>190</sup> Aufgrund der Tatsache, dass es sich bei der Einführung einer NCLC-Plattform um eine Anwendung handelt, die viele Bereiche des Unternehmens betrifft und verändert, ist eine Veränderung in der Unternehmenskultur eine Auswirkung, welche Unternehmen berücksichtigen müssen. Ein Unternehmen, das eine sehr traditionelle Unternehmenskultur hat, welche auf der Grundlage basiert, dass Mitarbeiter Aufgaben zugewiesen bekommen und nur diese bearbeiten sollen, wird erhebliche Schwierigkeiten mit der Einführung des Citizen-Development-Konzepts haben. Teil dieses Konzepts ist es, Mitarbeiter dazu zu befähigen, selbstständig die eigenen Prozesse zu analysieren und zu verbessern mithilfe der Plattform.<sup>191</sup> Dieses Konzept kann aber nur voll realisiert werden, wenn in diesem Punkt ein Umdenken entsteht. Zu diesem Zweck sollten die Mitarbeiter ermutigt und ihnen sollte die Freiheit gelassen werden, neue Vorgehensweisen auszuprobieren. In diesem Kontext ist eine gute Fehler- und Feedback-Kultur wichtig. Es wird von den Experten als schädlich angesehen, wenn Mitarbeiter zu enge Schranken erhalten oder diese herabwürdigend behandelt werden. Ein skizziertes Szenario hierbei ist, dass eine Entwicklungsidee nicht den Ansprüchen genügt, die das Unternehmen hat oder die Idee von bestehenden Vorgehensweisen abweicht. Die Entstehung eines Arbeitsklimas, in dem Ideen berücksichtigt werden, die als unorthodox betrachtet werden, kann ein wichtiger Treiber für Innovationen sein. Beim Vorhandensein eines eher negativen Unternehmensklimas, welches zu starke Schranken aufbaut und bei der Feedback auf eine destruktive Weise an die Citizen Developer weitergegeben wird, wird eine hohe Wahrscheinlichkeit gesehen, dass diese sich nicht weiter aktiv an der Realisierung beteiligen,

---

<sup>188</sup> Subcodes: 1.9.1 Negativer Auswirkungen IT; 3.1 Schatten IT

<sup>189</sup> Subcode: 1.1 Auswahl Citizen Developer; Subcode: 1.8. Governance Einfluss

<sup>190</sup> Subcode: 1.5 Umgang Mitarbeiter der Unternehmenskultur

<sup>191</sup> Kapitel 2.3

was zum Scheitern des Projekts führen kann, weil die IT-Abteilung durch die Citizen Developer entlastet werden und nicht das Aufgabenfeld dieser übernehmen soll.

## 8.2 Vorphase des Einführungsprojekts

Die Experten haben in ihren Interviews besonders herausgestellt, dass die Vorbereitung des Projekts ein essenziell wichtiger Faktor ist. Zu diesem Zweck sollte eine intensive Vorbereitung durch das Unternehmen durchgeführt werden. Es sollte dafür berücksichtigt werden, dass diese Vorbereitungen bereits erfolgt sind, bevor die Umsetzung des Projektes beginnt, da es sonst während des Projektes zu einer erheblichen Veränderung des Projektumfangs kommen kann. Dies beeinflusst die in Relation dazu stehenden Projektfaktoren Projektkosten und Projektzeitplan.

### 8.2.1 Organisatorische Vorbereitung

Zum Zweck der Vorbereitung sollte durch das Unternehmen im Vorfeld evaluiert werden, welche Bedingungen im Unternehmen vorhanden sind und wie diese auf der Plattform umgesetzt werden sollen. Ein wichtiges Verständnis, das in einem Unternehmen vorhanden sein muss, ist das Projektbewusstsein.<sup>192</sup> Durch die Interviews wurde herausgestellt, dass viele Experten in der Praxis erfahren, dass der Umfang des Projektes unterschätzt wird. Die Einführung einer NCLC-Plattform ist eine große Veränderung für ein Unternehmen, wird jedoch oftmals mit der Einführung eines neuen Softwaretools gleichgesetzt, welches einen wesentlich geringeren Einfluss auf die Organisation als Gesamtheit hat. Die Experten weisen allerdings darauf hin, dass dies eine Fehleinschätzung ist, da bei der Einführung eines neuen Softwaretools der involvierte Stakeholder-Kreis sowie Prozessveränderungen ein überschaubares Maß erreichen. Dies ist bei NCLC-Einführungsprojekten nicht der Fall. Vor allem zu Beginn müssen wichtige personelle und finanzielle Ressourcen zur Verfügung stehen, um das Projekt starten zu können. Dies beinhaltet die Stakeholder, die am Projekt teilnehmen sowie das Festlegen von Mitarbeitern, die sich um die Administration der Plattform und das Changemanagement für die Organisation kümmern. Die Planung der entsprechenden Ressourcen sollte im Vorfeld des Projektes erfolgen, um Verzögerungen in dessen Verlauf zu vermeiden.

Die Rahmenbedingungen der Plattform sollten durch das Unternehmen ebenfalls im Vorfeld bereits geplant werden. Dem liegt zugrunde, dass eine strategische Planung für die Implementierung der Plattform durchgeführt und als Teil einer Digitalisierungsstrategie angesiedelt wurde.<sup>193</sup> Eine Digitalisierungsstrategie zeigt einem Unternehmen, dass diesem Thema eine hohe Priorität beigemessen wird und dass auf strategischer Ebene festgelegt ist, welche Ziele mithilfe der Plattform erreicht werden sollen.

Im weiteren Verlauf der Planung sollten verschiedene Evaluierungsprozesse begonnen werden. Der Zweck dieser Evaluierungsprozesse ist es, festzustellen, welche Prozesse es im Unternehmen gibt und diese darauf zu prüfen, ob sie in der Unternehmenspraxis auf die vorgegebene Weise durchgeführt werden.<sup>194</sup> Sollte es Änderungen an den bestehenden Prozessen gegeben haben, sollten diese dokumentiert und visualisiert werden. Im Zusammenhang mit diesem Thema sollte ein Unternehmen planen, wie es bei Änderungen von

---

<sup>192</sup> Subcode: 1.3 Projektbewusstsein

<sup>193</sup> Subcode: 1.4 Vorbereitung Unternehmen

<sup>194</sup> Subcode: 1.4 Vorbereitung Unternehmen

Prozessen vorgehen will und welche Änderungen sich im Zuge der Umsetzung ergeben werden. Ein weiterer Aspekt der organisatorischen Planung, der evaluiert werden sollte, neben dem Umgang mit Prozessen, ist die Fragestellung der Skalierung. Dabei sollte eine Organisation eine erste Grobplanung durchführen, für welche Bereiche die NCLC-Plattform im Roll-out zur Verfügung gestellt wird. Zusätzlich sollte überlegt werden, wie vorgegangen wird, wenn ein erweiterter Bedarf für Skalierung während des Einführungsprojekts entsteht. Dies kann der Fall sein, wenn eine Abteilung, die vorher nicht für die Skalierung geplant war, im späteren Verlauf trotzdem in diese Skalierung einbezogen werden soll. Innerhalb der Organisation sollte ein Informationsmanagementprozess eingeführt werden, welcher dazu dient, die entsprechenden Informationen im Unternehmen zu verteilen und eine Rückkopplung aus den Organisationseinheiten ermöglichen.

Die Auswahl der NCLC-Plattform sollte ebenfalls als Teil der vorhergehenden Planung in einem Unternehmen sehr genau durchgeführt werden. Teil dieser Plattformplanung ist es, ob eine Single- oder Multiplattformstrategie angewendet werden soll. Welche relevanten Problemstellungen gibt es im Unternehmen und wie lösen einzelne Plattformen diese Probleme? Auch andere im Unternehmen vorhandene Systeme und deren Anbindung an die neue NCLC-Plattform müssen bei der Auswahl berücksichtigt werden. Durch das Unternehmen sollte im Vorfeld evaluiert werden, wie eine Exit-Strategie aussehen könnte, für den Fall, dass die gewählte Plattform doch nicht das zu erfüllende Ziel erreicht. Ein Unterschätzen dieser strukturellen Aspekte kann im weiteren Verlauf des Einführungsprojektes zu erheblichen Problemstellungen führen. Ist das Thema zu niedrig priorisiert, kann es sein, dass wichtige Ressourcen nicht zur Verfügung stehen, weil dies für eine so niedrige Priorisierung nicht vorgesehen ist. Dies kann zu Beginn der Umsetzung zu Verzögerungen führen, da die Ressourcen umdisponiert werden und zusätzlich zu ihren originären Unternehmensaufgaben, dem Projekt zur Verfügung stehen müssen.<sup>195</sup> Dies führt zu einer Mehrfachbelastung der entsprechenden Ressourcen.

## 8.2.2 Governance-, Daten- und IT-Sicherheitsrichtlinien

Die Themen Governance-, Daten- und IT-Sicherheitsrichtlinien wurden in den Experteninterviews sowie in der Literaturrecherche herausgehoben.<sup>196</sup> Viele der Experten sehen hier einen erhöhten Handlungsbedarf durch Unternehmen. Es sollte durch das Unternehmen definiert werden, wie diese Rahmenbedingungen aussehen sollen. Diese Richtlinien können bei Bedarf während des Einführungsprojektes erneut angepasst werden. Eine grundlegende Planung sollte allerdings im Vorfeld festgelegt werden. Die Planung im Vorfeld ist wichtig, da ein Unternehmen selbst wissen muss, wie es mit diesen Themen umgehen kann, will und darf. Das Thema Governance bezieht sich in diesem Zusammenhang darauf, wie mit den Anwendungen umgegangen wird, wenn sie erstellt werden. Die Regeln, die hierbei festgelegt werden, sollen der Vermeidung einer Schatten-IT dienen. Unternehmen sollten festlegen, welche Rechte nach welchen Kriterien für Nutzer vergeben werden. Im Weiteren sollte festgelegt werden, wie ein Application-Life-Cycle aussehen soll und wie mit veralteten Anwendungen umgegangen wird. Wurden bereits eine oder mehrere Plattformen ausgewählt, sollten Nutzungsrechte dahingehend festgelegt werden, welche Bestandteile der Plattform genutzt werden dürfen. Diese Art der Plattformen bieten oftmals weitergehende Tool-Sets, wobei nicht jedes Tool für jeden Nutzer sinnvoll ist. Zur Entwicklung von

---

<sup>195</sup> Subcode: 3.3 Thema unterschätzt

<sup>196</sup> Subcode: 1.8 Governance Einführung; Subcode: 3.2 Cybersecurity und Datenschutz; Kapitel 2.2.1 Vorteile Punkt 8, 10; Kapitel 2.2.1 Nachteile Punkt 2, 3

Anwendungen sollten Richtlinien festgelegt werden, wie eine kohärente Struktur aussehen sollte. Dadurch wird vermieden, dass viele verschiedene Anwendungsdesigns im Unternehmen vorhanden sind, in die sich jedes Mal individuell hineingedacht werden muss, um diese nutzen zu können. Dies kann ein abschreckender Faktor für Nutzer sein.

Diese Richtlinien sollten ebenso auf die Plattform erweitert werden. Dabei besteht ein Klärungsbedarf, auf welche Daten eine solche Plattform zugreifen darf und mit welcher anderen Software letztere über Connectoren verbunden werden darf.

Administrationsrichtlinien sollten ebenso in diesem Zusammenhang bereits geklärt werden. Wer hat die Berechtigungen, Nutzer freizugeben und wie sieht ein Prozess aus, der für diese Berechtigungen erfüllt sein muss? Eine Befähigungsmaßnahme für diese Plattform ist das Bilden von Communities. Teil der Governance-Strategie sollte es sein, dass Unternehmen hier schon festlegen, wer diese überwachen soll und wie die zugrundeliegende Infrastruktur aussehen könnte.

Teil der Governance sind ebenfalls Dokumentationsrichtlinien. Wie werden entwickelte Anwendungen und die damit verbundenen Prozesse dokumentiert? Steht ein Entwickler aus irgendwelchen Gründen dem Unternehmen nicht mehr zur Verfügung, sollte nicht die Situation eintreten, dass die entwickelte Applikation nicht mehr verwendet werden kann, weil niemand versteht, wie die Applikation funktioniert. Diese Notwendigkeit erweitert sich auf den Bereich der Weiterentwicklung. Entwickelte Anwendungen sollten nachvollziehbar sein, damit keine erhöhten Zeitaufwände für das Hineinarbeiten in die Plattform nötig werden.

Eine Festlegung im Bereich der Zugriffsberechtigung beugt dem unberechtigten Zugriff vor und schützt die unternehmensinterne IT vor ungewollten Datenzugriffen. Hierbei gilt die Empfehlung, die zukünftigen Citizen Developer einzubinden, um ein sinnvolles Rahmenwerk aufzubauen, das nicht zu restriktiv ist. Ein zu restriktives Rahmenwerk könnte die Nutzung für die Citizen Developer unpraktikabel machen, was zur Ablehnung führen kann.

Teil eines funktionierenden Datenschutzes ist es, für die Unternehmens-IT-Sicherheit erste Festlegungen zu treffen. Zugriffsrechte für Daten wurden bereits angeführt, es sollten jedoch ebenfalls Regelungen getroffen werden, wie diese weiterverarbeitet und transferiert werden dürfen. Im schlimmsten Fall kann es dazu kommen, dass wichtige unternehmensinterne Daten ungewollt das Unternehmen verlassen. Der Umgang mit Daten ist aus Sicherheits- und regulatorischen Gründen ein sehr wichtiges Thema. Plattformhersteller erlangen Zugriff auf die Daten des Unternehmens und es sollte in diesem Zusammenhang abgewogen werden, welcher Grad an Freigabe von Daten sich mit dem Nutzen durch die Plattform amortisiert. Aufgrund der essenziellen Bedeutung dieser Themen sollten bereits im Vorfeld dedizierte Positionen im Unternehmen vorhanden sein, die sich mit den Themen Governance, Datenschutz und IT-Sicherheit beschäftigen. Diese dienen als Ansprechpartner während des Einführungsprojektes und sollten entsprechend Ressourcen verwalten können und im Einführungsprojekt eingebunden werden. Zusätzlich sollte festgestellt werden, welche Stakeholder durch Prozesse des Projekts betroffen sind, um sie zu einem möglichst frühen Zeitpunkt in das Informationsmanagement einzubinden.

### **8.2.3 Changemanagement**

Ein Teilbereich, der in dieser Vorplanungsphase einfließt, ist die Etablierung eines Changemanagement-Systems. Hierbei ist nicht die Veränderung gemeint, die durch Änderungswünsche im Projekt entsteht, sondern die Veränderung der gesamten Organisation.



Als ein stark begünstigender Faktor für Unternehmen wurde ein dediziertes Mindset in den Experteninterviews herausgestellt.<sup>197</sup> Durch die weitreichenden Fähigkeiten, die eine solche Plattform bereitstellt, und die damit verbundenen neuen Möglichkeiten für die Arbeitsweise innerhalb einer Organisation, kann es nötig werden, ein neues Bewusstsein im Unternehmen zu etablieren.<sup>198</sup> Diese Problematik kann je nach Unternehmen und Unternehmenskultur unterschiedlich kritisch sein. Ein Unternehmen, das lange in starren hierarchischen Strukturen gearbeitet hat, wird in diesem Prozess mehr verändern müssen als ein Unternehmen, welches eine agile Struktur mit weniger hierarchischen Ebenen besitzt. Teil dieser Veränderung sind Organisationseinheiten und Mitarbeiter gleichermaßen. Vorgesetzte müssen sich offen dafür zeigen, neue Vorgehensweisen anzunehmen. Prozesse, die funktional sind, dürfen nicht als unabänderlich betrachtet werden. Dieser Aspekt ist ebenso wichtig für Mitarbeiter, welche offen für Veränderungen sein sollten und dazu bereit sind, eigene Verantwortung zu übernehmen. Diese Veränderung darf allerdings weder den Vorgesetzten noch den Mitarbeitern aufgezwungen werden, sondern sie muss von diesen selbst gesehen werden. Zu diesem Zweck sollte mit Aufnahme der strategischen Planung ebenfalls begonnen werden, Mitarbeiter und Vorgesetzte für das Projekt zu motivieren. Ein Grund für eine fehlende Motivation kann sein, dass innerhalb der Organisation solch ein Thema wie NCLC als klassisches IT-Thema gesehen wird und die Mitarbeiter dies nicht in ihren originären Aufgabenbereich einordnen.<sup>199</sup> Teil einer Lösungsstrategie kann es sein, beispielhafte Anwendungsfälle im Sinne einer schnellen und effektiven Lösung über NCLC-Anwendungen einzubringen, die Mitarbeiter und Vorgesetzte selbst als Problem wahrnehmen. Zusätzlich muss der Mehrwert der NCLC-Plattform sichtbar sein und es muss eine intrinsische Motivation für die Mitarbeiter geschaffen werden.

In diesem Zusammenhang ist es wichtig, dass Unternehmen bereits wissen, als Teil der strategischen Planung, was erreicht werden soll mit der Plattform und was damit realistisch erreicht werden kann.<sup>200</sup> Negativ auf die Motivation von Stakeholdern kann es sich auswirken, wenn Unternehmen zu hohe Erwartungen an die entwickelten Anwendungen stellen. Diese Plattformen können keine hochkomplexen Anwendungen wie ein vollständiges ERP-System ersetzen und dies sollte nicht der Maßstab sein. Es wird ebenso wenig möglich sein, die Entwicklung durch eine IT-Abteilung vollständig zu ersetzen. Eine Empfehlung wäre es, für diesen Bereich eine dedizierte Rolle zu benennen, welche sich um diesen Aspekt des Changemanagements kümmert, um ein Scheitern der Veränderung zu vermeiden und zu dem Zweck, dass ein Ansprechpartner vorhanden ist, der für Fragen zur Verfügung steht.

#### 8.2.4 Vorgehensweisen Initialisierungsphase/Initialisierungsprojekt

Aufgrund des klaren und planbaren Projektumfangs bietet sich hier ein Vorgehen in einem der klassischen Projektmanagement-Modelle an.<sup>201</sup> Wegen der Tatsache, dass in dieser Arbeit nur große Unternehmen mit mehr als 250 Mitarbeitern berücksichtigt werden, kann die Empfehlung für kleine Unternehmen abweichen. Ein klassisches Vorgehen kann entweder als eigenständiges Vorprojekt oder als Teil eines hybriden Modells durchgeführt

<sup>197</sup> Subcode: 2.1 Mindset

<sup>198</sup> Kapite 3.4 Change Projektmanagement; Kapite 2.2 No Code/ Low Code als Lösungsansatz; Subcode: 1.4 Vorbereitung Unternehmen

<sup>199</sup> Subcode: 1.2 Motivation Stakeholder

<sup>200</sup> Subcode: 1.7.2 Falsche Erwartungen; Subcode: 3.3 Thema unterschätzt; Kapite 2.4 Einordnung von No Code/ Low Code im Softwareumfeld

<sup>201</sup> Kapite 3.1 Linear/sequenzielle Modelle

werden.<sup>202</sup> Bei einem hybriden Vorgehen sollte ein verteilter, sequenzieller Ansatz in Betracht gezogen werden, wobei dieser Bestandteil der Implementierung keinen zu großen zeitlichen Rahmen einnehmen sollte, weil hier lediglich vorbereitende Arbeitspakete behandelt werden, die sich während der Einführung ändern können.<sup>203</sup>

## 1. Phase Projektcommitment und strategische Planung

Der Output aus diesem Teil des Projektes sollte sein, dass die Geschäftsführung ein Commitment gegenüber dem Projekt empfindet und das Thema entsprechend der Priorisierung in der Hierarchie des Unternehmens eingliedert. Zu diesem Zweck sollte die Aufnahme in die strategische Planung des Unternehmens erfolgen. Dementsprechend sollte in der strategischen Planung berücksichtigt werden, dass Ressourcen für das Projekt benötigt werden und es sollte ein Commitment dafür vorhanden sein, diese bereitzustellen. Mit dieser Phase sollte ebenfalls begonnen werden, innerhalb der Organisation ein entsprechendes Mindset zu etablieren sowie, falls nötig, einen kulturellen Change zu beginnen. Diese Phase sollte vor den anderen durchgeführt werden, da sie für die nachgelagerten Phasen der Ressourcenplanung und der operativen Planung des zu Erreichenden wichtig ist. Ebenfalls wichtig ist diese Phase für den Change-Prozess der Organisation, weil ein Commitment zu der Themenstellung durch die Geschäftsleitung in ein Unternehmen hineinwirkt.<sup>204</sup>

## 2. Phase Organisatorische Vorbereitungen und Planung

Projektoutput dieser 2. Phase sollte sein, dass alle Prozesse im Unternehmen, die durch die Plattform betroffen sein könnten, bekannt sind. Ein entsprechender Changemanagement-Prozess sollte etabliert werden für die Änderungen, die an den Prozessen erfolgen. Die weiteren relevanten Stakeholder für das Umsetzungsprojekt sollten ebenfalls in dieser Phase benannt werden. Explizit sollten Ressourcen für die Rollen Changemanagement, Governance, IT-Security und Datenbeauftragter zugewiesen sein und zu Beginn des Umsetzungsprojekts verfügbar sein. Die technischen Rollen für Administration und IT-Infrastruktur sollten ebenfalls benannt und mit entsprechenden Ressourcen versehen werden. In dieser Phase sollte eine Abfrage an die relevanten Fachabteilungen erfolgen, welche Mitarbeiter sich für das Citizen-Developer-Konzept eignen, um diese, wenn benötigt, im Umsetzungsprojekt einbinden zu können. Mindestens sollten Anforderungen aufgenommen werden, welche die Fachabteilungen an die Implementierung der Plattform haben. Ein weiterer Output, der aus dieser Phase hervorgehenden sollte, ist eine erste Planung für Regeln zu den Themen Governance, IT-Security und Datenschutz. Es sollte innerhalb der Organisation dazu evaluiert werden, welche vorhandenen Systeme existieren und welche regulatorischen Anforderungen für die entsprechende Organisation gelten. Die technischen Voraussetzungen der Organisation sollten ebenfalls evaluiert werden, um eventuellen Handlungsbedarf zu ermitteln, falls die Organisation diese nicht erfüllt.<sup>205</sup> Die Etablierung eines Informationsmanagements kann ebenfalls sinnvoll sein, um möglichst frühzeitig die Mitarbeiter auf die Veränderung in der Organisation vorzubereiten. Dafür sollten die entsprechenden Strukturen geschaffen werden, wenn diese nicht bereits bestehen. Ein weiterer wichtiger Bestandteil des Informationsmanagements ist es, alle relevanten Stakeholder, die identifiziert werden konnten, in den Informationsfluss einzubinden. Dadurch sollen

---

<sup>202</sup> Kap te 3.1L near/sequenz e e Mode e; Kap te 3.3 Hybr des Projektmanagement

<sup>203</sup> Subcode: 5.2 Hybr des Meet ng

<sup>204</sup> Subcode: 2.1 M ndset

<sup>205</sup> Subcode: 2.4 Techn sche P anung

Verzögerung im Verlauf des Projektes vermieden werden. Wenn zum Beispiel Mitarbeitervertretungen erst während der Implementation von dem Projekt erfahren und dieses wegen ungeklärter Fragen stoppen, hat das einen starken Einfluss auf den Projektzeitplan.<sup>206</sup>

Diese Schritte können parallel organisiert und je nach Themenbereich aufgegliedert werden. Bei diesem Vorgehen kann entweder über die CPM-Methodik oder in einem agilen Scrum-Vorgehen gearbeitet werden. Hier sollte berücksichtigt werden, welches Vorgehen in der Organisation verbreiteter ist, um Unsicherheiten und Umsetzungsfehler innerhalb des Vorgehensmodells zu vermeiden.<sup>207</sup>

### **Phase 3 Auswahl Plattform**

In einer abschließenden Phase sollten die gewonnenen Erkenntnisse aus den vorgelagerten Phasen verwendet werden, um eine Plattform auszuwählen, die zu den Anforderungen des Unternehmens passt. Hierbei sollte die strategische Entscheidung getroffen werden, ob eine Single- oder Multiplattformstrategie angewendet werden soll. Bei der Auswahl sollte berücksichtigt werden, inwieweit Plattformen für die Zielstellung, die erreicht werden soll, verwendet werden können. Es sollte ein Kriterienkatalog erstellt werden auf Basis der gewonnenen Erkenntnisse zu Governance-, Daten- und IT-Security-Anforderungen und dem Feedback der Citizen Developer und Fachabteilungen. Anhand dieses Kataloges können die verschiedenen Plattformen auf Übereinstimmung überprüft und es kann dadurch eine Auswahl getroffen werden.

Zusätzlich sollte die Möglichkeit einer Exit-Strategie evaluiert werden. Diese kann nötig werden, falls während des Einführungsprojektes die gewählte Plattform doch nicht die Anforderungen der Organisation erfüllt. Ein zweites Szenario, was bei dieser Strategie berücksichtigt werden sollte, ist die Möglichkeit, dass in der aktiven Betriebsphase die Plattform nicht mehr die laufenden oder geänderten Anforderungen erfüllt. In diesem Fall sollte es möglich sein, ohne größere Probleme die Plattform durch eine andere zu ersetzen.

---

<sup>206</sup> Subcode: 11.5 Mitarbeitervertretung

<sup>207</sup> Kapitel 3.1.1 Wasserfallmodell ; Kapitel 3.2.1 Scrum

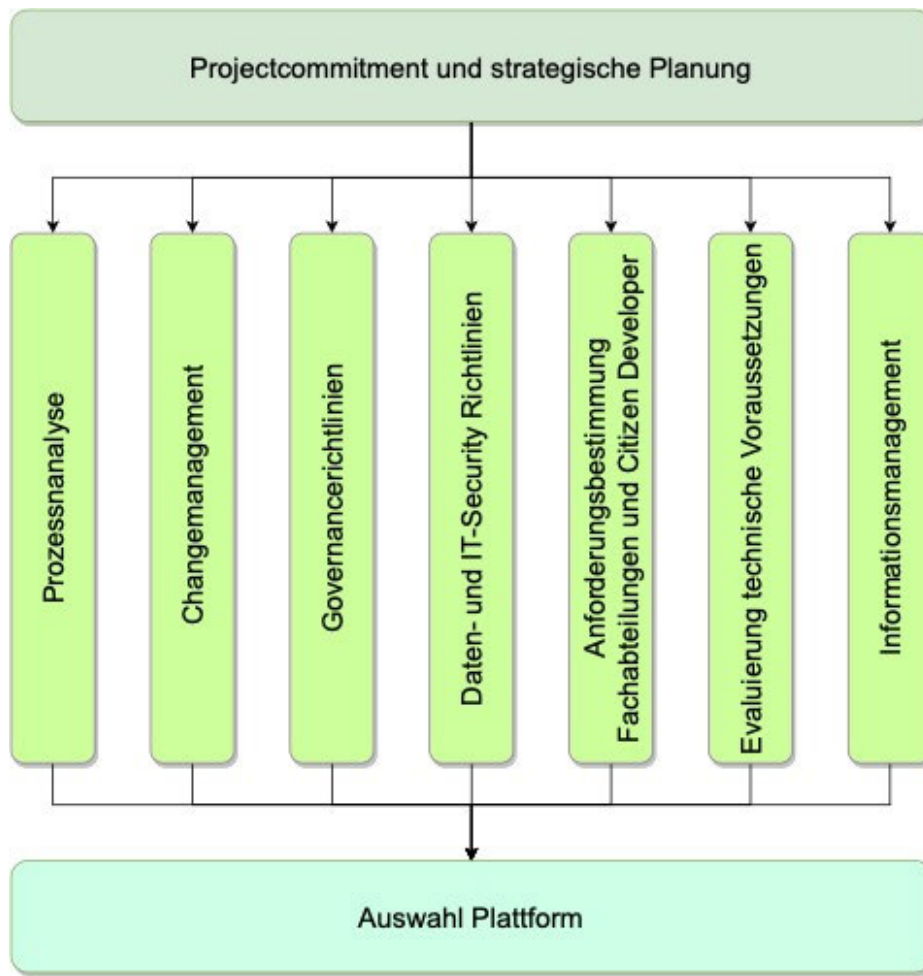


Abbildung 14 - Ablaufmodell Initialisierungsprojektphase

## 8.3 Umsetzung der Implementierung

Je nachdem, wie sich in der Initialisierungsphase des Projektes entschieden wurde, beginnt mit der Umsetzung die aktive Phase des Vorgehensmodells, entweder als neues Projekt oder neue Phase des hybriden Vorgehens. Voraussetzung für diese Phase ist es, dass die Auswahl der Plattform erfolgt ist und die entsprechenden Richtlinien und personellen Ressourcen bestimmt wurden. In dieser Phase wird mit dem Unternehmen, welches die Implementation vornimmt, eng zusammengearbeitet. Wichtig für diese Zusammenarbeit ist, dass alle Ressourcen in der geforderten Qualität und Menge zur Verfügung stehen, um einen problemlosen Übergang sicherzustellen.

### 8.3.1 Projektorganisation

Bei der Organisationsform während der Implementierung beschreiben die meisten Experten ein Vorgehen in der Form einer Matrixorganisation.<sup>208</sup> Bei dieser Vorgehensweise werden die Fachabteilungsmitarbeiter in ihren originären Abteilungen belassen und nur zum Zwecke der Abstimmung und für dedizierte Projektworkshops sowie Arbeitsphasen aus ihren Abteilungen herausgelöst. Dies kann aus verschiedenen Gründen notwendig sein.

<sup>208</sup> Hauptcode: 4 Projektorgan sat on

Neben dem Verlust von wichtigen Produktivkräften bei einer vollständigen Projektzuordnung geht der Faktor des Fachabteilungsfeedbacks verloren. Durch die Arbeit der Projektmitglieder in den Abteilungen können diese die Fortschritte des Projektes kommunizieren und vorstellen. Auf dieser Grundlage kann wichtiges Feedback aus den Abteilungen gesammelt werden, welches in das Projekt zurückgespiegelt wird. Ein weiterer Vorteil, der sich durch dieses Vorgehen generiert, ist, dass für die Plattform ein Verständnis erzeugt wird, wodurch die Akzeptanz durch die Mitarbeiter und Fachabteilungen steigen kann. Eine Rolle, die in diesem Zusammenhang gesehen wird, ist die Rolle des Projektverantwortlichen, der das Projekt in der Organisation voranbringt. Diese Rolle definiert die Zielstellung des Projektes und sollte innerhalb der Hierarchie Entscheidungskompetenz haben. Für die Umsetzung wird eine agile Vorgehensweise zugrunde gelegt und in diesem Zusammenhang würde diese Rolle derjenigen eines Product Owners entsprechen. Eine weitere Rolle ist diejenige der Citizen Developer, welche im weiteren Verlauf des Projekts als Multiplikatoren dienen und den Projektfortschritt in die Fachabteilungen tragen.

In Abgrenzung zu den Rollen, die teilweise dem Projekt zugeordnet sind, sollten einige Projektrollen dauerhaft im Projekt verbleiben. Dies betrifft Rollen, welche eine dedizierte Projektaufgabe verfolgen, wie den Changemanagement-Beauftragten, der sich ausschließlich um diese Aufgaben kümmern sollte.<sup>209</sup> Diese Rolle hat ein umfassendes Aufgabenfeld, das nicht neben einem weiteren Aufgabenfeld erledigt werden sollte. Dies begründet sich darin, dass viele Maßnahmen ergriffen werden müssen, um die Organisation auf die Veränderung vorzubereiten.

Der Projektleiter dient als Schnittstelle zwischen Projektteam und dem restlichen Unternehmen. Dieser sollte als Ansprechpartner für die Projektmitglieder zur Verfügung stehen und er sollte Entscheidungen, welche durch das Projektteam getroffen werden, in das Unternehmen tragen. Diese Aufgabe erfüllt er zusätzlich zu den Projektplanungsaufgaben. Eine Aufgabe, die zusätzlich durch den Projektleiter erfüllt werden kann, ist abhängig vom Projektsetup. Wird das Projekt in einer Scrum-Vorgehensweise durchgeführt, kann es sein, dass die Aufgabe des Scrum Masters mit durch den Projektleiter übernommen wird. Dies ist ebenfalls abhängig von der Projektgröße, da in größeren Projekten eine vollumfängliche Scrum-Master-Rolle nötig ist. Ist diese Bedingung erfüllt, ist der Scrum Master eine eigene dedizierte Rolle, die vollständig dem Projekt zugeordnet sein sollte. Weitere wichtige Rollen, die im Projekt angesiedelt sind, sind IT-Security, Governance und Datenschutzrollen. Diese technischen Rollen sind für den Aufbau der jeweiligen Richtlinien auf der Plattform und die entsprechende praktische Umsetzung im Unternehmen verantwortlich. Ein weiterer Teil dieser Aufgabe ist ebenfalls der Aufbau einer Supportstruktur für die späteren Citizen Developer. Die Funktion des Enablers wird ebenso als Aufgabenfeld für das Projekt wichtig. Diese Tätigkeit dient dazu, auf der einen Seite die Citizen Developer zu Beginn zu befähigen und auf der anderen Seite die Organisation für den späteren aktiven Betrieb der Plattform vorzubereiten und zu befähigen.

---

<sup>209</sup> Hauptcode: 10 Rollen und Stakeholder im Projekt

### 8.3.2 Vorgehensmodell agile Phase

Eine agile Vorgehensweise bei der Implementierung der NCLC-Plattformen wird durch Experten und Literatur als die sinnvollste angesehen.<sup>210</sup> Aufgrund des agilen Charakters der Plattformen und des sich kontinuierlich verändernden Funktionsumfangs ist eine agile Vorgehensweise wegen der charakteristischen hohen Reaktionsgeschwindigkeit dem traditionellen Vorgehen vorzuziehen. Bei dieser Projektphase ist es im Gegensatz zur Vorphase wesentlich schwieriger, den genauen Projektumfang abzuschätzen. Zwar erleichtert die Planung in der Vorphase in vielen Punkten die Planung der Umsetzung und grenzt den Projektscope ab, allerdings wirkt der agile Charakter der Plattform dem entgegen. Ein Vorgehen nach einem traditionellen Vorgehensmodell ist nur sinnvoll, solange es keine Erfahrungen mit agilen Methodiken gibt und die Organisation sich einem agilen Vorgehen im Gesamten verweigert. Von diesem Vorgehen wird allerdings nachhaltig abgeraten und wenn ein klassisches Vorgehen gewünscht wird, sollte eher ein hybrider Ansatz verfolgt werden, mit einem Schwerpunkt auf dem traditionellen Vorgehen. Dadurch können einige Vorteile der agilen Methodiken in das Projekt übernommen werden, wie das schnelle Feedback.

Gibt es innerhalb der Organisation jedoch eine Offenheit für ein neues Vorgehen oder sind agile Vorgehensweisen bereits bekannt, sollte auf ein agiles Vorgehen zurückgegriffen werden. Dies kann entweder in einer reinen Form realisiert werden oder in einer hybriden Form mit dem Schwerpunkt auf agilen Methodiken. Sollten Unsicherheiten in Bezug auf ein agiles Vorgehen in der Organisation existieren, sollte über eine Befähigungsphase vor der Umsetzung nachgedacht werden und eine Zusammenarbeit mit einem Partner, welcher diese Vorgehensweise bereits kennt.<sup>211</sup>

Vor allem die Möglichkeit, innerhalb des agilen Vorgehens in Scrum-Teams zu arbeiten, wird durch nahezu alle Experten als positiver Faktor hervorgehoben.<sup>212</sup> Dieses Vorgehen bietet sich aufgrund der Eigenschaften von Scrum als vorteilhaft an, weil jegliche Entwicklung sichtbar gemacht wird und im Weiteren während des iterativen Vorgehens über Feedback schnell Schlussfolgerungen gezogen werden können. Diese Schlussfolgerungen helfen dem Vorgehen, sich an neue Anforderungen anzupassen. Dies ermöglicht dem Implementierungsteam, sich immer wieder schnell anpassen zu können und damit der Problematik vorzubeugen, dass die Plattform durch den Plattform-Betreiber weiterentwickelt wird, allerdings um diese Weiterentwicklung in das Projekt einzubinden, ein langes Changerequest-Verfahren durchgeführt werden muss. Anforderungen können ebenfalls aus Organisationsgründen verschiedenen Änderungen unterliegen, die nicht planbar sind. Ein Experte führt als Beispiel die Schwangerschaft eines wichtigen Projektmitglieds an. Auf solche Veränderungen schnell und flexibel reagieren zu können, kann sich für das Projekt in Kennzahlen und Qualität positiv auswirken. Trotzdem kann es in diesem Fall nötig sein, bei solchen elementaren Änderungen eine Neuplanung für bestimmte Projektaspekte, wie Projektzeitplan oder Projektkosten, vorzunehmen.

Wie bereits erwähnt wurde, sind zu Beginn des Projekts nicht alle Fähigkeiten der Plattform bekannt, die diese am Ende des Projekts besitzt. Dies begründet sich dadurch, dass die Plattform auf der einen Seite durch den Plattformbetreiber weiterentwickelt wird und auf der anderen Seite es nicht möglich ist für ein Unternehmen, jedes Detail über den Funktionsumfang einer Plattform zu kennen. Dies kann es nötig machen, in der Vorphase getroffene

---

<sup>210</sup> Hauptcode: 5 Wah des Vorgehensmode ; Kap te 3.2 Ag es Projektmanagement; Kap te 3.3 Hybr des Projektmanagement

<sup>211</sup> Subcode: 5.1.2 Begründung ag es Vorgehen

<sup>212</sup> Subcode: 5.1 Ag es Vorgehensmode ; Kap te 3.2.1 Scrum

Planungen bzgl. Governance-, Prozess- oder Sicherheitsrichtlinien an die geänderten Gegebenheiten agil anzupassen. NCLC-Projekte können ebenfalls aufgrund der weitreichenden Möglichkeit der Plattformen einen größeren Projektumfang erreichen, weshalb die Kosten und die zeitliche Planung gleichfalls steigen können. In diesem Zusammenhang ist ein Vorgehen in klassischen Vorgehensmodellen mit mehr negativen Aspekten verbunden, als das bei agilen Methoden der Fall wäre. Es gibt zwar die Möglichkeit der Parallelisierung von Aufgabenpaketen mittels der Critical-Path-Methodik und der Verwendung eines Netzplans, diese unterliegen allerdings trotzdem den Schwächen des klassischen Vorgehens. Viele Arbeitspakete sind auf Vorgänger-Arbeitspakete oder vorherige Phasen geplant und von diesen abhängig. Änderungen müssen über ein Changerequest-Verfahren sowie die Auswirkungen auf die möglichen Nachfolger eingeplant werden. Je nach Projektfortschritt können sich daraus weitreichende Änderung an der ursprünglichen Planung ergeben. Bei einem agilen Vorgehen ist die Möglichkeit der Parallelisierung wesentlich besser gegeben und durch den flexiblen Projektscope werden Projektkosten und der Projektterminplan zu den limitierenden Faktoren.<sup>213</sup> Dies ermöglicht den Aufbau von parallel arbeitenden Scrum-Teams, welche ihre Aufgabe entsprechend themenbezogen vorantreiben, jedoch freier sind in der Gestaltung der Bearbeitung der Arbeitspakete durch das flexible Eingehen auf Änderungen der Anforderungen.

Eines dieser Scrum-Teams ist das Changemanagement-Team. Dieses Team beschäftigt sich dediziert mit dem Change in der Organisation. Ein weiteres Team dient der technischen Implementierung der NCLC-Plattform. Eine weitere Aufgabe ist der Supportaufbau und das Enabling der Organisation. Teil des Enablings ist die Befähigung auf die Plattform und die Vorbereitung der Organisation auf den aktiven Betrieb aus einer strukturellen Perspektive. Dedizierte Governance-, IT-Sicherheit- und Datenteams können ebenfalls sinnvoll in die Scrum-Struktur eingebunden werden und unabhängig von den anderen Teams die Anpassung der Plattform für die jeweiligen Themenkomplexe bearbeiten. Ein weiteres Scrum-Team sollte sich mit dem Thema Citizen Development beschäftigen und wie dieses Themengebiet innerhalb der Organisation eingebunden werden sollte. Dabei sollte entschieden werden, ob die Citizen Developer als Multiplikatoren eingesetzt werden und welche unternehmensrelevanten Use Cases bereits existieren. Diese Use Cases zu lösen, kann im späteren Verlauf des Projektes bei der Skalierung helfen.<sup>214</sup> Ein dediziertes Scrum-Team dient dem Informationsmanagement und sollte sich darum kümmern, dass alle relevanten Stakeholder Zugang zu den für sie relevanten Informationen erhalten, falls diese nicht Teil des Projektteams sind. Die Organisation der verschiedenen Scrum-Teams kann auf verschiedene Arten erfolgen, bei sehr großen Projekten entweder über das SAFe Framework oder ein Scrum-of-Scrums.<sup>215</sup> Kleinere Projekte können mittels eines Projektleiters organisiert werden, der optional die Einhaltung der Methodik überwacht. Dieses Vorgehen wird dadurch möglich, dass Änderungen, die als Ergebnis aus dem Sprint des einen Scrum-Teams erfolgen, flexibel in die zukünftige Sprintplanung von anderen Scrum-Teams aufgenommen werden können.

Die agile Vorgehensweise mit den positiven Eigenschaften, dass schnell Mehrwerte mittels eines MVP gezeigt werden können und durch Feedback die Möglichkeit gegeben ist, schnell Änderungen durchzuführen, kann jedoch ebenfalls negative Folgen haben. Durch einige Experten wird darauf hingewiesen, dass es passieren kann, dass sich das Projektteam in zu vielen Detailarbeiten verliert und die Gesamtübersicht über das Projekt verloren

---

<sup>213</sup> Abbildung 5 Vergleich Magisches Dreieck traditioneller & agiler Ansatz

<sup>214</sup> Hauptcode 10 Rollen und Stakeholder im Projekt; Subcode: 5.1 Agiles Vorgehen

<sup>215</sup> vgl. Komus and Kuberg 2020, S. 25.

geht.<sup>216</sup> Dies steht in Verbindung mit den bekannten Nachteilen der agilen Methodik, dass diese wesentlich schlechtere Planungsbestandteile bieten, in Kontrast zu den klassischen Vorgehensweisen. Ein weiteres Problem ist die Möglichkeit, dass einzelne Projektmitglieder die Priorisierung des Aufgaben-Backlogs aufgrund individueller Bedürfnisse beeinflussen können. Zur Lösung dieser Problemstellungen empfehlen einige Experten statt eines reinen agilen Ansatzes ein hybrides Vorgehen. Dadurch soll der Methodik ein verbessertes Monitoring sowie Planung und Struktur gegeben werden, ohne dabei die Vorteile der agilen Vorgehensweise komplett zu verlieren. Dies bringt allerdings den Nachteil mit sich, dass die Agilität verringert wird. Dies passiert wegen der Tatsache, dass Planung voraussetzt, dass Anforderungen und Bearbeitung nicht ausschließlich von Sprint zu Sprint durchgeführt werden, sondern bereits an einem früheren Zeitpunkt erfolgen, weil für ein kontrolliertes Vorgehen ein übergeordneter Plan erstellt wird. Dies schränkt die Fähigkeit der Scrum-Teams ein, schnell neue Anforderungen aufzunehmen, da diese Anforderungen ebenfalls mit der übergeordneten Planung abgeglichen und in diese eingearbeitet werden müssen. Weitere Stärken aus den klassischen Vorgehensmodellen sind die Dokumentation und das Monitoring des Projektfortschritts. Erfahrungen der Experten zeigen, dass diese Arbeitspakete oftmals durch das kleinteilige Arbeiten in der agilen Vorgehensweise vernachlässigt werden. Dies führt zu Problemen in der Nachvollziehbarkeit über den Bearbeitungsstand. Diese Problematik würde durch ein hybrides Vorgehen verringert werden.

### 8.3.3 Skalierung und Befähigung

Der Punkt Skalierung des Projekts in das größere Unternehmen bildet zusammen mit der Befähigung der Nutzer den Abschluss des Projekts. Ist die Einführung der Plattform so weit fortgeschritten, dass technische Struktur, Richtlinien, Prozesse und Anbindung der Plattform in die Unternehmensstruktur erfolgt sind, sollte damit begonnen werden, die Skalierung der Plattform aus der Pilotabteilung heraus zu beginnen. Es bietet sich aus Gründen der Effizienz ein sequenziell-paralleles Vorgehen an.<sup>217</sup> Dieses ist dargestellt in Abbildung 15. Ein Roll-out in allen Abteilungen erweist sich als problematisch, da dadurch eine hohe Bündelung von Ressourcen für Supportstruktur, Monitoring und IT-Abteilung benötigt wird.<sup>218</sup> Vorteilhaft ist jedoch, dass der zeitliche Faktor der Skalierung wesentlich verringert wird. Im Kontrast dazu ist ein rein sequenzielles Vorgehen nicht mit der Problematik belastet, dass die Gefahr einer Überforderung entsteht. Ebenfalls ist es bei einem sequenziellen/schrittweisen Vorgehen möglich, aus den Erfahrungen vorangegangener Skalierungsphasen zu lernen und diese Erfahrungen für neue Abteilungen zu übernehmen. Eine Problemstellung, die in diesem Zusammenhang allerdings entsteht, ist, dass ein erheblicher zeitlicher Faktor für die Umsetzung notwendig wird.<sup>219</sup>

---

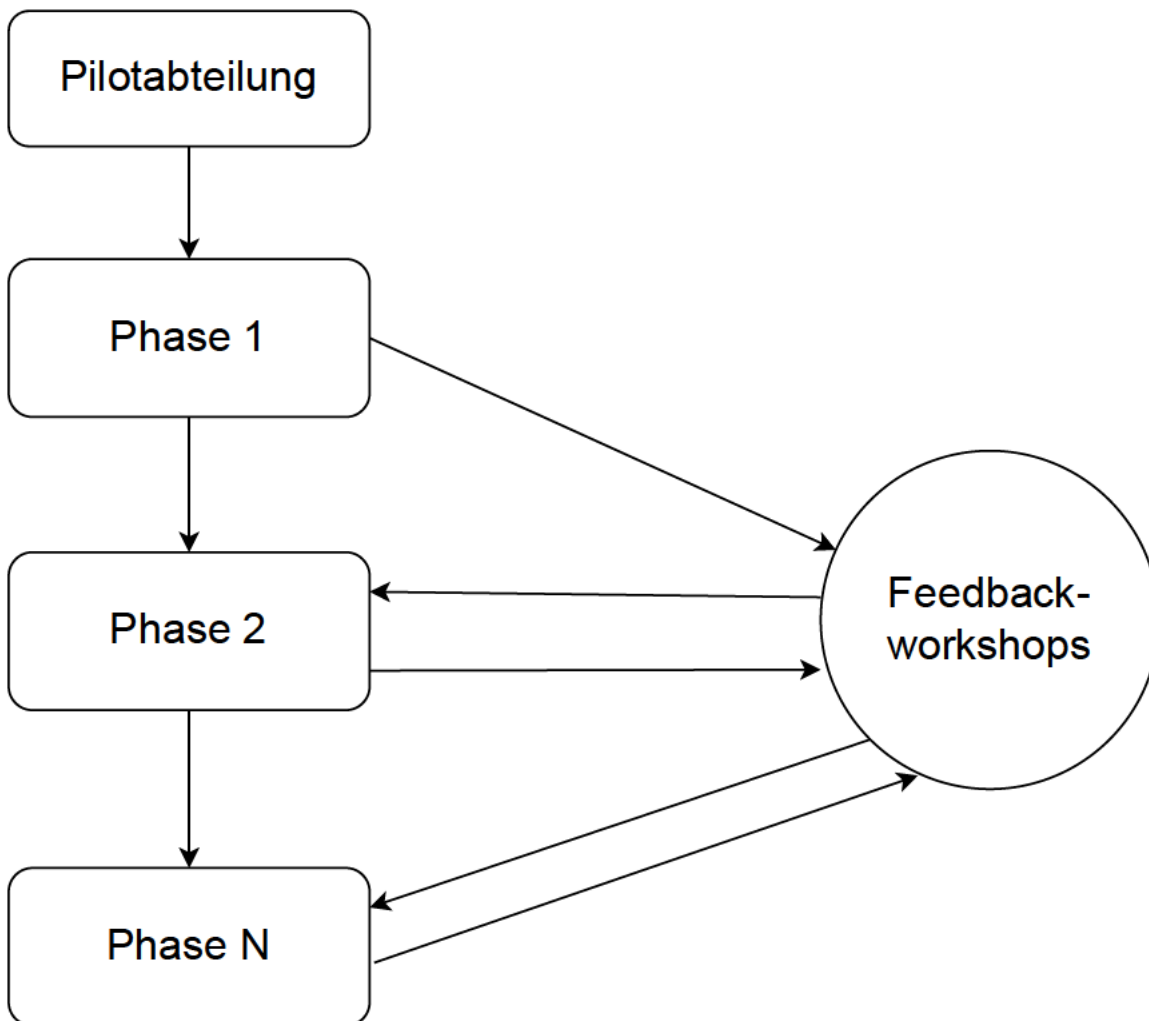
<sup>216</sup> Subcode: 7.1.1 Nachteile agiler Vorgehensmodelle

<sup>217</sup> Subcode: 12.3 Sequenzielles Vorgehen

<sup>218</sup> Subcode: 12.2 Paralleles Vorgehen

<sup>219</sup> Subcode: 12.1 Sequenzielles Vorgehen





**Abbildung 15 - Schematische Darstellung sequenziell-paralleler Skalierung**

Beim sequenziell-parallelen Vorgehen werden schrittweise Abteilungen in die Infrastruktur der Plattform eingebunden. Jede Skalierungsphase wird dadurch abgeschlossen, dass Feedback evaluiert wird, welches für die nächste Skalierungsphase einfließt. Bei diesem iterativen Vorgehen ist es möglich, eine kontinuierliche Verbesserung zu erreichen. Mit jeder Phase können mehrere neue Abteilungen in die Skalierung aufgenommen werden, um so den Roll-out im Unternehmen schneller voranzubringen. Die Anzahl der neuen Abteilungen sollte jedoch nicht zu groß sein, um eine Überlastung der IT und Supportstruktur zu vermeiden. Durch die Skalierung werden neue Entwickler schrittweise in die Unternehmensinfrastruktur aufgenommen. Es ist hier förderlich, wenn die Citizen Developer aus den vorgelagerten Skalierungsphasen als Multiplikatoren dienen können.<sup>220</sup> Diese erfahrenen Nutzer können die neuen Citizen Developer unterstützen, die Plattform zu nutzen. Als ein weiterer Teil der Befähigung sollte eine Einstiegshürde in der Plattform etabliert werden, in Form einer initialen Schulung.<sup>221</sup> Teil dieser Schulung sollte das Vermitteln der Governance-, IT-Sicherheits- und Datenschutzgrundlagen sein. Dadurch soll sichergestellt werden, dass die Richtlinien bekannt sind, damit diese eingehalten werden. Die Etablierung einer Community kann ebenfalls positive Auswirkungen für die Citizen Developer haben. Die Möglichkeit, sich untereinander auszutauschen und Hilfestellungen zu geben sowie Feedback zu sammeln, kann für die Skalierung förderlich sein, da eine schnelle Befähigung

<sup>220</sup> Subocode: 13.3 Befähigung durch Multiplikatoren

<sup>221</sup> Subocode: 13.1 Befähigung durch Schulung/ Workshops

innerhalb der Abteilungen möglich wird.<sup>222</sup> Durch das schrittweise Hinzunehmen neuer Citizen Developer sollte es zudem nicht zu einer Überlastung kommen, weil mit steigender Anzahl neuer User ebenfalls neue Multiplikatoren in der Community entstehen. Es sollte ebenfalls ein dediziertes Schulungsangebot für Citizen Developer aufgebaut werden. Dieses Schulungsangebot kann aus verschiedenen Maßnahmen bestehen. Eine Möglichkeit ist es, einen Videokurs zu erstellen. Durch diesen ist es für die Citizen Developer möglich, automatisiert sowie unabhängig von Ort und Zeit sich Wissen aneignen zu können.<sup>223</sup> Ergänzend dazu können verschiedene Workshopangebote für eine kollaborative Problemlösung von Aufgabenstellungen bereitgestellt werden, um das erlangte Wissen weiter zu vertiefen und um effektiv bessere Lösungen zu erreichen. Diese Befähigungsmaßnahmen werden parallel zur Skalierung durchgeführt und sollten kontinuierlich an geänderte Anforderungen angepasst werden.

Während der Skalierung wirken sich die Ergebnisse des Informationsmanagements und des Changemanagements aus.<sup>224</sup> Die Zielstellung ist, dass die Abteilungen über die Plattform informiert sind und anhand der vorher festgestellten unternehmensrelevanten Use Cases die Vorteile der Plattformen erkennen. Dadurch soll die Adaption durch die Mitarbeiter positiv beeinflusst werden. Neben der Befähigung ist ein wichtiger Aspekt die Vorbereitung der Abteilungen für die Veränderung der Prozesse, die sich mit der Plattform ergeben. Das Changemanagement-Team sollte vorbereitende Maßnahmen ergriffen haben, um einen möglichst effizienten Prozess für die Änderung von Abteilungsabläufen zu etablieren. Es sollten ebenfalls die Schnittstellen identifiziert werden zu anderen Abteilungen und wie diese beeinflusst werden durch die aktive Anwendung der Plattform durch die Abteilung. Ein iteratives Vorgehen ist für die Skalierung sinnvoll, um geänderte Anforderungen zu berücksichtigen, die sich aus der Entwicklung der Plattform oder im Zusammenhang mit im Vorfeld nicht berücksichtigten Abteilungsgegebenheiten ergeben.

Zur Sicherung der Qualität während des Einführungsprojekts und der nachfolgenden Skalierung werden durch die Experten verschiedene Maßnahmen genannt. Die Motivation der Mitarbeiter ist hier ebenfalls ein wichtiger Faktor. Durch ein hohes Maß an Motivation kann dafür gesorgt werden, dass die Mitarbeiter für die Plattform ein intrinsisches Interesse entwickeln, aktiv daran mitzuarbeiten und neue Features einsetzen zu wollen. Zu diesem Zweck sollte durch den Projektverantwortlichen dafür gesorgt werden, dass ein kontinuierlicher Zustrom an neuen Aufgaben besteht, der jedoch nicht den Eindruck von Überlastung im Team erzeugen darf. Existiert eine hohe Motivation, wird durch die Experten erwartet, dass die Nutzer an der Weiterentwicklung ein höheres Interesse haben und ebenfalls daran, wie die entwickelten Anwendungen weiterverwendet werden. Eine niedrige Motivation kann dazu führen, dass Nutzer auf Basis eines minimal funktionierenden Standes aufhören, sich aktiv weiterzuentwickeln und nach der Erfüllung ihrer Aufgabe das Projekt einfach verlassen. Dadurch geht Wissen verloren und diese Nutzer transferieren ihre negativen Erfahrungen in die entsprechenden Fachabteilungen, was in diesem Fall einen negativen Einfluss haben kann.

Nach der Implementierung muss in einer Organisation ein erhöhter Monitoring-Aufwand eingeplant werden.<sup>225</sup> Bei der Einführung geht es um eine neue Plattform, die neue Arbeitsweisen ermöglicht. Dies erfordert eine gewisse Lernkurve bei den Nutzern und den Fachabteilungen. Dies sollte durch das Unternehmen akzeptiert werden und der entsprechende Raum zum Lernen gegeben sein. Einige Werkzeuge dieses Monitorings können

---

<sup>222</sup> Subcode: 13.2 Befähigung durch Community

<sup>223</sup> Subcode 13.4 Befähigung durch selbsternennen

<sup>224</sup> Subcode: 12.4 Vorbereitende Maßnahmen

<sup>225</sup> Hauptcode 14 Qualitätssicherung Projekt

Qualitätschecks sein, verbunden mit Rückkopplungsprozessen. Wenn bei der Anwendung einer Richtlinie ein Fehler aufgetreten ist oder ein Problem festgestellt wurde, sollte dies so schnell wie möglich korrigiert werden. Durch den Rückkopplungsprozess können diese Problemstellungen mit Lösungen dann in den entsprechenden Schulungsmaßnahmen und Prozessen angepasst werden. Ein weiteres Mittel der Qualitätssicherung ist das Nutzer-Feedback, das aus den bereits skalierten Abteilungen gesammelt wird.

### 8.3.4 Nachbereitung des Projekts

Mit dem Ende des Einführungsprojektes beginnt die aktive Phase der Nutzung. In diesem Zusammenhang sollte es einen Workshop mit dem Projektteam geben, in dem evaluiert wird, wie der Projektverlauf war. Dieses wertvolle Wissen kann für zukünftige ähnliche Implementierungsprojekte nützlich sein, allerdings ebenso für die aktive Phase des Betriebes. Problemstellungen und Best-Practice-Erfahrungen, die während der Implementierung festgestellt wurden und die den aktiven Betrieb beeinflussen, sollten aufgearbeitet und für den laufenden Betrieb berücksichtigt werden. Hier kann ein Lessons-Learned-Workshop nützlich sein.

Ein weiterer wichtiger Punkt, der nach der Einführung des Projektes erfolgen sollte, ist die Messung, wie erfolgreich das Projekt im aktiven Betrieb ist. Neben den traditionellen Projektmanagementwerten wie in Time, in Scope und in Budget können hier quantitative und qualitative Messmethoden verwendet werden.<sup>226</sup> Zur quantitativen Messung des Erfolgs können verschiedene KPIs zu unterschiedlichen Daten gebildet werden, welche häufig bereits durch die Plattformen bereitgestellt werden. Diese Kennzahlen ermöglichen eine Analyse, wie die Plattform genutzt wird, wie Anwendungen genutzt werden und wie die Nutzeradaption für die Plattform funktioniert.

Diese Kennzahlen können in Verbindung mit qualitativen Daten über die Plattformen eine wertvolle Datengrundlage bieten, um den Erfolg der Plattform im Unternehmen zu steigern. Qualitative Daten beziehen sich hierbei auf die Zufriedenheit mit der Plattform und geben einen tieferen Detailblick auf Problemstellungen, welche Nutzer bei der Verwendung der Plattform erfahren haben. Es kann ebenso evaluiert werden, ob sich Prozesse durch die Verwendung verändert haben und welcher Mehrwert durch die Nutzer für das Unternehmen gesehen wird. Es können durch die Evaluation der qualitativen Daten weitere Problemstellungen entdeckt werden, die behoben werden müssen bzw. aufzeigen, ob es weiteren Handlungsbedarf bei den bestehenden Richtlinien gibt. Diese qualitativen Daten können auf unterschiedliche Weise erhoben werden, als Beispiele werden dabei die Möglichkeiten von strukturierten Interviews und Fragebögen sowie das Tool der Retrospektiven/Lessons-Learned-Workshop genannt.

## 8.4 Begrenzung der Forschung

Die in dieser Arbeit dargelegte Forschungsthematik hat einen starken Fokus auf ein Einführungsprojekt, welches die vollständige Einführung von NCLC-Anwendungen als Grundannahme versteht. Dies ist jedoch nicht immer der Fall bei der Einführung, da es ebenso Unternehmen gibt, die diese Plattformen ohne das Konzept Citizen Developer implementieren. Dieser Aspekt wurde nachteilig behandelt und es sollte für weiterführende Forschung

---

<sup>226</sup> Hauptcode 15 Erfogsmessung

in Erwägung gezogen werden, ob es im Vorgehen zwischen diesen beiden Einführungs-konzepten Unterschiede gibt. In der hier vorliegenden Arbeit wurde allerdings die volle Berücksichtigung des Citizen-Developer-Konzepts bevorzugt, weil diese Art der Einführung ebenfalls durch die Experten als bessere Wahl betrachtet wurde. Zielgruppe dieser Arbeit sind mittelständische und große Unternehmen, die Ergebnisse dieser Studie sind nicht für kleine Unternehmen ausgelegt. Es wurde in den Interviews durch einige Experten angesprochen, dass dort Unterschiede bei der Implementierung existieren, die sich auf die Unternehmensgröße zurückführen lassen. Dabei können zwar einige Aspekte dieser Arbeit in einem Einführungsprojekt in kleineren Unternehmen übernommen werden, es sollte jedoch eine genauere Untersuchung der Unterschiede stattfinden und evaluiert werden, welches Vorgehen sich für kleinere Unternehmen eignet.

Für die Interviews war es von Vorteil, dass die Interviewpartner Teil verschiedener Organisationen waren und verschiedene Erfahrungen mit NCLC-Plattformen gesammelt hatten. Dadurch konnte Wissen über ein weitreichendes Feld von Unternehmen und Organisationen erlangt werden. Dies zeigt sich in der Vielfältigkeit der Antworten und den damit verbundenen verschiedenen Lösungsansätzen für Problemstellungen. Aufgrund dieser Vielfältigkeit war es aber zugleich eine Hürde, eine Detailbeschreibung von Tools des Vorgehens während der Umsetzung zu erhalten. Dies hatte Auswirkungen für die Detailplanung des Vorgehens, die teilweise nur ungenau beschrieben werden konnte. Es wurde versucht, die Empfehlungen der einzelnen Experten auf ein entsprechendes Abstraktionsniveau zu erhöhen, bei dem sich zum Teil allgemeine Aussagen ableiten ließen. Diesbezüglich besteht allerdings weiterer Forschungsbedarf, um zu evaluieren, welche Tools sich in der Detailplanung eignen.

Als sehr positiv wird betrachtet, dass die Interviewpartner einen Leitfaden zum Interview vorher zur Verfügung gestellt bekommen hatten. Dadurch konnte die Qualität der Antworten gesteigert werden. Dies wurde so durch Experten während oder im Vorgespräch des Interviews bestätigt. Es hat sich allerdings das Problem herausgestellt, dass der Interviewleitfaden keinem ausführlichen Pretest unterzogen worden ist, weshalb es zu Unklarheiten während der Interviews kam und in einigen Fällen die Experten zuerst die aktive Phase des Betriebs einer NCLC-Plattform beschrieben haben, statt die Einführungsphase einer solchen Plattform. Dieses Missverständnis musste durch Nachfragen oder Hinweise korrigiert werden. In zukünftigen Projekten sollte ein ausführlicher Pretest des Interviewleitfadens durchgeführt werden sowie des Leitfadens, der an die Experten verschickt wird, um dem entgegenzuwirken.

Im Rückblick hat es sich in diesem Zusammenhang als positiv herausgestellt, dass die Möglichkeit für Rückfragen durch den Interviewer bestand, um ein besseres Ergebnis erzielen zu können. Diese Befragungsform hatte jedoch den Nachteil, dass das Datenset auf die acht Interviews beschränkt war und bei einem größeren Datenset eventuell sich weitere Möglichkeiten für Vorgehen aufgezeigt hätten. Eine weitere Beschränkung des Datensets ist, dass alle Experten aus Deutschland waren und in Deutschland arbeiten. Hier könnte sich für zukünftige Forschungen ein Abgleich mit Experten aus anderen Regionen als Deutschland als positiv herausstellen.

Die computergestützte Analyse hat sich als sinnvolle Ergänzung für die Bearbeitung der Interviews herausgestellt. Dies hatte einen vorteilhaften Einfluss auf die Bearbeitungszeit, die für die Transkription der Interviews, die Kodierung sowie für die Erstellung des Summary Grids benötigt wurde. Allerdings gab es einige Probleme zu Beginn der Transkription, da in einem ersten Versuch das gewählte Programm MAXQDA die Zeitstempel der Interviews falsch dargestellt hat. Eine ausführliche Befähigung auf das Programm war ebenfalls nötig,

um den Umfang der bereitgestellten Tools des Programms zu erfassen. Dies hatte einen erhöhten Zeitaufwand zur Folge.

Die gewählte Methodik der qualitativen Inhaltsanalyse hat sich für die Erreichung des Forschungsziels als förderlich erwiesen, wodurch die Forschungsfragen beantwortet werden konnten. Die Methodik nach Kuckartz hat viele strukturelle Aspekte des Interviews hervorgebracht. Vor allem die Praxistauglichkeit der Methode ist hier hervorzuheben. Als Nachteil ist allerdings zu nennen, dass eine qualitative Inhaltsanalyse trotz forschungsorientiertem Design gewissen subjektiven Faktoren des Durchführenden unterliegt.<sup>227</sup>

Die gewählte Form der Interviews hat weitreichende Erkenntnisse ermöglicht, die in Verbindung mit der qualitativen Inhaltsanalyse zu einem zufriedenstellenden Ergebnis geführt hat.

---

<sup>227</sup> vgl. Kuckartz and Radiker 2022, S. 30.

## 9 Schlussbetrachtung

Die dieser Arbeit zugrundeliegende Fragestellung konnte mithilfe der Experteninterviews erfolgreich beantwortet werden. Es wurde klar herausgestellt, welche wichtigen Aspekte von NCLC-Plattformen deutsche Unternehmen im Vorfeld der Einführung beachten sollten und welche Erwartungen diese an den Funktionsumfang stellen können. Hierbei kann es in der weiteren Forschung interessant sein, zu evaluieren, ob bestimmte Aspekte für verschiedene Unternehmen und Branchen unterschiedlich wichtig sind. Wichtige Aspekte, die sich durch die Analyse herauskristallisiert haben, wirken sich in den Bereichen Unternehmenskultur, Prozessveränderung und Veränderung der Arbeitsweisen aus. Die Gruppe der Experten ist mit einer Anzahl von acht jedoch gering. Hier könnte sich eine weitere Befragung anbieten, um weitere Aspekte zu evaluieren, die bei einer größeren Anzahl von Befragungen gefunden werden könnten.

In der Expertenbefragung hat sich herausgestellt, dass die Phase vor der Einführung als sehr wichtig empfunden wird. In dieser Phase sollte sich intensiv mit der Plattform und den Anforderungen des Unternehmens auseinandergesetzt werden. An diesem Punkt wird ein erhöhter Planungsaufwand gesehen, um Regularien innerhalb des Unternehmens zu erstellen oder zu evaluieren. Anhand dieser Regularien kann eine Auswahl der Plattform durchgeführt werden. Wegen des klar planbaren Projektumfangs und der Charakteristika des klassischen Projektmanagements sind hier zwei verschiedene Vorgehensweisen möglich. Diese Vorphase kann entweder als eigenständiges kleines Wasserfallprojekt umgesetzt werden oder als Teil eines größeren hybriden Vorgehensmodells. Beim hybriden Vorgehen würde das umliegende Rahmenkonstrukt wasserfallartig angelegt und die Vorphase als klassisches Vorgehen realisiert werden. Eine Evaluation, welche von beiden Vorgehensweisen den bestmöglichen Erfolg für die Gesamtintegration der Plattform bringt, könnte in einer weiteren Forschungsarbeit herausgestellt werden. Hierbei ist hervorzuheben, dass einige Experten in dieser Phase eher ein agiles Projektmanagement als sinnvoll ansehen. Es müsste ebenfalls untersucht werden, ob ein agiles Vorgehen bei so starken Planungsanteilen eine Verbesserung herbeiführt gegenüber dem traditionellen Projektmanagement.

Wichtige Vorbedingungen, deren Erfüllung sich als positiv für das Projekt erweisen, sind verschiedene organisatorische Vorbereitungen, zu denen eine Evaluation der im Unternehmen vorhandenen Prozesse gehört. Ebenfalls berücksichtigt werden sollten die Themenbereiche Governance-, Daten- und IT-Sicherheitsrichtlinien. Die richtigen Rahmenbedingungen können meist nur durch das Unternehmen selbst erstellt werden, weshalb es wichtig ist, dass ein Unternehmen hier bereits proaktiv agiert. Des Weiteren wird zu der Implementierung eines Changemanagements geraten, um die Organisation und die Personengruppen der Organisation auf die Veränderungen durch die Plattform vorzubereiten. Aufgrund der kleinen Anzahl an Befragten kann eine Problemstellung sein, dass nicht alle wichtigen Faktoren erfasst werden konnten. Es kann sich für zukünftige Forschungen als sinnvoll erweisen, an dieser Stelle eine Befragung eines größeren Kreises von Projektmanagern durchzuführen und weitere Faktoren zu identifizieren.

Abhängig davon, ob eine hybride Vorgehensweise zu Beginn des Projektes gewählt wurde oder ein 2. Projektdesign, wird in der Umsetzungsphase der Implementierung ein neues Projekt begonnen oder es wird in eine neue Phase übergegangen. Hier wurde durch die Experten die dominante Meinung vertreten, dass diese Umsetzungsphase wegen der

schnellen Entwicklungsfähigkeit der Plattformen agil gestaltet werden sollte. Jedoch wurde ebenfalls darauf hingewiesen, dass eine agile Vorgehensweise für einige Unternehmen mit Problemen verbunden sein kann. Diese Probleme treten in Fällen auf, wo das Unternehmen oder die Organisationseinheit mangelhafte Kenntnisse im Bereich des agilen Vorgehens besitzt. Als Lösungsansatz wurde die Möglichkeit ermittelt, einen initialen Befähigungsworkshop für die agile Vorgehensweise durchzuführen. Wie das Design dieser initialen Schulung für eine optimale Adaption der agilen Vorgehensweise aussehen könnte, wäre zu evaluieren und in einer Praxisstudie zu testen. Es sollte ebenso evaluiert werden, ob eine solche initiale Befähigung ausreichend ist, um die agile Vorgehensweise sinnvoll anwenden zu können.

Durch die Analyse wurde als bevorzugte Methode der agilen Vorgehensweise die Scrum-Methode bzw. das Vorgehen in Sprints für Einführungsprojekte im Bereich NCLC ermittelt. Dies ermöglicht ein schnelles Reagieren auf geänderte Anforderungen sowie auf Feedback, das dem Projektteam auf verschiedene Weise zur Verfügung steht. Dieses Vorgehen löst auch die Fragestellungen, wie mit Änderungswünschen und geänderten Anforderungen umgegangen wird, weil diese der Methodik entsprechend in den Product Backlog oder Sprint Backlog übernommen wird. Ein größeres Change-Request-Verfahren kann nötig werden, wenn Änderungen grundlegender Art entstehen, die auf einer höheren Ebene des Projekts verarbeitet werden müssen. Als weiteres Tool, das als Ergänzung zur Scrum-Methodik verwendet werden kann, wurde die Kanban-Methodik ermittelt. Diese wird allerdings nicht als dominierendes Vorgehen empfohlen, sondern als Ergänzung zum Scrum-Vorgehen.

Beim Thema Skalierung wird davon abgeraten, die Plattform in einem Schritt für die gesamte Organisation freizuschalten. Dies kann erhebliche Ressourcen-Aufwendungen mit sich bringen, zusätzlich zu Problemen des Monitorings. Es wird eher ein Vorgehen empfohlen, das zumindest zu Beginn schrittweise erfolgt. Dabei wird neben der Pilotabteilung, wo die initiale Einführung erfolgt, ein Roll-out in einer weiteren Abteilung durchgeführt. Begleitend zu diesem Roll-out wird ein Feedback-System verwendet, um Erfahrungen, die bei der Skalierung gesammelt werden konnten, aufzuarbeiten und für die weiteren Skalierungsiterationen zu verwenden. Es ist möglich, ab einem bestimmten Punkt in mehreren Abteilungen gleichzeitig den Roll-out-Prozess durchzuführen. Dabei sollten die in der Organisation vorhandenen Ressourcen berücksichtigt werden, um diesen entsprechend begleiten zu können, ohne dabei einen Kontrollverlust zu riskieren.

Ein weiteres wichtiges Thema ist die Befähigung der zukünftigen Anwendungsentwickler auf die Plattform. Es wurden in diesem Themenbereich vier Möglichkeiten zur Befähigung durch die Experten genannt: ein klassisches Schulungssystem bzw. Wissenstransfer über Workshops, Communitybildung, Wissenstransfer durch Multiplikatoren und das selbstständige Aneignen von Wissen. Diese wurden von unterschiedlichen Experten benannt und es wäre in diesem Zusammenhang interessant, welche Befähigungsmethodik sich für welche Art von Mitarbeitern als empfehlenswert erweist. Zusätzlich könnte ebenfalls in einer größer angelegten Befragung ermittelt werden, ob es weitere Befähigungsmethodiken gibt und wie diese gestaltet wären.

Zur Qualitätssicherung des Projekts wurden verschiedene qualitative und quantitative Möglichkeiten ermittelt. Im quantitativen Bereich können unter anderem Kennzahlen zum Projektfortschritt, zu Projektkosten und zum Projektzeitplan ermittelt werden. Für den qualitativen Bereich lässt sich die Kreierung und Erhaltung der Motivation der Projektmitglieder hervorheben. Hier könnte es interessant sein, welche Methoden zum Motivationserhalt sich am besten eignen. Eng verbunden mit diesem Themenbereich ist die Thematik der Erfolgsmessung im Anschluss an die Einführung. Hierfür werden durch die Experten verschiedene

qualitative Evaluationsmethoden wie Retrospektiven und Befragungsbögen genannt. Bei der Einführung von NCLC-Plattformen handelt es sich ebenfalls um die Einführung einer Software, die oftmals weitreichende Monitoring-Tools besitzt. Diese Tools können zur Evaluation quantitativer Daten genutzt werden, um diese später zu interpretieren und den Erfolg zu messen. Teil einer zukünftigen Forschung könnte es sein, ein vollständiges Kennzahlenset zu ermitteln und zu evaluieren, welche Rückschlüsse diese Daten auf den Projekterfolg zulassen.

Schlussendlich kann festgehalten werden, dass es möglich war, ein Vorgehensmodell zu evaluieren, mit dessen Hilfe NCLC-Plattformen eingeführt werden können.



## 10 Literaturverzeichnis

AP Verlag. 2022. manage it | IT-Strategien und Lösungen. Available at <<https://ap-verlag.de/die-digitalisierung-forciert-die-low-code-entwicklung-was-low-code-plattformen-konkret-bieten-sollten/75010/>>. Accessed 30 June 2022

Augsten, Stephan. 2019. Keine Angst vor Citizen Developers. Available at <<https://www.dev-insider.de/keine-angst-vor-citizen-developers-a-841609/>>. Accessed 4 August 2022

Baur, Nina, und Jörg Blasius. 2014. *Handbuch Methoden der empirischen Sozialforschung*. Wiesbaden: Springer VS, ISBN978-3-531-18939-0.

Beck, Kent, Mike Beedle, Alistair Cockburn, und Ward Cunningham. 2001. Manifesto for Agile Software Development. *Manifesto for Agile Software Development*. Available at <<http://agilemanifesto.org/>>. Accessed 14 May 2022

Becker, Lutz, ed. 2013. *Basiswissen Projektmanagement. 1: Grundlagen der Projektarbeit / mit Beitr. von Lutz Becker*. 1. Aufl., ISBN978-3-86329-597-4.

Benac, Ryan, und Khan Mohd. 2022. *Proceedings of the Future Technologies Conference (FTC) 2021. Volume 3*. Edited by Kohei Arai. Lecture notes in networks and systems 360. Cham: Springer, ISBN978-3-030-89912-7.

Bertagnolli, Frank. 2022. *Lean management: introduction and in-depth study of Japanese management philosophy*. Wiesbaden [Heidelberg]: Springer, ISBN978-3-658-36087-0.

Bertram, Horst, Peter Blönnigen, und Adolf-Peter Bröhl. 1993. *CASE in der Praxis Softwareentwicklungsumgebungen für Informationssysteme*. Available at <<https://doi.org/10.1007/978-3-642-95690-4>>. Accessed 25 May 2022

Bigelow, Stephen J. 2020. Die Vor- und Nachteile von Low-Code-Plattformen bewerten. *ComputerWeekly.de*. Available at <<https://www.computerweekly.com/de/tipp/Die-Vor-und-Nachteile-von-Low-Code-Plattformen-bewerten>>. Accessed 7 July 2022

Bitkom, IDC, ed. 2022. ITK-Marktzahlen. Available at <[https://www.bitkom.org/Marktdaten/ITK-Konjunktur/ITK-Markt-Deutschland.html#:~:text=In%20der%20Informationstechnik%20wird%20f%C3%BCr,Euro%20das%20gr%C3%B6%C3%9Fte%20Wachstum%20verzeichnen](https://www.bitkom.org/Marktdaten/ITK-Konjunktur/ITK-Markt-Deutschland.html#:~:text=In%20der%20Informationstechnik%20wird%20f%C3%BCr,Euro%20das%20gr%C3%B6%C3%9Fte%20Wachstum%20verzeichnen.)>. Accessed 30 June 2022

Bock, Alexander C. und Ulrich Frank. 2021. Low-Code Platform. *Business & Information Systems Engineering* 63 (6): 733–740.

Bogner, Alexander, Beate Littig und Wolfgang Menz. 2014. *Interviews mit Experten: eine praxisorientierte Einführung*. Qualitative Sozialforschung. Wiesbaden: Springer VS, ISBN978-3-531-

19416-5.

Borgman, Hans. 2022. *Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS): January 4-7, 2022, Hyatt Regency Maui, Hawaii, USA*. Honolulu, HI: University of Hawai'i at Manoa, Hamilton Library, ISBN978-0-9981331-5-7.

Borgman, Hans und Doortje Hoogsteen. 2022. *Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS): January 4-7, 2022, Hyatt Regency Maui, Hawaii, USA*. Honolulu, HI: University of Hawai'i at Manoa, Hamilton Library, ISBN978-0-9981331-5-7.

ComputerWoche. 2021. Anwendungsentwicklung: Mit Low-Code gegen den IT-Fachkräftemangel. Available at <<https://www.computerwoche.de/a/mit-low-code-gegen-den-it-fachkraeftemangel,3552246>>. Accessed 30 June 2022<https://www.computerwoche.de/a/mit-low-code-gegen-den-it-fachkraeftemangel,3552246>.

Dechange, André. 2020. *Projektmanagement - schnell erfasst*. Wirtschaft - schnell erfasst. Berlin [Heidelberg]: Springer Gabler, ISBN978-3-662-57667-0.

Destatis. 2022. Statistisches Bundesamt Deutschland - GENESIS-Online. Text. Available at <<https://www-genesis.destatis.de/genesis/online?operation=previous&levelindex=1&step=1&titel=Ergebnis&levelid=1652014586721&acceptscookies=false#abreadcrumb>>. Accessed 8 May 2022<https://www-genesis.destatis.de/genesis/online?operation=previous&levelindex=1&step=1&titel=Ergebnis&levelid=1652014586721&acceptscookies=false#abreadcrumb>.

Detscher, Stefan, ed. 2021. *Digitales Management und Marketing: so nutzen Unternehmen die Markchancen der Digitalisierung*. Wiesbaden [Heidelberg]: Springer Gabler, ISBN978-3-658-33731-5.

Deutscher Industrie- und Handelskammertag. 2021. Fachkräftemangel – mit gravierenden Folgen. Available at <<https://www.dihk.de/de/themen-und-positionen/fachkraefte/beschaeftigung/fachkraeftereport-2021/fachkraeftemangel-mit-gravierenden-folgen--61818>>. Accessed 30 June 2022<https://www.dihk.de/de/themen-und-positionen/fachkraefte/beschaeftigung/fachkraeftereport-2021/fachkraeftemangel-mit-gravierenden-folgen--61818>.

Döring, Nicola und Jürgen Bortz. 2016. *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften*. 5. vollständig überarbeitete, Aktualisierte und erweiterte Auflage. Springer-Lehrbuch. Berlin Heidelberg: Springer, ISBN978-3-642-41089-5.

Dräger, Erich. 2020. *Ein kompetenzorientiertes, generisches Vorgehensmodell zur Einführung einer prozessorientierten Arbeitsorganisation*. Chemnitz.

Epping, Thomas. 2011. *Kanban für die Softwareentwicklung*. Informatik im Fokus. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, ISBN978-3-642-22595-6.

Europäische Union, Amtsblatt. 2003. Definition von kleineren und mittleren Unternehmen. Available at <<https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32003H0361&from=DE>>. Accessed 5 May 2022<https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32003H0361&from=DE>.

Freier, Sebastian. 2022. Low-Code-Idee: Auch ohne IT-Studium fit für die Jobs der Zukunft. *DIE WELT*. Available at <<https://www.welt.de/wirtschaft/karriere/bildung/plus236436799/Low-Code-Idee-Auch-ohne-IT-Studium-fit-fuer-die-Jobs-der-Zukunft.html>>. Accessed 30 June 2022<https://www.welt.de/wirtschaft/karriere/bildung/plus236436799/Low-Code-Idee-Auch-ohne-IT-Studium-fit-fuer-die-Jobs-der-Zukunft.html>.

Gillmann, Barbara. 2021. IW-MINT-Report : Mehr als 275.000 fehlende Fachkräfte – Mangel an

Experten schon jetzt größer als vor Corona. Available at <<https://www.handelsblatt.com/politik/deutschland/iw-mint-report-mehr-als-275-000-fehlende-fachkraefte-mangel-an-experten-schon-jetzt-groesser-als-vor-corona/27821334.html>>. Accessed 4 August 2022<<https://www.handelsblatt.com/politik/deutschland/iw-mint-report-mehr-als-275-000-fehlende-fachkraefte-mangel-an-experten-schon-jetzt-groesser-als-vor-corona/27821334.html>>.

Glatzer, Jens. 2013. *Schönheit: Ein Klärungsversuch*. Available at <<https://doi.org/10.1515/9783110320343>>. Accessed 1 July 2020<<https://doi.org/10.1515/9783110320343>>, ISBN978-3-11-032034-3.

Habermann, Frank. 2013. Hybrides Projektmanagement — agile und klassische Vorgehensmodelle im Zusammenspiel. *HMD Praxis der Wirtschaftsinformatik* 50 (5): 93–102.

Heilmann, Monika. 2014. So gehen Sie souverän mit Killerphrasen um. Available at <[https://www.projektmagazin.de/artikel/so-gehen-sie-souveraen-mit-killerphrasen-um\\_1087289](https://www.projektmagazin.de/artikel/so-gehen-sie-souveraen-mit-killerphrasen-um_1087289)>. Accessed 21 May 2022<[https://www.projektmagazin.de/artikel/so-gehen-sie-souveraen-mit-killerphrasen-um\\_1087289](https://www.projektmagazin.de/artikel/so-gehen-sie-souveraen-mit-killerphrasen-um_1087289)>.

Helfferrich, Cornelia. 2011. *Die Qualität qualitativer Daten: Manual für die Durchführung qualitativer Interviews*. 4. Aufl. Lehrbuch. Wiesbaden: VS, Verl. für Sozialwiss, ISBN978-3-531-17382-5.

Hochschule Luzern. 2020. Auswahl der Erhebungsmethode. Available at <<https://www.empirical-methods.hslu.ch/forschungsprozess/qualitative-forschung/auswahl-der-erhebungsmethode/>>. Accessed 2 July 2020<<https://www.empirical-methods.hslu.ch/forschungsprozess/qualitative-forschung/auswahl-der-erhebungsmethode/>>.

Institute der deutschen Wirtschaft. 2020. Fachkräftemangel bleibt für Unternehmen akut. *iwd*. Available at <<https://www.iwd.de/artikel/fachkraeftemangel-bleibt-fuer-unternehmen-akut-491758/>>. Accessed 4 August 2022<<https://www.iwd.de/artikel/fachkraeftemangel-bleibt-fuer-unternehmen-akut-491758/>>.

Känel, Siegfried von. 2020. *Projekte und Projektmanagement*. Lehrbuch. Wiesbaden [Heidelberg]: Springer Gabler, ISBN978-3-658-30085-2.

Keßler, Heinrich und Georg Winkelhofer. 2004. *Projektmanagement*. Springer Verlag. Available at <<http://public.eblib.com/choice/PublicFullRecord.aspx?p=6295327>>. Accessed 9 May 2022<<http://public.eblib.com/choice/PublicFullRecord.aspx?p=6295327>>, ISBN978-3-642-17025-6.

Kneuper, Ralf und Günther Müller-Luschnat. 1998. *Vorgehensmodelle für die betriebliche Anwendungsentwicklung*. Edited by Andreas Oberweis. Available at <<https://doi.org/10.1007/978-3-663-05994-3>>. Accessed 29 May 2022<<https://doi.org/10.1007/978-3-663-05994-3>>, ISBN978-3-663-05994-3.

Komus, Ayelt und Moritz Kuberg. 2020. Status Quo Agile. Edited by Gesellschaft für Projektmanagement e.v. and Hochschule Koblenz. Available at <<https://www.process-and-project.net/studien/download/downloadbereich-status-quo-scaled-agile-2019-2020/>>. Accessed 26 May 2022<<https://www.process-and-project.net/studien/download/downloadbereich-status-quo-scaled-agile-2019-2020/>>.

König, Hans. 2020. Cybersecurity: Fachkräftemangel erhöht das Sicherheitsrisiko. *Computerwoche*. Available at <<https://www.computerwoche.de/a/fachkraeftemangel-erhoeht-das-sicherheitsrisiko,3550024>>. Accessed 7 July 2022<<https://www.computerwoche.de/a/fachkraeftemangel-erhoeht-das-sicherheitsrisiko,3550024>>.

Kopper, Andreas und Markus Westner. 2016. Deriving a Framework for Causes, Consequences,

- and Governance Shadow IT from Literature. Available at <<https://www.researchgate.net/publication/297698668>>. Accessed 7 July 2022<https://www.researchgate.net/publication/297698668>.
- Krcmar, Helmut. 2015. *Informationsmanagement*. Berlin, Heidelberg: Springer Berlin Heidelberg. Available at <<http://link.springer.com/10.1007/978-3-662-45863-1>>. Accessed 5 August 2022<http://link.springer.com/10.1007/978-3-662-45863-1>, ISBN978-3-662-45862-4.
- Krejci, Désirée, Satu Iho und Stephanie Missonier. 2021. Innovating with employees: an exploratory study of idea development on low-code development platforms: 17.
- Kuckartz, Udo und Stefan Rädiker. 2022. *Qualitative Inhaltsanalyse: Methoden, Praxis, Computerunterstützung: Grundlagentexte Methoden*. 5. Auflage. Grundlagentexte Methoden. Weinheim Basel: Beltz Juventa, ISBN978-3-7799-5533-7.
- Kuster, Jürg, Eugen Huber, Robert Lippmann, Alphons Schmid, Emil Schneider, Urs Witschi und Roger Wüst. 2011. *Handbuch Projektmanagement*. Handbuch Projektmanagement. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, ISBN978-3-642-21243-7.
- Lauer, Thomas. 2019. *Change Management: Grundlagen und Erfolgsfaktoren*. 3., Vollständig überarbeitete und erweiterte Auflage. Berlin [Heidelberg]: Springer Gabler, ISBN978-3-662-59102-4.
- Leimeister, Jan Marco. 2021. *Einführung in die Wirtschaftsinformatik*. Berlin, Heidelberg: Springer Berlin Heidelberg. Available at <<https://link.springer.com/10.1007/978-3-662-63560-5>>. Accessed 5 August 2022<https://link.springer.com/10.1007/978-3-662-63560-5>, ISBN978-3-662-63559-9.
- Lewin, Kurt. 1947. Frontiers in Group Dynamics: Concept, Method and Reality in Social Science; Social Equilibria and Social Change. *Human Relations* 1 (1): 5–41.
- Madauss, Bernd-J. 2020. *Projektmanagement: Theorie und Praxis aus einer Hand*. 8. Auflage. Berlin [Heidelberg]: Springer Vieweg, ISBN978-3-662-59384-4.
- Mayring, Philipp. 2015. *Qualitative Inhaltsanalyse: Grundlagen und Techniken*. 12., Überarb. Aufl. Weinheim Basel: Beltz, ISBN978-3-407-29393-0.
- Mayring, Philipp. 2016. *Einführung in die qualitative Sozialforschung: eine Anleitung zu qualitativem Denken*. 6., Überarbeitete Auflage. Pädagogik. Weinheim Basel: Beltz, ISBN978-3-407-25734-5.
- Mensing, Wilfried. 2015. *Erfolgreiches Projektmanagement ohne externe Berater in KMUs: Praxisleitfaden zur Etablierung Interner Projektmanager*. Wiesbaden: Springer Gabler, ISBN978-3-658-06663-5.
- Microsoft. 2022. Citizen Development: Das Center of Excellence bringt geordnete Verhältnisse in die Low-Code-Entwicklung. *Microsoft.de*. Available at <<https://microsoft.cio.de/a/das-center-of-excellence-bringt-geordnete-verhaeltnisse-in-die-low-code-entwicklung,3652961>>. Accessed 8 July 2022<https://microsoft.cio.de/a/das-center-of-excellence-bringt-geordnete-verhaeltnisse-in-die-low-code-entwicklung,3652961>.
- Paternò, Fabio und Volker Wulf, eds. 2017. *New Perspectives in End-User Development*. 1st ed. 2017. Cham: Springer International Publishing : Imprint: Springer, ISBN978-3-319-60291-2.
- Peters, Theo und Nicole Schelter. 2021. *Kompakte Einführung in das Projektmanagement: mit vielen praxisnahen Beispielen und modernen didaktischen Instrumenten*. Lehrbuch. Wiesbaden [Heidelberg]: Springer Gabler, ISBN978-3-658-31194-0.
- Raithel, Jürgen. 2008. *Quantitative Forschung: ein Praxiskurs*. 2., Durchgesehene Auflage.

- Lehrbuch. Wiesbaden: VS Verlag für Sozialwissenschaften, ISBN978-3-531-16181-5.
- Rechsteiner, Frank. 2016. *Erfolgreiches IT-Recruiting trotz Fachkräftemangel: Methoden zur Personalbeschaffung und -bindung*. Essentials. Wiesbaden: Springer Gabler, ISBN978-3-658-13158-6.
- Rentrop, Christian. 2019. Laienentwickler erfolgreich einsetzen. Available at <<https://www.dev-insider.de/laienentwickler-erfolgreich-einsetzen-a-840659/>>. Accessed 4 August 2022<https://www.dev-insider.de/laienentwickler-erfolgreich-einsetzen-a-840659/>.
- Richardson, Clay und John R Rymer. 2016. Vendor Landscape: The Fractured, Fertile Terrain Of Low-Code Application Platforms: 23.
- Roth, Sven L. und Thomas Heimann. 2022. IT wird Kern der Werschöpfung - Studie IT-Trends 2022. Available at <<https://www.capgemini.com/de-de/wp-content/uploads/sites/5/2022/03/Studie-IT-Trends-2022.pdf>><https://www.capgemini.com/de-de/wp-content/uploads/sites/5/2022/03/Studie-IT-Trends-2022.pdf>.
- Royce, Winston W. 1970. MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS: 11.
- Sanchis, Raquel, Óscar García-Perales, Francisco Fraile und Raul Poler. 2019. Low-Code as Enabler of Digital Transformation in Manufacturing Industry. *Applied Sciences* 10 (1): 12.
- SAP. 2022. What is low-code/no-code application development? | SAP Insights. *SAP*. Available at <<https://www.sap.com/insights/what-is-low-code-no-code.html>>. Accessed 7 July 2022<https://www.sap.com/insights/what-is-low-code-no-code.html>.
- Schaffry, Andreas. 2022. No-Code/ Low-Code 2022.
- Schareika, Nora. 2020. Young Professionals Survey 2020: Junge Fachkräfte wollen das Rundum-Sorglos-Paket. Available at <<https://www.wiwo.de/erfolg/trends/young-professionals-survey-2020-junge-fachkraefte-wollen-das-rundum-sorglos-paket/26239202.html>>. Accessed 9 August 2022<https://www.wiwo.de/erfolg/trends/young-professionals-survey-2020-junge-fachkraefte-wollen-das-rundum-sorglos-paket/26239202.html>.
- Schwaber, Ken und Jeff Sutherland. 2020. Scrum Guide. Available at <<https://scrumguides.org/index.html>>. Accessed 27 May 2022<https://scrumguides.org/index.html>.
- Seiferlein, Werner. 2022. *Change-Management & Co: Einsatz von relevanten Prozessen und Methoden*. essentials. Wiesbaden, Germany [Heidelberg]: Springer Vieweg, ISBN978-3-658-36326-0.
- Specht, Frank. 2020. MINT-Fächer: Frühjahrsreport 2020: Bedarf an IT-Experten trotz Corona ungebrochen. *Handelsblatt.com*. Available at <<https://www.handelsblatt.com/politik/deutschland/mint-faecher-fruehjahrsreport-2020-bedarf-an-it-experten-trotz-corona-ungebrochen/25880102.html>>. Accessed 30 June 2022<https://www.handelsblatt.com/politik/deutschland/mint-faecher-fruehjahrsreport-2020-bedarf-an-it-experten-trotz-corona-ungebrochen/25880102.html>.
- Specht, Frank. 2021. Personalsuche: Der deutschen Wirtschaft fehlen Hunderttausende Leute – es droht der Stillstand. Available at <<https://www.handelsblatt.com/karriere/personalsuche-der-deutschen-wirtschaft-fehlen-hunderttausende-leute-es-droht-der-stillstand-/27811796.html>>. Accessed 4 August 2022<https://www.handelsblatt.com/karriere/personalsuche-der-deutschen-wirtschaft-fehlen-hunderttausende-leute-es-droht-der-stillstand-/27811796.html>.
- Stier, Winfried. 1999a. *Empirische Forschungsmethoden*. Berlin, Heidelberg: Springer Berlin Heidelberg. Available at <<http://www.springerlink.com/content/978-3-642-58460-2>>. Accessed 1 July

2020<http://www.springerlink.com/content/978-3-642-58460-2>, ISBN978-3-642-58460-2.

Stier, Winfried. 1999b. *Empirische Forschungsmethoden*. Available at <<https://doi.org/10.1007/978-3-642-58460-2>>. Accessed 6 May 2022<https://doi.org/10.1007/978-3-642-58460-2>, ISBN978-3-642-58460-2.

Stöhler, Claudia, Claudia Förster und Lars Brehm. 2018. *Projektmanagement lehren: studentische Projekte erfolgreich konzipieren und durchführen*. Wiesbaden [Heidelberg]: Springer Fachmedien Wiesbaden GmbH, ISBN978-3-658-18279-3.

Strangle, Werner. 2022. Abstraktionsniveau . Online Lexikon für Psychologie und Pädagogik. Online-Enzyklopädie. Available at <<https://lexikon.stangl.eu/6911/abstraktionsniveau>>. Accessed 19 August 2022<https://lexikon.stangl.eu/6911/abstraktionsniveau>.

Süddeutsche Zeitung. 2015. SZ-Gipfel Zitate. *Süddeutsche.de*. Available at <<https://www.sueddeutsche.de/wirtschaft/zitate-des-tages-von-altmaier-und-co-1.2744728>>. Accessed 7 July 2022<https://www.sueddeutsche.de/wirtschaft/zitate-des-tages-von-altmaier-und-co-1.2744728>.

Takeda, Hitoshi und Andreas Meynert. 2013. *Das synchrone Produktionssystem: Just-in-time für das ganze Unternehmen*. 7. Aufl. Management competence. München: Vahlen, ISBN978-3-8006-4608-1.

Talesra, Khushi und Nagaraja G. S. 2021. Low-Code Platform for Application Development. *International Journal of Applied Engineering Research* 16 (5): 346.

Tay, Nicole. 2021. 7 Pros and Cons of Low-Code/No-Code. *Major Online Business and Marketing*. Available at <<https://blog.hslu.ch/majorobm/2021/04/05/7-pros-and-cons-of-low-code-no-code-ntsy-2-ua-192667621-1/>>. Accessed 7 July 2022<https://blog.hslu.ch/majorobm/2021/04/05/7-pros-and-cons-of-low-code-no-code-ntsy-2-ua-192667621-1/>.

techconsult GmbH, ed. 2021. Low-Code-/No-Code-Development Enabler der digitalen Transformation - Status quo und Planungen von Low-Code-/No-Code- Development-Plattformen in deutschen Unternehmen. Available at <[https://www.smapone.com/no-code-studie/?utm\\_term=low%20code%20studie&utm\\_campaign=DACH+%7C+Studien+%26+Trends&utm\\_source=google&utm\\_medium=cpc&hsa\\_acc=6453249265&hsa\\_cam=1697237023&hsa\\_grp=127727026722&hsa\\_ad=573710259549&hsa\\_src=g&hsa\\_tgt=kwd-1408975803035&hsa\\_kw=low%20code%20studie&hsa\\_mt=b&hsa\\_net=adwords&hsa\\_ver=3](https://www.smapone.com/no-code-studie/?utm_term=low%20code%20studie&utm_campaign=DACH+%7C+Studien+%26+Trends&utm_source=google&utm_medium=cpc&hsa_acc=6453249265&hsa_cam=1697237023&hsa_grp=127727026722&hsa_ad=573710259549&hsa_src=g&hsa_tgt=kwd-1408975803035&hsa_kw=low%20code%20studie&hsa_mt=b&hsa_net=adwords&hsa_ver=3)>. Accessed 2 June 2022[https://www.smapone.com/no-code-studie/?utm\\_term=low%20code%20studie&utm\\_campaign=DACH+%7C+Studien+%26+Trends&utm\\_source=google&utm\\_medium=cpc&hsa\\_acc=6453249265&hsa\\_cam=1697237023&hsa\\_grp=127727026722&hsa\\_ad=573710259549&hsa\\_src=g&hsa\\_tgt=kwd-1408975803035&hsa\\_kw=low%20code%20studie&hsa\\_mt=b&hsa\\_net=adwords&hsa\\_ver=3](https://www.smapone.com/no-code-studie/?utm_term=low%20code%20studie&utm_campaign=DACH+%7C+Studien+%26+Trends&utm_source=google&utm_medium=cpc&hsa_acc=6453249265&hsa_cam=1697237023&hsa_grp=127727026722&hsa_ad=573710259549&hsa_src=g&hsa_tgt=kwd-1408975803035&hsa_kw=low%20code%20studie&hsa_mt=b&hsa_net=adwords&hsa_ver=3).

Versteegen, Gerhard, ed. 2002. *Software Management: Beherrschung des Lifecycles*. Available at <<http://link.springer.com/openurl?genre=book&isbn=978-3-642-62712-5>>. Accessed 26 May 2022<http://link.springer.com/openurl?genre=book&isbn=978-3-642-62712-5>, ISBN978-3-642-56367-6.

Violino, Bob. 2021. 7 Fehler: Warum Low-Code bei Ihnen nicht läuft. Available at <<https://www.computerwoche.de/a/warum-low-code-bei-ihnen-nicht-laeuft,3552371>>. Accessed 4 August 2022<https://www.computerwoche.de/a/warum-low-code-bei-ihnen-nicht-laeuft,3552371>.

Waszkowski, Robert. 2019. Low-code platform for automating business processes in

manufacturing. *IFAC-PapersOnLine* 52 (10): 376–381.

Wellenreuther, Günter und Dieter Zastrow. 2008. *Automatisieren mit SPS — Theorie und Praxis*. Wiesbaden: Vieweg+Teubner. Available at <<http://link.springer.com/10.1007/978-3-8348-9018-4>>. Accessed 29 September 2022<http://link.springer.com/10.1007/978-3-8348-9018-4>, ISBN978-3-528-23910-7.

Witzel, Andreas und Herwig Reiter. 2021. *Das problemzentrierte Interview*. Preprint. SocArXiv. Available at <<https://osf.io/uetq8>>. Accessed 6 May 2022<https://osf.io/uetq8>.

Woo, Marcus. 2020. The Rise of No/Low Code Software Development—No Experience Needed? *Engineering* 6 (9): 960–961.

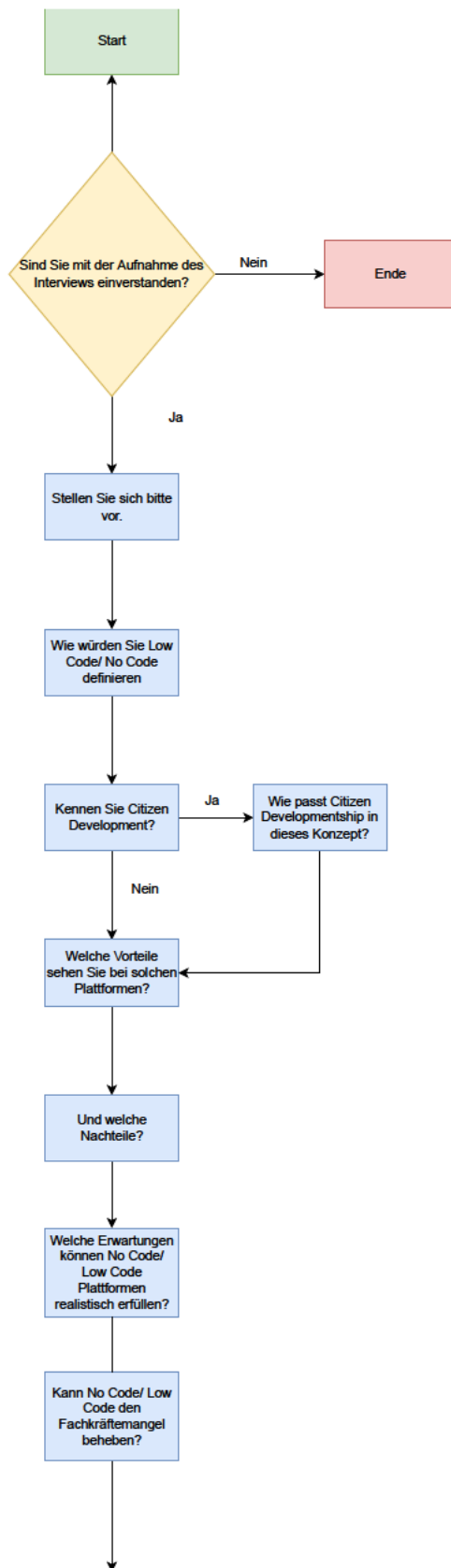
Zimmermann, Stephan und Christopher Rentrop. 2012. Schatten-IT. *HMD Praxis der Wirtschaftsinformatik* 49 (6): 60–68.

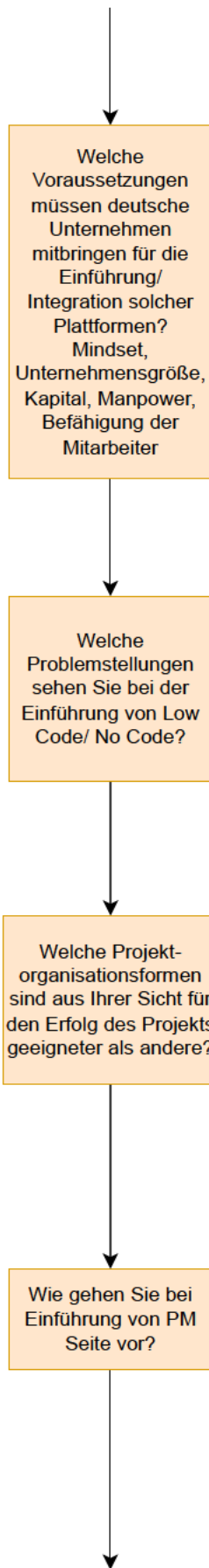
# Anlagen

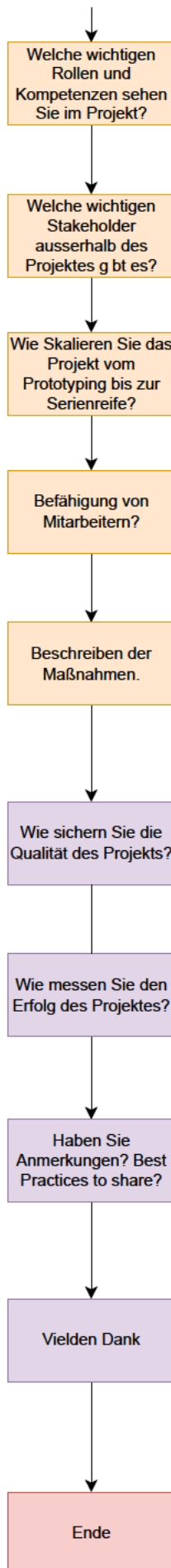
Anlage 1, Interviewleitfaden.....	A-I
Anlage 2, Interview EB 01.....	A-VI
Anlage 3, Interview EB 02.....	A-XXII
Anlage 4, Interview EB 03.....	A-XXXV
Anlage 5, Interview EB 04.....	A-LI
Anlage 6, Interview EB 05.....	A-LXIII
Anlage 7, Interview EB 06.....	A-LXXIX
Anlage 8, Interview EB 07.....	A-LXCIV
Anlage 9, Interview EB 08.....	A-XCIV
Anlage 10, Fallzusammenfassung.....	A-CXX
Anlage 11, Codebuch.....	A-CXXIX
Anlage 12, Summery Grid.....	A-CLVI

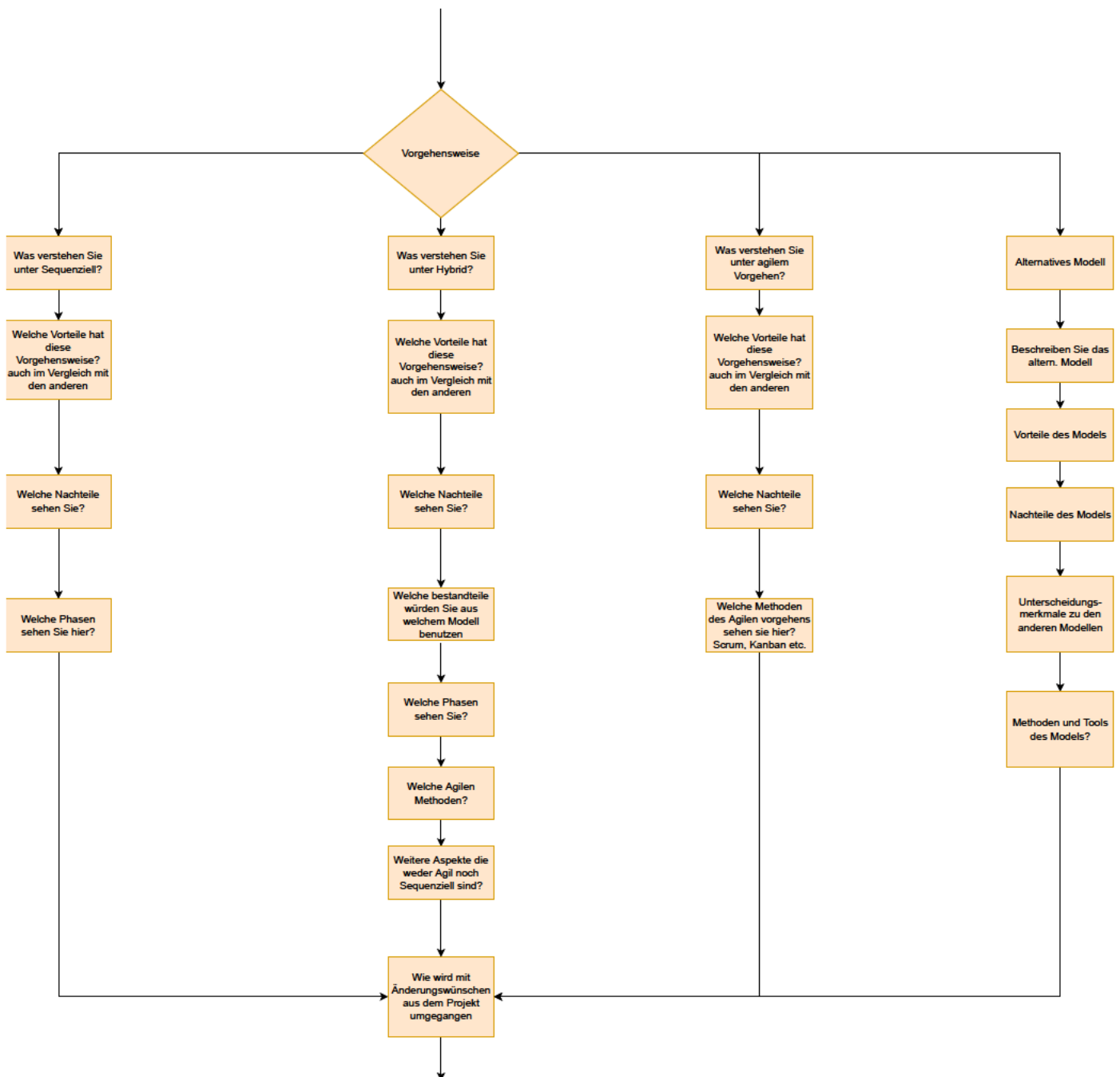


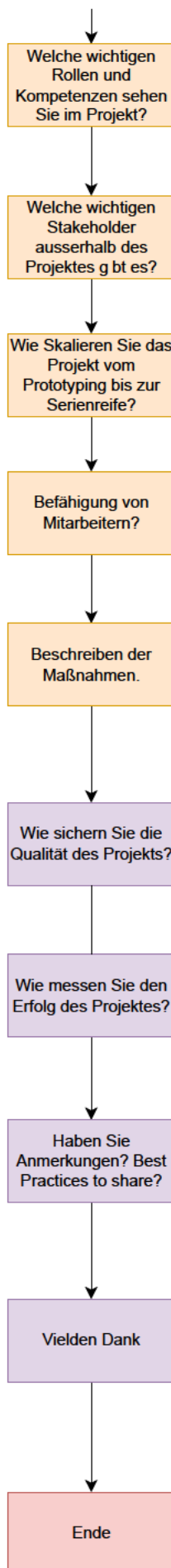
# Anlage 1, Interviewleitfaden











## Anlage 2, Interview EB01

RS: Rocco Schenk

EH: Eckhard Hauenherm

1	[0:00:00.0] <b>Start (EH - Eckhart Hauenherm/ RS - Rocco Schenk)</b>
2	[0:00:20.3] <b>I: RS</b> Ich begrüße Sie ganz herzlich zum Interview. Ich würde jetzt mit der Aufnahme starten. Sind Sie damit einverstanden?
3	[0:00:27.8] <b>B: EH</b> Ja bin ich.
4	[0:00:29.0] <b>I: RS</b> Ganz kurzer Datenschutzhinweis. Zu Transkriptionszwecken zeichne ich dieses Interview auf. Sie werden innerhalb der Masterarbeit nicht anonymisiert, aber ich werde alle persönlichen Daten, die während des Interviews aufkommen oder die ich von Ihnen gespeichert habe, im Nachhinein löschen. Außerdem werde ich diese meinen Betreuern zur Verfügung stellen, falls von denen Rückfragen an Sie kommen sollten.
5	[0:00:53.1] <b>B: EH</b> Ok, einverstanden.
6	[0:00:55.3] <b>I: RS</b> Dann würde ich gerne mit Ihnen einsteigen, indem Sie sich mal ganz kurz vorstellen. Sagen, wer sie sind, wie ihre berufliche Laufbahn, wie ihre wissenschaftliche Laufbahn so war?
7	[0:01:04.8] <b>B: EH</b> Ok also mein Name ist Eckhard Hauenherm, das ist schon mal wichtig. Ich bin promoviert in Germanistik, vor 20 Jahren. Bin während der Promotionen habe ich mich stark mit IT beschäftigt. Kann man tatsächlich so sagen. Bin dann nach der Promotion in ein IT-Beratungshaus eingestiegen und habe da IT-Projekte geleitet und Trainings gegeben und so weiter. Habe mich spezialisiert auf Microsoft-Umgebungen, mit Tools für die Zusammenarbeit, Colloboration, Kommunikation, was natürlich dann so ein bisschen auch mit meinem Studium zu tun hat. Also Kommunikationstheorien, Ansätze, Netzwerke und so weiter sind Themen, mit denen ich mich schon immer beschäftigt habe und bin seit 17 Jahren selbstständig. Berate Unternehmen, seit 7 Jahren ungefähr, bei den Wegen in die Cloud, muss man sagen. Also seit Microsoft sein Office 365 eingeführt hat, ist das ein sehr starkes Geschäft bei mir geworden. Und mit No Code/ Low Code beschäftige ich mich eigentlich schon relativ lange, weil im Microsoftumfeld, in einem der Hauptprodukte, mit denen ich gearbeitet habe, nämlich SharePoint Server, gab es natürlich immer schon eine No Code Workflow Engine. Eigentlich schon von Anfang an, also spätestens seit 2003, gab es eine Workflow Engine im SharePoint, mit denen ich so klassische Prozesse gestaltet habe. Und es gab so relativ, sagen wir mal, Code-arme Tools, die XML basiert zum Beispiel so Ansichten machten. Ich bin selber kein Entwickler, habe also nie wirklich programmiert und jetzt ist das natürlich gerade in meinem Umfeld Microsoft Cloud, Azure und mit Power Platform bin ich halt sehr stark drin. Und bin auch seit 2009 im PMI aktiv und leite seit 2 Jahren zusammen mit dem Peter Fey diese Community of Practice for Citizen Development beim Deutschland Chapter des PMI. Genau. Und habe an den Übersetzungen des PM BOK's der letzten beiden Versionen auch so ein bisschen mitgearbeitet, solche Sachen. Ja, gut, das ist so mein Werdegang.
8	[0:03:27.2] <b>I: RS</b> Dann lassen Sie uns doch direkt mit der Frage einsteigen. Wie würden Sie denn für sich No Code/ Low Code erst mal definieren?
9	[0:03:34.4] <b>B: EH</b> Gute Frage, definieren. No Code/ Low Code definiere ich als die Möglichkeit eines Nicht-Programmierers, eigene technisch basierte Prozesse/ Applikationen in einer

	<p>Plattform umzusetzen, um entweder sich selber die Arbeit zu vereinfachen oder um im Unternehmen neue Tools zur Verfügung zu stellen, sagen wir mal so. Genau, das wäre die Definition.</p>
10	<p>[0:04:10.6] <b>I: RS</b> Sie haben das Wort Citizen Development jetzt schon mal erwähnt. Ich gehe davon aus Sie kennen es. Wie passt dieses Konzept dann in die No Code/ Low Code-Plattform mit rein?</p>
11	<p>[0:04:19.6] <b>B: EH</b> Also ich verwende das fast synonym, würde ich sagen. Also Citizen Development ist sozusagen die Beschreibung der Tätigkeit und No Code/ Low Code sozusagen typischerweise die Beschreibung der Plattform. Also Citizen Developer oder Fachbereichsentwicklung ist, glaube ich, der deutsche Begriff, aber eigentlich hat sich der englische inzwischen, glaube ich, mehr eingebürgert, sind Leute, die mit No Code oder Low Code Plattformen im Unternehmen Applikationen herstellen. Das ist im Gegensatz zu der Idealform, wie es eigentlich gedacht ist, nicht unbedingt der normale Geschäftsanwender. Das sind schon nach meiner Erfahrung im Unternehmen eher die Leute, die eh schon so ein bisschen technisch orientiert sind. Also halt die Leute, die früher auch Makros in Excel geschrieben haben oder so was. Die gehen eher in die Richtung Citizen Development und manchmal sind es auch wirklich IT-Leute, die nicht wirklich Anwendungsentwickler sind und das jetzt nutzen, um ihre eigenen Prozesse zu verbessern oder zu automatisieren. Also ich mache viel in der Prozessautomatisierung damit, das kann ich schonmal sagen.</p>
12	<p>[0:05:23.7] <b>I: RS</b> Und wenn Sie auch praktischer Anwender selber sind, wie sehen Sie so die Vorteile von diesen No Code/ Low Code-Plattformen und vom Citizen Development?</p>
13	<p>[0:05:34.3] <b>B: EH</b> Die Vorteile sehe ich natürlich insbesondere darin, dass ich kein Kommunikationsproblem mit irgendwelchen Entwicklern habe, denen ich erst mal deutlich machen muss, was ich eigentlich will. Ich kann sozusagen selber meinen eigenen Prozess, so wie er mir vorschwebt, ohne Code-Kenntnisse mit den relativ einfachen Werkzeugen als logischen Prozess besser umsetzen. Also, das ist ein großer Vorteil. Das hat auch was mit Geschwindigkeit zu tun. Ich bin in den meisten Fällen dann schon relativ schnell fertig. Die Plattformen sind heute wirklich so mächtig und so gut, dass man wirklich in ein paar Stunden schon sehr gute Dinge zustande bringt. Ja, das sind die Vorteile. Sie haben ja nach den Vorteilen gefragt, die Nachteile kommen wahrscheinlich später.</p>
14	<p>[0:06:18.7] <b>I: RS</b> Mich würde auch interessieren, welche Problemstellungen und Nachteile Sie bei No Code/ Low Code-Plattformen sehen? Das würde mich auch interessieren.</p>
15	<p>[0:06:25.5] <b>B: EH</b> Ach doch. Sehe ich natürlich tatsächlich auch. Also es gibt natürlich erst mal aus Programmsicht so ein bisschen, sagen wir mal, die Qualität dessen, was da rauskommt, hängt natürlich sehr stark davon ab, wie gut auch das technische und das logische Verständnis desjenigen ist, der programmiert. Es ist wirklich unterschiedlicher Qualität oder, sagen wir mal so, die Qualitätsschwankungen im Vergleich zu einer, sagen wir mal, wirklich von Entwicklern getriebenen Entwicklung ist sicherlich da. Das Zweite ist natürlich aus Unternehmenssicht sehr stark, zu Fragen wie ein Application Life Cycle. Also wer betreut eigentlich die Anwendung weiter? Wer kümmert sich darum, dass die angewendet werden? Wer kümmert sich um die Sicherheit? Wo gehen die Daten eigentlich hin? Also all das, was so Governance, Application Life Cycle und so weiter ist. Das ist natürlich ein Punkt, über den man sehr stark nachdenken muss und das ist natürlich gerade bei der Einführung in Unternehmen dann eine der Kernfragen, mit denen ich auch gerade relativ häufig zu kämpfen habe. Oder zumindest zu reden habe, zu kämpfen habe ich damit weniger, aber das ist so ein Punkt, über den man explizit nachdenken muss, wie man damit umgeht. Das sind so die Nachteile, Anwendungsqualität, sage ich mal, und Einhaltung Governance.</p>

- 16 [0:07:42.7] **I: RS** Und wenn Sie sagen, Sie sind damit auch in Unternehmen. Wie sind Ihre Erfahrungen, was wirklich realistische Erwartungen von No Code/ Low Code-Plattformen sind? Oder die man an diese Plattformen stellen kann? Es gibt ja auch viel Hype darum, dass die IT damit komplett ersetzt wird, was sind aber wirklich realistische, also Ihrer Meinung nach, realistische Anforderungen?
- 17 [0:08:04.0] **B: EH** Also ich glaube nicht, dass man die IT damit ersetzen kann. Ich glaube, dass man die IT damit sehr stark unterstützen kann. Tatsächlich ist meine Erfahrung aus den Unternehmen, dass Low Code/ No Code schwerpunktmäßig in der IT eingesetzt wird und weniger in den Fachbereichen tatsächlich, also das, was man eigentlich will, da sind wir, glaube ich, noch nicht. Das hat wahrscheinlich auch was mit einem Rollenverständnis zu tun, denn das klassische Rollenverständnis ist so, für Applikationen ist die IT zuständig und deswegen machen die das auch. Es ist natürlich auch ein bisschen so, dass ich meine Anwender dann sozusagen dahin bringen muss und... Was war die Frage, jetzt habe ich vergessen, wo ich war?
- 18 [0:08:44.8] **I: RS** Was realistische Erwartungen sind, die man an No Code/ Low Code-Plattformen stellen kann?
- 19 [0:08:48.8] **B: EH** Realistische Erwartungen aus meiner Sicht, was man heutzutage sehr gut schafft, ist einfache Prozesse zu automatisieren. So klassische Geschäftsprozesse, wie Rechnungseingang und solche Dinge, Prozesse automatisieren und wo ich auch einen sehr starken Vorteil, aber ich weiß gar nicht, ob das so richtig in No Code/ Low Code fällt, sind diese ganzen BI Mechanismen. Bei Microsoft sind die mit in der Power Platform drin, deshalb fasse ich das da immer mit drunter. Das geht auch sehr gut, da sehe ich auch Leute, weil da wird nicht Fach-Know-How gefragt oder muss ich nur Datenstrukturen kennen. Wo ich sage, ist die Erwartungshaltung noch ein bisschen übertrieben, ist wirklich das Applikationen erstellen. Also das ich so Applikationen im Fachbereich erstelle, die extern zum Beispiel angewendet wird, in der Kundenkommunikation und so was. Da, glaube ich, sind wir noch nicht so weit und da weiß ich auch gar nicht, ob das wirklich so eine Zielsetzung sein kann. Also ich sehe die realistische Erwartung in der Umsetzung interner Prozesse, interner kleiner Applikationen, die ich selber nutze, um Daten zu pflegen, um Prozesse zu automatisieren. Das kann, würde ich sagen, ist eine realistische Erwartung. Das kann ich mit dem, was wir heute zur Verfügung haben, gut machen. Ja.
- 20 [0:10:07.5] **I: RS** Und wo wir schon beim Thema IT vorhin waren. Es gibt ja einen IT-Fachkräftemangel, auch innerhalb der Unternehmen. Kann No Code/ Low Code dazu beitragen, den zu beheben, den zu entlasten? Hat es gar keine Auswirkung darauf?
- 21 [0:10:25.9] **B: EH** Finde ich persönlich tatsächlich schwer einzuschätzen. Ich weiß aber, dass das in der Perspektive immer damit verbunden wird. Ich brauche keine Entwickler mehr, weil die Fachleute können das ja selber machen. Aber es ist ja nicht so, dass das Business nichts anderes zu tun hätte. Also das ist ja eine Milchmädchen-Rechnung zu sagen, ich leiste die IT-Arbeit, die ich jetzt nicht leisten kann, die gebe ich in die Fachabteilung und die haben ja Zeit dafür, die können das ja machen. Das würde ja nicht hinhauen. Ich habe vielleicht den Vorteil, dass ich mit anderen Qualifikationen arbeiten kann. Also es geht weniger von der rein technischen Qualifikation IT-Wissen mehr in die Logik, Businesslogik, die ich auskenne. Also ich kann Betriebswirtschaftler mit einem guten technischen Verständnis sicherlich dann auch an so was dransetzen. Insofern kann das tatsächlich ein Nutzen sein, die Frage ist, haben wir nur einen Fachkräftemangel bei der IT oder haben wir nicht auch einen Fachkräftemangel in anderen Bereichen. Weil wie gesagt, ich würde auch nicht sagen, dass No Code/ Low Code so was ist, was jeder mit Kindergartenwissen soeben mal machen kann, sondern bisschen Verstand muss man da schon hereinbringen.



- 22 [0:11:31.8] **I: RS** Wie würden Sie da die Voraussetzungen sehen für die Mitarbeiter? Also jetzt rein von ihrem Skillset her, damit die das anwenden könnten.
- 23 [0:11:40.7] **B: EH** Die müssen, sage ich mal, also ich muss immer so ein bisschen aufpassen. Wie gesagt, ich mache sehr viel mit Prozessen und Prozessautomatisierung, deswegen denke ich immer so in Prozesslogik. Also ein Verständnis für Prozesslogik, was halt ein guter Applikationsentwickler auch hat. Also wann brauche ich eine Schleife? Was bilde ich über Verzweigungen ab? Wie gehe ich mit Variablen um? Das sind ja so Dinge, soviel technisches Verständnis muss tatsächlich da sein, um wirklich gute, sagen wir mal, gute Anwendungen oder gute Umsetzungen in Low Code/ No Code machen zu können. Also ich muss einerseits ein gutes Verständnis davon haben, wie arbeitet eigentlich eine Applikation, also so ein gewisses technisches Grund-Know-how. Das ist logisches Verständnis. Eigentlich ist es, ich würde mal sagen, so ein klassisches mechanisches Verständnis. Wer mir erklären kann, wie ein Fahrrad funktioniert und warum unterschiedliche Übersetzungen unterschiedliche Geschwindigkeiten bewirken, der kann in der Regel auch einen Prozess ganz gut darstellen, sage ich mal. Das ist das eine. Also wirklich, das müssen Leute sein, die halt das, was sie machen, auch tatsächlich in solche Prozesse umsetzen können. Das ist eine Umsetzungsleistung, die erst mal in der Ausbildung vielleicht mit drin sein muss oder ähnlichen Dingen. Das Zweite ist natürlich, was gebraucht wird, ein Wissen über die Plattform selber. Das wird häufig auch unterschätzt. No Code/ Low Code, das sind ja auch nicht zwingend simple Plattformen. Also wenn ich an die Umgebung, in der ich am meisten arbeite, die Power Plattform, wenn ich da in die Formeln und solche Dinge, Azure, Functions und so was reingucke, dann ist da ja schon noch Code drin. Also ich muss schon so mathematische Klammerung verstehen, Funktionsaufrufe verstehen und solche Sachen. Also so was kommt auch noch mit rein und überhaupt, was ich in der Plattform machen kann. Die dritte Wissensbasis, die ich brauche, ist tatsächlich so ein Gefühl für Risiken und Governance, was den Umgang mit Daten angeht. Also die meisten Plattform sind extrem mächtig, was den Austausch von Daten zwischen unterschiedlichen Datenplattformen, Hostingplattformen und so weiter angeht. Und dass es vielleicht keine gute Idee ist, meinen aktuellen Geschäftsbericht in Drahte mal eben von, was weiß ich, von One Drive in die Dropbox zu kopieren, nur weil ich das automatisch machen kann und ich von da aus einen leichteren Zugriff habe. Das muss natürlich bei den Anwendern auch ankommen und das ist, was ich eben meinte, mit der Governance im Unternehmen. Da muss ich entweder eine Governance drüberlegen oder ich muss ein zumindest eine Awareness bei den Nutzern erzeugen dafür, dass nicht alles, was machbar ist, auch wirklich gut ist.
- 24 [0:14:15.8] **I: RS** Ganz kurz als Spoiler. Alle Fragen, die ich von jetzt ab stelle, sind immer unter dem Maßstab: Ein Unternehmen möchte gerne bei sich eine No Code/ Low Code-Plattform einführen und darauf beziehen sich quasi alle Fragen. Es ist immer dieses Ausgangsszenario. Es geht also mir nicht um die Situation, wenn es implementiert ist im Unternehmen, sondern in der Vorphase, in dem Einführungsprojekt.
- 25 [0:14:38.1] **B: EH** Ok.
- 26 [0:14:40.2] **I: RS** Sie haben ja schon gesagt, die Anwender im Unternehmen müssen ein gewisses Verständnis auch mitbringen. Sehen Sie noch andere Voraussetzungen, die deutsche Unternehmen mitbringen müssten, wenn Sie jetzt an die Einführung einer No Code/ Low Code-Plattform denken? Also da spielt die Frage so ein bisschen auf Mindset im Unternehmen ab. Muss das Unternehmen eine gewisse Größe haben? Ein gewisses Kapital dahinterstehen? Oder sogar eine gewisse Manpower, also eine gewisse Anzahl an Mitarbeitern da sein?
- 27 [0:15:07.2] **B: EH** Das ist eine schwierige Frage. Also ich würde sagen, Größe eigentlich nicht. Ich kann das sowohl in kleinen als auch in großen Unternehmen machen. Mindset würde ich schon sagen, weil Low Code/ No Code ist natürlich sozusagen schon eine sehr

agile Form des Arbeitens. Ich mache halt Sachen selber, die ich vorher vielleicht ausgelagert habe. Also ich muss mit flexiblem Rollenverständnis rangehen. Also Mindset, im Sinne von: so stark Hierarchie-abgegrenzte Rollen und so weiter, da funktioniert das meistens nicht, sondern ich muss sowohl aus, sagen wir mal, aus IT oder aus Operationssicht ein Verständnis dafür haben. "OK wir machen das jetzt selber und wir vertrauen den Anwendern auch so weit." Und gleichzeitig müssen die Anwender sagen: "Ok, ich übernehme auch Verantwortung für das, was ich hier tue." Also da würde ich schon sagen, dass das ein gewisser Mindset ist, der da existieren muss oder der herausgearbeitet werden muss. Das kann sich auch mit der Zeit entwickeln, das muss vielleicht nicht von Anfang an da sein, aber ich würde eben aus genau dieser Berücksichtigung so was nicht einfach flächendeckend im Unternehmen verteilen, sondern schon eine Einführungsstrategie mir überlegen, in welchen Ecken ich anfangen damit, um auch zu testen und zu probieren. Das würde ich schon sagen und sonst: Größe, technische Voraussetzungen sehe ich eigentlich wenig. Was natürlich sehr stark eine Rolle spielt, wenn man so ein bisschen darüber nachdenkt, die meisten Unternehmen, die ich kenne, die so was einführen, sind schon sehr stark in der Cloud vertreten. Also bilden viele Funktionalitäten im Cloudgeschäft ab oder haben halt Cloudplattformen im Einsatz. Also bei der reinen ON Prem Variante kommt No Code/ Low Code deutlich seltener vor.

- 28 [0:16:55.6] **I: RS** Ich weiß nicht, ob ich es jetzt nicht mitgekriegt habe, aber hatten Sie gesprochen, wie so die finanzielle Situation von so einem Unternehmen sein muss?
- 29 [0:17:04.7] **B: EH** Nee, da hatte ich nichts zu gesagt.
- 30 [0:17:06.0] **I: RS** Könnten Sie dazu noch etwas sagen bitte?
- 31 [0:17:07.8] **B: EH** Die finanzielle Situation des Unternehmens.
- 32 [0:17:11.0] **I: RS** Oder wie viel Kapital dahinterstehen muss? Muss da viel dahinterstehen oder sagen sie: "Ist ziemlich entspannt"?
- 33 [0:17:17.0] **B: EH** Also, ich muss die Lizenzen bezahlen können. Das würde ich sagen, das reicht. Also ich würde echt sagen, ich kann mir durchaus vorstellen, dass so ein Einzelunternehmer wie ich, also ich habe mir auch so kleine Prozesse auch automatisiert und da reicht halt meine Lizenz für aus. Ich würde jetzt sagen, nicht unbedingt, dass das Kapital sein muss, um mit Low Code/ No Code zu arbeiten. Das sehe ich einfach unabhängig davon. Es ist keine Riesen-Anfangsinvestition, die ich machen muss, deswegen kann ich auch mit wenig Kapital beginnen.
- 34 [0:17:47.2] **I: RS** Ok, und würden Sie Hürden sehen, auf die Unternehmen schnell stoßen können, wenn sie jetzt bei der Einführung sind, auf die Sie vielleicht achten sollten?
- 35 [0:17:59.7] **B: EH** Ja, wie schon eben erwähnt, die Governance. Das ist ein ganz wichtiger Punkt. Das hängt ein bisschen davon ab, mit welcher Plattform ich arbeite, aber die meisten Plattformen sind am Anfang immer relativ offen und bieten relativ viele Möglichkeiten und wenn ich bei der Einführung nicht darauf achte und erst mal mit offenem Scheunentor alle Anwender sozusagen da rauf lasse und die Daten von links und rechts verschieben lasse, auswerten lasse, irgendwelche Formulare dem Kunden zur Verfügung stelle, ohne an eine Datenschutzerklärung zu denken und solche Sachen. Da sehe ich tatsächlich eine Hürde, das kann gefährlich werden für ein Unternehmen. Also das ist jetzt keine Einführungshürde, aber es ist ein hohes Risiko, an das ein Unternehmen denken sollte.
- 36 [0:18:36.4] **I: RS** Gibt es noch andere Risiken, die Sie sehen oder ist Governance das Wichtigste und quasi Einzige?
- 37 [0:18:43.1] **B: EH** Einzige würde ich jetzt nicht sagen, aber ist zumindest für mich eines der zentralen Risiken. Man kann natürlich auch technische Risiken sehen, so was wie Datenverlust und solche Dinge. Also dass wirklich Daten verschwinden, gelöscht werden

	<p>versehentlich oder solche Dinge. Das ist aber in den meisten Plattformen, die ich kenne, eigentlich relativ gut händelbar. Und wie gesagt, technisch brauche ich auch nicht so wahnsinnig viel. Also Governance ist für mich immer so das Leuchtturm-Risiko. Also da versuche ich, den Blick drauf zu schärfen.</p>
38	<p>[0:19:17.0] <b>I: RS</b> Wenn Sie jetzt so ein Projekt einführen bei einem Unternehmen ist ja auch immer die Frage des Projektteams, wie sich das organisiert. Also Matrix-Organisation, werden die Mitarbeiter in den Abteilungen belassen und zu 50 % zum Beispiel ins Projekt ausgeliehen? Oder vertreten Sie eher die Meinung, die sollten zu 100 % im Projekt arbeiten, herausgelöst aus ihren Abteilungen? Also wie würden Sie das organisieren?</p>
39	<p>[0:19:44.2] <b>B: EH</b> Also, ich versuche ein gemischtes Team zusammenzustellen, das tatsächlich aus der IT Leute hat. Weil, ich brauche Leute, die die Plattform pflegen, die die Administration der Plattform machen, wo man darüber reden muss. Ich brauche Leute aus dem Fachbereich, die bereit sind, da mitzuarbeiten. Das muss jetzt kein 100 % Projektteam sein, das da mitarbeitet, weil meistens sind die Einführungen dann eher selber so eine Art agiles/ iteratives Projekt (<b>unverständlich</b>), das heißt, ich fange mit einem kleinen Piloten an. Probiere das ein bisschen aus. Guck mal, ob die Leute damit arbeiten können. Begleite das durch Schulungen und dann kann ich eigentlich zum Beispiel von Fachbereich zu Fachbereich gehen, oder mal strategisch zu überlegen, wie wollen wir das jetzt weiter ausrollen. Wo stellen wir das alles zur Verfügung, deshalb ist das schon eine Art Matrix-Organisation, aber typischerweise mit Teilzeit-Teams, die sich regelmäßig treffen, mal zusammensetzen. Das muss jetzt nicht kontinuierlich betreut werden. Also das ist jetzt nicht so, zumindest in dem Umfeld, in dem ich arbeite, ist das nicht so ein Riesenaufwand, diese Plattform einzuführen, wo ich dann wirklich Leute 3 Wochen lang oder ein halbes Jahr nur mit dem Projekt beschäftigen muss, sondern das kann ich schön nach und nach machen und das liebe ich eigentlich, das in kleinen Häppchen zu machen.</p>
40	<p>[0:21:06.4] <b>I: RS</b> Und von der Vorgehensweise her. Also Sie haben es ja schon leicht angesprochen, Sie machen das nach und nach. Ist das dann ein klassischer Wasserfallansatz oder ein hybrider Ansatz oder ist tatsächlich rein agil?</p>
41	<p>[0:21:22.5] <b>B: EH</b> Tatsächlich agil. Also das würde ich tatsächlich nicht als Wasserfall machen. Erstens, weil ich bei der Einführung zwar Ziele mir festlegen kann. Das mache ich bei jedem Cloud-Projekt, dass sozusagen die erste Frage ist, warum wollt ihr in die Cloud? Warum wollt ihr in diese Plattformen? Also gucken, was will das Unternehmen damit erreichen? Ich habe aber eine relativ schlechte Planbarkeit, weil ich erstens nicht genau weiß, welche Funktionen jetzt damit umgesetzt werden können oder wollen oder wo das Unternehmen arbeitet. Das spricht schon gegen einen klassischen, sequenziellen Wasserfallprozess, wo ich ja in der Regel ein fertiges Gewerk am Ende plane. Das habe ich hier nicht, sondern es ist eine Einführung einer Plattform, die sich dann sozusagen immer weiterentwickelt. Und die Plattformen selber werden ja bei den Anbietern auch agil entwickelt. Also so eine Power Plattform ist ja auch nicht fertig und ich führe die ein, sondern da kommen jeden Monat neue Funktionalitäten rein. Das heißt, ich kann das eigentlich nur agil machen und immer gucken, in kleinen Zielen: "Wo gucken wir jetzt? Was machen wir als nächstens? Was nehmen wir uns für die nächsten 4 Wochen vor oder 2 Wochen?" Also klassisches Scrum-Projekt oder klassisches agiles Projekt und das passt auch am besten zu den Plattformen selber. Das passt dann gut zusammen und man muss ja auch mit dem Produkt dann lernen. Das ist ja genau das, ich habe mit einem iterativen Projekt meine Feedbackschleife und kann dann sehr schön mit der Plattform lernen, was uns die im Unternehmen bringt oder nicht bringt. Und ich kann auch mit relativ wenig Kosten schnell wieder aussteigen, wenn ich sage: "Nee, das ist für uns nichts." Oder "Die Plattform ist für uns nichts. Wir nehmen mal eine andere." Das kann ich bei einem fertig geplanten, also bei einem klassischen, sequenziellen Projekt</p>

habe ich das nicht. Da habe ich am Ende alles eingeführt und keiner kann damit arbeiten. Das wollen wir nicht.

- 42 [0:23:02.7] **I: RS** Und würden Sie sagen es ist rein agil oder sagen Sie: "Ein, zwei Sachen würde ich schon noch aus dem klassischen heraus nehmen für zum Beispiel ein übergeordnetes Framework oder..." Jetzt die Frage an Sie ist es ganz klassisch rein Agil oder sind da auch ein paar Formen aus einem Wasserfallmodell, die da mit reingemischt werden, was es zu einem hybriden Modell machen würde?
- 43 [0:23:26.7] **B: EH** Zu einem hybriden Modell... Kann man muss man aber nicht. Also man könnte natürlich tatsächlich für so ein Projekt auch ein Wasserfall drum herum legen, sage ich mal, für so ein hybrides Modell. Sehe ich aber tatsächlich keinen Mehrwert drin. Es gibt ein paar Dinge die ich Anfangs in der Zielsetzung machen würde, die vielleicht so ein bisschen nicht hundertprozentig Agil sind und da ist wieder mein Lieblingsthema Governance da, das sollte man am Anfang entscheiden wie viel Risiken gehe ich damit ein und wie kann ich die Risiken streuen. Da muss am man tatsächlich ganz klar am Anfang mit IT und Operations und was weiß ich, Security Officer, wer auch immer im Unternehmen dafür zuständig ist, sich wirklich mal klar machen, welche Datenschutzerfordernungen haben wir? Mit welchen Daten gehen wir um? Manchmal führt das zu einem Vorlauf, in so einem Projekt, das ich erstmal mit Datenklassifikationen und so weiter anfangen muss zu arbeiten. Das ich sicherstelle das bestimmte Daten da nicht rauskommen. Aber das kann man auch Agil nach und nach machen in dem Mann, wenn man das Risiko kennt, tatsächlich guckt wo setzen wir das ein? Wem gebe ich Zugriff auf die Plattform und so weiter? Das kann man schon noch steuern.
- 44 [0:24:33.2] **I: RS** Und wenn Sie sagen, Sie sind jetzt bei einem rein agilen Vorgehen, jetzt auf die Einführung von No Code/ Low Code bezogen. Wie sehen Sie denn da die Vorteile, warum sie genau dieses bevorzugen gegenüber den anderen beiden? Sie haben jetzt die Nachteile vom Wasserfall gesagt, was sind jetzt die Vorteile vom agilen, jetzt mal gesprochen?
- 45 [0:24:53.4] **B: EH** Also, die Vorteile vom agilen sind, ich habe es schon ein bisschen so erwähnt, tatsächlich die Tatsache, dass ich relativ schnell Feedback kriege über das, was ich mache. Ich arbeite ja sozusagen in so einem Projekt typischerweise nicht einfach so, dass ich erst die Plattform einführe und dann mit Anwendern arbeite, sondern ich arbeite mit einigen Anwendern daran, die Plattform einzuführen und zu gucken, wie kann man die nutzen in ihrem Arbeitsbereich. Dann habe ich ein schnelles Feedback und kann damit wieder überlegen, was muss ich in der Konfiguration der Plattform machen für die nächste Gruppe von Anwendern. Also diese kurzen Feedbackschleifen sind ein extremer Vorteil in solchen Projekten. Zweitens kann ich sehr viel schneller reagieren, zum Beispiel, auf Änderungen, die vielleicht in der Plattform auch erfolgen. Ich habe eben schon gesagt, die Plattformen selber entwickeln sich agil weiter. Plötzlich kann ich etwas machen, was ich vor einem Monat noch nicht machen konnte. Kann ich sofort überlegen: "Sollen wir nicht mal versuchen, das umzusetzen. Hier, jetzt suchen wir mal 3 Leute. Machen mit denen mal 2 Wochen Sprint und versuchen das mal umzusetzen in irgendeiner Applikation, ob das was für euch wäre. Hört sich gut an, könntet ihr nutzen." So was kann ich machen. Die Schnelligkeit, auch Änderungen einarbeiten zu können, ist ein sehr starker Vorteil des agilen Ansatzes und damit natürlich auch verbunden, ohne große upfront Investitionen schnell wieder aus dem Projekt auszusteigen, wenn man feststellt: "Nee, das ist nichts für uns. Unsere Anwender ziehen nicht mit." Oder "Die Risiken sind uns zu groß." Oder "Die Plattform kann nicht das, was wir brauchen." Das sehe ich relativ schnell mit einem agilen Vorgehen. Das würde ich bei einem plangetriebenen Ansatz oder auch bei einem hybriden Ansatz habe ich es häufig so, dass ich sehr viele Vorinvestitionen habe, um erst mal einen Plan aufzubauen, um erst mal die Plattform zu checken, um erst mal die Plattform auszuwählen und so weiter und so fort. Das wird deutlich teurer und führt auch eher zu Frustration. Der motivatorische Faktor bei agilem Vorgehen spielt auch noch eine Rolle. Mit einer kleinen Gruppe von Anwendern schnell mal eben was

	umzusetzen, passt eigentlich wunderbar zu Low Code/ No Code. Also da würde ich immer sehr stark dazu tendieren, das ist agil. PMI nennt das ja sogar hyper agil.
46	[0:27:00.3] <b>I: RS</b> Jetzt sind natürlich aber meistens ja auch mit der Vorgehensweise ja auch gewisse Hürden verbunden. Wie würden Sie die sehen, bei dem No Code/ Low Code-Einführungsprojekt?
47	[0:27:11.8] <b>B: EH</b> Hürden?
48	[0:27:14.2] <b>I: RS</b> Oder wo ein Unternehmen darauf achten sollte, wenn es diese Vorgehensweise jetzt nimmt? Oder wo es darüber stolpern könnte?
49	[0:27:19.4] <b>B: EH</b> Stolpern, also eine Hürde ist bei so einem Vorgehen natürlich, da man bei solchen Projekten häufig auch mit wechselnden Teilnehmern, je nachdem, mit welchem Fachbereich man gerade zu tun hat, arbeitet. Dass man so ein bisschen die Konstanz in der Arbeit verliert. Dass manche Leute punktuell mal was machen und dann bleibt das alles liegen. Dann habe ich so meine halbfertigen Applikationen, oder Applikationen, die gut genug sind, dass sie sie anwenden können und dann kümmert sich keiner mehr darum. Also das ist natürlich eine Sache, die bei agilem Vorgehen schnell mal passiert. Insbesondere nämlich, wenn das Team nicht konstant ist. Wenn ich ein festes Team habe, was die ganze Zeit arbeitet, dann ok, für eine Einführung kann man das auch machen, aber häufig ist es so, dass man ja so nach und nach in die Breite geht und dann immer wieder andere Leute hat. Und dann für eine Konstanz zu sorgen, das ist sicherlich eine Schwierigkeit bei der Einführung. Ich würde das jetzt auch nicht als Hürde bezeichnen, aber es ist eine Schwierigkeit dazu. Eine grundsätzliche Hürde ist aber natürlich, die Leute zu finden, die mitmachen. Die Zeit haben, die mitmachen, die Spaß daran haben, die Interesse daran haben. Ich weiß aber gar nicht, ob das nur bei einem agilen Projekt so wäre. Ich bin wahrscheinlich näher im Team und deswegen kommt gerade das da eher vor, weil bei einem klassischen Projekt bestimme ich das Team und dann müssen sich halt alle darum kümmern, so ungefähr. Könnte man eventuell als Hürde in der Vorgehensweise festhalten.
50	[0:28:49.1] <b>I: RS</b> Es passiert ja dabei viel aus Eigenverantwortung und so weiter.
51	[0:28:52.5] <b>B:EH</b> Ja
52	[0:28:54.3] <b>I: RS</b> Was mich besonders interessieren würde, jetzt wirklich von dem, wie sie so ein Projekt angehen, ein agiles. Also das klang jetzt schon durch, Scrum benutzen sie offensichtlich. Was gibt es noch für Methodiken, die Sie da gern aus dem agilen Baukasten, der ist ja riesig mittlerweile, was nehmen Sie da noch gerne?
53	[0:29:13.3] <b>B: EH</b> Was nehme ich denn da noch gerne? Also, Scrum ist so für den Prozess eine ganz gute Geschichte, so mit Sprints arbeiten, regelmäßig ein Review dazu machen, so mal darüber zu gucken. Ja gut, aber wo man guckt, was man typischerweise hat. Für so ein Einführungsprojekt, also ich muss immer so ein bisschen aufpassen, dass ich mich darauf konzentriere. Wir haben jetzt so ein Cloud- Einführungsprojekt bzw. ein No Code/ Low Code-Einführungsprojekt. Da brauche ich eigentlich nicht zwingend große Design-Thinking oder ähnliche Techniken oder Produktgestaltungstechniken spielen da für mich erst mal keine große Rolle. Also mir gehts tatsächlich um den Ablauf, um die Feedbackschleifen. Was man natürlich gut nutzen kann, ist eventuell so was wie Kanban, um so ein bisschen den Applikationsprozess oder den Entwicklungslebenszyklus sozusagen abzubilden in einem Board. Das kann man natürlich ganz gut nutzen. Wenn man so weit ist, aber das ist typischerweise erst in späteren Schritten bei solchen Projekten, dass man wirklich Applikationen für eine Zielgruppe entwickelt, dann könnte man natürlich auch mit so Design Thinking Personas oder ähnlichen Dingen, User Journeys, arbeiten und so was. Aber meistens sind die

Applikationen, die man im ersten Schritt in so einem Projekt macht, eigentlich so einfach gestrickt, dass ich mit solchen Tools noch nicht viel machen kann. Das wäre sozusagen in späteren Schritten, kämen dann wirklich so User Journey und Story Mapping und solche Sachen, könnten da eine Rolle spielen.

54

[0:30:51.9] **I: RS** Also beim klassischen agilen ist es ja eine der Stärken, dass es gut mit Änderungen umgeht. Bei dem, wie Sie das praktizieren, ist das anders? Oder wie gehen Sie mit Änderungswünschen an das Projekt um? Sie können das gerne, wenn Sie sagen, da gibt es einen Unterschied, betrachten von außerhalb des Projekts und auch von innerhalb des Projekts selber.

55

[0:31:23.5] **B: EH** Wie gehe ich mit Änderungswünschen um. Ja, so wie man klassischerweise im agilen damit umgeht. Man nimmt den Änderungswunsch auf, egal woher er kommt und bespricht mit den Leuten, die sozusagen für diese eigene Low Code Plattform-Einführung tatsächlich zuständig sind und den technischen Umsetzern vielleicht, ob das eine sinnvolle Änderung ist oder nicht. Also im Scrum würde man jetzt sagen, ich bespreche mit dem Product Owner, der für Einpriorisierung zuständig ist und ich nehme mir dann aber noch die Entwicklerin, also in meinem Fall dann die IT vielleicht dazu, um zu gucken, ist das technisch eine sinnvolle Geschichte. Vielleicht muss ich nochmal bei manchen Änderungswünschen nochmal den Datenschutz machen. Also ich mache dann letzten Endes so eine Bewertung der Änderung. Irgendwo muss ja einer mal darüber gucken und sagen, macht das Sinn, das so zu machen. Es ist eigentlich egal, ob sie von innen oder von außen kommt. Und dann wird sie halt in der Regel relativ zeitnah im Projekt umgesetzt, das heißt, typischerweise in die Sprintplanung mit aufgenommen.

56

[0:32:38.6] **I: RS** Und aber vorher ins Backlog genommen?

57

[0:32:41.7] **B: EH** Ja.

58

[0:32:42.8] **I: RS** Im Scrum haben Sie ja auch typischerweise relativ gut vorgegebene Rollen wie Scrummaster, den Product Owner, den Sie schon erwähnt haben, das Development Team. Sehen Sie die Rollen auch bei sich im Projekt? Oder sagen Sie bei irgendwelchen Rollen: "Es ist jetzt nicht zwingend erforderlich, dass die vorhanden ist." Oder wo Sie sagen eine Rolle, die ich gerade nicht genannt habe, die muss auf jeden Fall dabei sein, die will ich im Projekt haben.

59

[0:33:10.7] **B: EH** Lassen Sie mich kurz überlegen. Lassen Sie mich mal kurz über meine Projekte nachdenken. Also die Rollen, unter den Namen brauche ich sie nicht unbedingt, aber ich brauche natürlich einen, der tatsächlich sozusagen dahintersteht, dass No Code/Low Code eingeführt wird. Der das will. Also, so eine Art Product Owner, der sagt: "Ok, das ist für mich wirklich eine Applikation." Oder "Es ist für mich ein Ziel, das ich erreichen möchte. Ich habe da eine Vision, das irgendwie im Unternehmen zu nutzen." Im klassischen Projekt wäre das sowas wie ein Sponsor oder sowas, der dann entsprechend dahintersteht. Bei so einem agilen Projekt entspricht das der Rolle eines Product Owners, würde ich sagen. Worauf ich immer Wert lege ist, dass das Team selber gemischt ist. Jetzt nicht im Sinne von Rollen, aber dass ich Technik dabei habe, also IT dabei habe, dass ich Business dabei habe. Dass ich zumindest Zugriff auf sozusagen Datenbewertung, Datenschutz, also irgendwie juristisches Know-how irgendwo mit im Team habe. Wenn ich über solche Plattformen nachdenke und einen richtigen Scrum Master, meistens ist das eher so eine moderative Funktion, weil viele der Probleme, die dabei auftauchen, tatsächlich im Team selber gelöst werden

können. Also es gibt selten Dinge, wo wirklich von außen, also wo ein Scrum Master mal so ein impediment aus dem Weg räumen muss oder sowas, sondern meistens, wenn ich technische Probleme habe, habe ich die IT dabei, insofern kann ich die technischen Probleme lösen. Wenn ich eine strategische Entscheidung brauche, was wollen wir nutzen oder was wollen wir nicht, welche juristischen Aspekte sind zu betrachten, habe ich entweder meinen Produkt Owner oder ich habe meine Datenschutzexpertise sozusagen mit im Team. So dass ich tatsächlich so einen Moderator zwar mache, aber ich brauche nicht so diese, wie im klassischen Scrum, so einen, der so richtig auf die Prozesse achtet und so weiter. Da würde ich sagen, das ist gar nicht so zwingend erforderlich bei so einem Team. Weil es meistens eben nicht so ein ganz streng agiles Projekt ist, wo ein Team konstant über eine ganze Zeit zusammenarbeitet, sondern, wie gesagt, das sind häufig dann sehr versetzte Arbeiten. Man probiert einfach was aus mal und dann ist man wieder unterbrochen, dann probiert man wieder was aus. Und dann muss man das nicht ganz so strikt machen. Also das würde ich jetzt so sagen, wenn ich an meine Projekte denke.

60

[0:35:52.5] **I: RS** Wenn Sie an ihre Projekte jetzt so denken, welche wichtigen Stakeholder außerhalb des Projekts sehen Sie denn dann?

61

[0:36:02.0] **B: EH** Also, wenn wir das Projekt so anlegen, wie ich es gerade gesagt habe, dann habe ich die wichtigen Stakeholder tatsächlich im Team drin. Weil, Product Owner ist so ein bisschen aus der Geschäftsführung oder aus der höheren Unternehmensführungsebene, der sagt: "Ok, das ist für uns was." Manchmal ist das so IT-Leitung. Manchmal ist es vielleicht auch einer aus dem Business, der sagt: „Wir brauchen das.“ Ich habe meine Datenschutzexpertise mit drin. Ich habe die IT, die für die technische Umsetzung der Governance, zum Beispiel, zuständig ist, für die Wartung der Plattform, Pflege der Plattform, die muss mit drin sein. Und ich habe, wie gesagt, das Business, eventuell mit wechselnden Teammitgliedern, je nachdem, mit welchem Geschäftsbereich wir gerade arbeiten, mit drin. Und wenn es dann schwierig wird, muss vielleicht mal so eine Bereichsleitung oder so was mit reingeholt werden, als wichtige Stakeholder, weil, die müssen natürlich Ressourcen geben. Also die Leute, die da aus dem Business kommen, die müssen da Zeit dafür haben. Natürlich, die IT-Leitung sollte der IT auch so bisschen Ressourcen geben, aber inhaltlich habe ich eigentlich das, was ich brauche, im Team schon vorhanden. Es gibt eigentlich tatsächlich nicht mehr so viele Stakeholder drum herum, die ich noch brauche. Das sind dann punktuelle Ressourcenbedarfe. Ich überlege gerade, wo vielleicht dann noch Dinge sind, wenn Sie ein Unternehmen haben, das selber keine wirkliche IT oder keine große IT hat, dann brauche ich natürlich so was wie den Provider oder so was mit drin, also ist das natürlich ein wichtiger Stakeholder, der darin eine Rolle spielt. Das ist aber nur eine andere Version von IT, die ich dann im Projektteam habe. Und typischerweise lege ich die Projekte so an, dass ich eben erst mal intern arbeite und tatsächlich die No Code/ Low Code-Plattform für interne Applikationen nutze und nicht sofort für externe. Wenn wir dann das Projekt so weit haben, dass wir tatsächlich damit vielleicht externe Applikationen, die Kundenportale aufbauen oder ähnliches, die mit einfachen Dingen und dann wäre tatsächlich auch noch wichtig, auch mal Kundentester oder Marketing oder solche Sachen mit als Stakeholder zu haben. Aber das ist eigentlich ein späterer Schritt. Wir sind ja sozusagen bei der Überlegung der Einführung, sozusagen, und da reicht eigentlich das, was ich habe. Da würde ich sagen, brauche ich nicht viel mehr.

62

[0:38:50.5] **I: RS** Gibt es im Unternehmen dann Leute, Rollen, wo Sie sagen, da halte ich es zumindest für wichtig, die informiert zu halten, auch wenn ich jetzt nicht zwingend die irgendwie im Projekt involviere, aber dass sie zumindest wissen, was geht jetzt hier vor?

63

[0:39:06.9] **B: EH** Ja, natürlich. Also, informiert halten würde ich so viele Leute wie möglich. Natürlich sollte so was wie die Geschäftsführung informiert darüber sein, weil wie gesagt, das ist durchaus eine Änderung der Arbeitsweise im Unternehmen. Natürlich muss ich eventuell, das fällt mir gerade ein, das wäre vielleicht noch so ein Stakeholder, wenn man so an Personalabteilungen denkt. Ich gebe natürlich Leuten Verantwortungsbereiche, also manche Unternehmen haben ja so Personalrollenbeschreibungen, die sehr strikt sind und da könnte das auch ein Punkt sein. Das habe ich gerade völlig übersehen, aber das könnte noch eine Rolle spielen. Also das sind aber auf jeden Fall Leute, die ich informieren müsste. Natürlich die Leute aus dem Fachbereich, weil, die Leute, die ich im Team habe, machen ja die Applikation nicht für sich, sondern für den gesamten Fachbereich. Also bauen sich einen neuen Prozess, mit dem dann alle arbeiten. Na, da muss natürlich eine Information sein. Da würde ich schon eher in die Breite gehen, was die Information angeht. Und auch in der Einführung, wenn wir da schon kleine Tools entwickeln, die vielleicht präsentieren und mal zeigen und "Guck mal, so geht das und so einfach war das." Und solche Sachen, das würde ich schon so sagen.

64

[0:40:18.9] **I: RS** Und wenn wir jetzt in der Einführung erfolgreich waren. Wir haben eine Pilotenabteilung erfolgreich mit Power Apps und Power Automate versorgt. Wie würden Sie die Skalierung in den Rest des Unternehmens gestalten?

65

[0:40:33.7] **B: EH** Schrittweise.

66

[0:40:40.2] **I: RS** Heißt Abteilung für Abteilung für Abteilung, oder?

67

[0:40:45.9] **B: EH** Tatsächlich, also mal gucken, weitere Anforderungen sammeln. Weitere Anforderungen sammeln, weitere Möglichkeiten sammeln, wo gibt es Bedarf oder wo gibt es Möglichkeiten, das, was wir jetzt gelernt haben, weiter zu treiben, Effizienzgewinne herauszuholen und solche Dinge, wo gibt es Wünsche? Aber tatsächlich schrittweise. Ich würde das, glaube ich, tatsächlich immer so machen. Zieht sich zwar vielleicht ein bisschen. Also die Alternative wäre, man kann sich so zwei Vorgehensmodelle überlegen. Entweder eine schrittweise Einführung. Ich gehe dann immer weiter, bis ich das so einigermaßen flächendeckend habe, dann wird es wahrscheinlich auch so ein bisschen zum Selbstläufer. Also irgendwann können die Leute sich das auch selber beibringen, weil, so schwierig sind die Plattformen dann auch nicht. Ich habe zwei Leute, die haben das schonmal gemacht, die gucken sich mit einem Dritten das an und zeigen dem dann mal, wie das geht und dann kann der das auch, so ungefähr. Die Alternative wäre natürlich eine zweite Vorgehensweise, mit einem Piloten anzufangen und zu gucken, wenn der funktioniert, dann stellen wir es flächendeckend erst mal allen Abteilungen zur Verfügung. Das wird man aber nach meinem Dafürhalten tatsächlich nicht so betreuen können, dass die Leute wirklich anfangen damit zu arbeiten. Es sei denn, es gibt wirklich Schmerzen in den Abteilungen, wo die sagen: "Die können wir nur damit lösen." Dass die sozusagen freiwillig darin sind. Sonst wird das nach meiner Erfahrung eher so enden, wenn ich sofort in die Breite gehe: "Ja hier hat sich schon wieder mal einer ein neues Tool überlegt, aber eigentlich ich komme mit meinen Excel-Dateien eigentlich ganz gut klar. Warum soll ich mich jetzt damit beschäftigen."

68

[0:42:13.6] **I: RS** Wenn wir beim Thema Skalierung sind, kommen wir auch zu dem Thema Befähigung der Mitarbeiter. Wie gesagt, also, auch wenn man von Abteilung zu Abteilung geht oder es einmal direkt austretet, kommt ja trotzdem irgendwann die Frage: Wie befähige ich die Leute innerhalb des Unternehmens, auch abgesehen vom ursprünglichen



Einführungsteam, dann quasi, dass der Rest das auch kann. Die kommen ja nicht an einem Tag durch das Werkstor und sagen: "Boa geil Power Apps, klar. Alles logisch kann ich." Sondern die müssen ja befähigt werden. Wie würden Sie Befähigung machen?

69

[0:42:45.0] **B: EH** Also, typischerweise machen wir das auf zwei Ebenen, würde ich jetzt mal grob sagen. Es gibt so ein bisschen so mehr so Einführungsworkshops. Wir stellen vor, was im Piloten so gelaufen ist. Was wir für tolle Apps gebaut haben und so weiter vielleicht. Um erst mal Leute dafür zu begeistern, so ein bisschen Projektmarketing zu machen und wir planen und bauen natürlich so eine Art Schulungskonzept auf, in Form von begleitenden Schulungen. Also manchmal auch so, dass wir sagen: "Ok, ihr müsst die Schulung machen, dann kriegt ihr Zugriff auf die Plattform." Also dass man wirklich sagt: Ein bisschen Know-how muss da sein. Wie gesagt, man kann schon bei der Mächtigkeit der Plattform relativ schnell an Stellen kommen, wo man auch wirklich Unsinn machen kann. Deswegen kann man das eigentlich ganz gut so machen, dass man sagt: „Ok wir, machen eine Schulung und Bestandteil der Schulung ist eben, zum Beispiel, auch so ein bisschen die Unternehmensrichtlinien, Governance, wie gehen wir damit? Warum dürft ihr bestimmte Dinge nicht machen und warum sind die abgeschaltet? Auch wenn ihr die woanders mal gelesen habt.“ Dass man das versucht zu begründen. Ich bin da immer eigentlich ein Freund von einer relativ offenen Kommunikation. Das ist nicht in allen Unternehmen so, da wird einfach abgeschaltet, dann hofft man, das keiner merkt, dass es auch etwas anderen gibt. Würde ich aber tatsächlich nur eine Schulungsbegleitung machen. Wie das dann konkret aussieht, ob man das applikationsbasiert, übergreifend macht, sei mal dahingestellt. Das hängt dann sehr vom Einzelfall im Unternehmen ab. Also ohne Schulung würde ich es nicht machen.

70

[0:44:35.8] **I: RS** Und wenn Sie jetzt irgendwann aus dem Unternehmen weggehen, nachdem Sie das Projekt eingeführt haben, müssen ja trotzdem neue Leute befähigt werden. Es kommen ja immer wieder neue Mitarbeiter in so ein Unternehmen. Es sind neue, die sich dafür interessieren. Wie würde da die Befähigung aussehen, wenn Sie nicht mehr im Unternehmen sind?

71

[0:45:12.9] **B: EH** Man kann verschiedene Dinge machen. Also man kann tatsächlich intern, wie ich das eben schon mal kurz gesagt habe, so eine Art Multiplikatoren aufbauen. Also manchmal kommt man in der Plattform auch ohne explizite Schulung aus, also nach der Einführung, wenn genug Leute im Unternehmen schon damit gearbeitet haben, kann sich das Know-how tatsächlich auch schon so verbreiten, da muss man nicht unbedingt zwingend eine Schulung machen. Ich würde trotzdem immer noch Schulungen für komplexere oder für so einen Gesamtüberblick immer noch anbieten. Die kann man dann tatsächlich, also wenn das Unternehmen keine Personalabteilung hat oder keine eigenen Trainer hat, dann sollen sie sich halt Trainer kaufen. Ich würde das immer, das ist jetzt meinem Geschäft auch so ein bisschen mit geschuldet, ich würde es immer mit einem Live-Training machen, um einfach besser auf Fragen von Teilnehmern und so eingehen zu können. Es gibt natürlich ganz viele Video-Anleitungen und so was und Blogs und so weiter, wo man sich vieles zusammensuchen kann. Aber schöner ist natürlich, wenn man so was beispielhaft mit einem Live-Training und mit Fragestellungen und so weiter durchführen kann. Das sollte man einfach als kontinuierliches Schulungsangebot, also die meisten Unternehmen haben ja so was wie einen Schulungskatalog, da kann sowas dann tatsächlich durchaus drin sein. Und wie gesagt, das kann gebunden sein an die Zulassung auf die Plattform oder nicht. Dass man einfach sagt: „Ok, wenn du das machen willst, dann mach die Schulung und dann lassen wir dich zu und dann darfst du Power Automate benutzen.“

72

[0:46:55.2] **I: RS** Und jetzt noch mal auf das Projekt an sich gesehen. Das Projekt ist fertig bzw. Sie sind während des Projektes. Wie sichern Sie so die Qualität des Projektes, dass es nicht abgleitet, zum Beispiel, in dass das Team immer weniger zusammen macht und es irgendwann einfach nur im Sand verläuft. Wie sichern Sie die Qualität, dass alles oben bleibt?

73

[0:47:20.0] **B: EH** Gute Frage. Motivation, Motivation, Motivation. Wie mache ich das am besten. Ich überlege gerade. Eigentlich kann man das am besten durch klare Ziele. Man sollte sozusagen nicht in die Situation kommen, wo man nicht weiß, wo man jetzt eigentlich als nächstes hin will. Das ist so der Punkt. Also Motivation aufrechterhalten: Ok, das haben wir jetzt geschafft. Das ist schön. Das ist toll. Jetzt nehmen wir uns das nächste vor, damit das nicht auseinanderläuft. Sonst besteht tatsächlich die Gefahr, dass in so einem Projekt irgendwas erreicht wurde, und dann sagen die Leute: "Mein Teil ist getan. Ich gehe jetzt weg. Ich brauche jetzt nicht mehr." So ungefähr, das ist schon tatsächlich ein Punkt. Das heißt, man muss so ein bisschen dafür sorgen, dass man auch vielleicht auch aus dem geschäftlichen Umfeld einen Nachschub an Anforderungen kriegt. Was könnte man vielleicht machen? Wo könnte man hingehen? Es ist natürlich zu einem gewissen Grad naturgegeben, dass man nach einer ersten Einführungsphase in den meisten Abteilungen erst mal so ein Stadium hat, wo man sagt: „Ok, lass uns das jetzt erst mal eine Weile ausprobieren. Wir wollen jetzt nicht noch mehr Applikationen machen, sondern wir wollen einfach jetzt mal so ein bisschen ausprobieren.“ Das würde ich auch immer so zulassen, dass man sagt: "Ok, dann treffen wir uns in einem Monat wieder, wenn ihr das ausprobiert habt. Guckt mal, wo haben wir Optimierungspotenzial. Sind vielleicht neue Dinge aufgetaucht, die wir angehen können. Und dann machen wir wieder mal eine kleine Iteration, wo wir uns mit der nächsten Applikation beschäftigen können." Solche Sachen, oder: "Wir treffen uns regelmäßig mit der IT und gucken mal, was hat sich so an Aktualisierungen der Plattformen ergeben, die uns neue Dinge gibt." Oder manchmal gehe ich auch einfach ran an das Unternehmen: "Hier, das hat Microsoft gerade gemeldet" oder: "Das können wir jetzt auch machen. Wäre das nicht was für euch? Das passt doch ganz gut, da haben wir doch so lange dran rum gekämpft und das nie geschafft. Aber jetzt soll das angeblich gehen." So was passiert mir auch immer mal. Dass ich sage: "Ok, das geht jetzt viel einfacher. Das können wir jetzt so machen." Das ist aber immer so eine schrittweise Motivation tatsächlich. Also so ein Projekt ist tatsächlich naturgegeben zum Teil so, dass man zwischendurch so Phasen hat, wo tatsächlich wenig passiert. Man muss ja auch einfach eine Möglichkeit haben, das einfach mal laufen zu lassen und zu probieren. Funktioniert das, ist das für uns sinnvoll. Ich kann ja nicht immer wieder neu entwickeln, entwickeln und entwickeln oder so was.

74

[0:50:03.4] **I: RS** Und wie messen Sie den Erfolg am Ende vom Projekt? Also, es wird ja in den meisten Fällen irgendwann entweder ein Geschäftsführer oder ein Controller zu Ihnen kommen und sagen: "Jetzt sag mir mal, woher weiß ich denn, dass das hier erfolgreich war? Dass sich das Geld, das ich hier eingesteckt habe, gelohnt hat."

75

[0:50:19.3] **B: EH** Ja, das ist eine gute Frage. Das messe ich anhand der Ziele. Also deshalb frage ich am Anfang, was wollt ihr damit erreichen. Also wenn jetzt tatsächlich als Ziel herauskommt, wir wollen damit Geld sparen, oder wir wollen damit schneller werden, dann muss ich mir natürlich überlegen, wie viel schneller sind wir denn geworden. Ich könnte natürlich aber sagen: "Ok, wir haben eine ganze Reihe von Applikationen zum Beispiel umgesetzt." Könnte zum Beispiel Messziel sein. "Das können wir jetzt und das können wir jetzt und das können wir jetzt." Das ist natürlich in so einem Projekt in der Regel, gerade am Anfang, schwer zu messen, ob sich hier das Geld jetzt wirklich gelohnt hat. Weil, da müsste ich tatsächlich gucken, was wir durch die Automatisierung zum Beispiel an Zeiteinsparung haben

und solche Sachen haben. Also, wenn ich wirklich klare Zahlen liefern will, muss ich das tatsächlich auf der Ebene messen. Bis jetzt bin ich noch nie wirklich in die Bredouille gekommen, dass der Erfolg über diese Art und Weise gemessen wurde. Also, dass ich wirklich klare Zahlen als KPI's sozusagen liefern musste, für den Erfolg des Projekts, sondern es war tatsächlich der Know-how-Aufbau im Unternehmen und der Umgang mit der Plattform und den Nutzungsgrad, ist sozusagen der Erfolgsfaktor. Also wird es tatsächlich eingesetzt? Arbeiten meine Leute damit? Haben wir die motiviert, das zu machen? Und kriegen die damit ihre Prozesse in den Griff? Und haben wir unsere Risiken dabei auch im Griff? Das ist natürlich auch ein Punkt, was ich hier eingangs ja schon sagte, das muss natürlich auch noch der Grund sein. Das sind eigentlich die Faktoren, an denen ich in dem Projekt gemessen werde.

76

[0:51:59.0] **I: RS** Wenn Sie jetzt sagen, Sie gucken darauf, arbeiten die Leute damit? Es kann ja nun auch passieren, dass die 50 Apps anfangen, keine wird zu Ende entwickelt. Woran stellen Sie fest, dass die auch erfolgreich damit arbeiten? Also das die die Applikation anfangen, mit der und der Anforderung und das auch zu dem und dem Ziel bringen?

77

[0:52:20.9] **B: EH** In dem man guckt, ob die Applikation auch fertig geworden ist. Ob sie auch eingesetzt wird. Läuft der Prozess, wird der tatsächlich genutzt. Das hängt jetzt ein bisschen davon ab, in welche Richtung es geht. Aber man hat ja durchaus in den Plattformen typischerweise ein Reporting, das so ein bisschen zeigt, wie oft so was genutzt wird, wie viele Leute damit arbeiten, wie oft etwas geteilt wird und so weiter. Das kann man schon so ein bisschen sehen, ob da kontinuierlich damit gearbeitet wird. Bei so einer Prozessautomatisierung ist es ja häufig so, das sehen sie in der Oberfläche gar nicht. Da hat irgendwer im Hintergrund einen Prozess über irgendein Datenposetry laufen lassen und der wird immer automatisch getriggert. Die Anwender merken da nichts von. Die machen nichts mehr, aber der Prozess läuft. Ist das jetzt gut oder schlecht? Da kann ich nur gucken, wie häufig wird der angestoßen? Kann ich mal ins Reporting gucken und gucken; läuft das, treten Fragen dazu auf oder funktioniert das. Am Anfang hat man typischerweise schon den Effekt, dass Leute auf einen zu kommen und sagen: "Das funktioniert noch nicht so richtig, das müssen wir noch optimieren" oder: „Da ist noch ein Punkt, an den wir noch mal ranmüssen.“ Sehr häufig habe ich das so, dass die das nach und nach sozusagen dann auch selber machen, weil; das ist ja die Idee bei der Einführung einer Plattform; dass die Anwender dann auch selber da hingehen und dann kann ich eigentlich nur gucken gehen und mal in die Read Folds gucken, ob irgendwo verwaiste Anwendungen liegen, die nie wieder genutzt wurden. Wo einer mal was gebastelt hat, was nie zu Ende gegangen ist. Da können wir vielleicht auch mal hingehen und gucken. Deswegen, das ist immer der Grund, warum ich so ein bisschen immer davon abrate, sofort in die Breite zu gehen, weil dann verlieren sie den Überblick. Dann wissen sie nicht, was passiert, dann lieber zielgerichtet mit einzelnen Leuten, mit kleinen Teams, fachbereichsspezifisch zu arbeiten, weil dann ist erstens die Motivation besser da und man kriegt auch einen besseren Überblick, was wird tatsächlich eingesetzt, was ist wirklich sinnvoll.

78

[0:54:11.0] **I: RS** Dann sind wir quasi schon am Ende von unserem Interview. Meine allerletzte Frage ist, haben Sie noch Anmerkungen zum Interview, Best Practices, die sie gerne teilen möchten? Oder möchten Sie noch gerne irgendwas sagen, wo Sie sagen: "Das ist mir ganz besonders wichtig bei dem Thema, das wurde aber jetzt nicht gefragt." oder "Ich bin nicht dazu gekommen, das zu sagen."

79

[0:54:32.2] **B: EH** Naja gefragt, gefragt wurde es schon, aber es gibt tatsächlich einen Punkt, der mir immer wichtig ist bei dem Thema. Geht bewusst an das Thema ran. Also macht daraus ein Projekt. Geht bewusst ran. Bindet die IT mit ein, weil, du stellst deinen Anwendern eine Plattform zur Verfügung, die in der Regel sehr viel kann, aber auch zu sehr viel Chaos

führen kann. Wirklich, meistens wird die Mächtigkeit unterschätzt und die Komplexität der Plattform auch unterschätzt. Also entweder stelle ich es einfach zur Verfügung und die Anwender nutzen es nicht, weil sie es einfach nicht kennen. Oder die fangen dann an es zu nutzen, weil ich es zur Verfügung gestellt habe und wissen nicht so richtig was sie tun, da passieren komische Dinge. Also, wirklich kann ich nur raten, daraus wirklich ein Einführungsprojekt zu machen. Also es wirklich bewusst anzugehen. Klein, iterativ, wirklich auszuprobieren. Also das was wir besprochen haben, das kann ich nur wieder als Anregung mitgeben, weil gerade im Cloudumfeld ist es natürlich so, ich setze einfach Lizenzen: "Ach ja, Power Plattform ja, Power Automate schalte ich einfach mal an für alle Benutzer." Passiert da irgendwas, passiert da nichts, welche Konnektoren dürfen verwendet werden, welche dürfen nicht verwendet werden? Wer macht da was mit. Davon würde ich abraten, das bringt einfach nichts.

80

[0:55:54.2] **I: RS** Dann danke ich Ihnen ganz sehr für das Interview und ich würde jetzt die Aufnahmen beenden.

81

[0:56:00.4] **Ende**

## Einwilligungserklärung Interview

Ich erkläre hiermit mein Einverständnis zur Nutzung der personenbezogenen Daten, die im Rahmen des folgenden Gesprächs erhoben wurden:

Datum des Interviews	22.07.2022
Namen der zu interviewenden Person	Eckhard Hauenhem
Name der interviewenden Person	Rocco Schenk
Kurzbeschreibung des Forschungsprojekts	Einführung von No Code/ Low Code aus Sicht des Projektmanagements
Durchführende Hochschule	Hochschule Mittweida – University of Applied Science

Die Daten werden im Rahmen eines mündlichen Gesprächs erhoben, das mit einem Aufnahmegerät und Videoaufnahmegerät aufgezeichnet wurde. Zum Zwecke der Datenanalyse werden die mündlich erhobenen Daten verschriftlicht (Transkription), wobei die Daten nicht anonymisiert werden.

Vorhandene Kontaktdaten, der interviewten Person, werden aus Dokumentationsgründen in einem separaten Schriftstück lediglich den Gutachter\*innen der wissenschaftlichen Ausarbeitung zur Verfügung gestellt. Nach dem Abschluss des Projekts werden diese Daten gelöscht.

Der Speicherung der personenbezogenen Daten zu Dokumentationszwecken kann durch die interviewte Person jederzeit widersprochen werden. Die Teilnahme an dem Gespräch erfolgt freiwillig. Das Gespräch kann zu jedem Zeitpunkt abgebrochen werden. Das Einverständnis zur Aufzeichnung und Weiterverwendung der Daten kann jederzeit widerrufen werden.

Dr. Eckhard Hauenhem  
Vorname und Name in Druckbuchstaben

  
Unterschrift

Essen, 22.7.22  
Datum, Ort

## Anlage 3, Interview EB02

RS: Rocco Schenk

RR: Robin Rosengrün

1	[0:00:00.0] <b>Start (RR - Robin Rosengrün/ RS - Rocco Schenk)</b>
2	[0:00:08.1] <b>I: RS</b> Ich würde dieses Interview aufzeichnen, bist du damit einverstanden?
3	[0:00:14.0] <b>B: RR</b> Ja, damit bin ich einverstanden.
4	[0:00:16.9] <b>I: RS</b> Könntest du dich ganz kurz vorstellen, einfach für den Kontext der Arbeit, wer du bist, was du so gemacht hast, wie deine Qualifikationen sind, hier auch was zu sagen.
5	[0:00:29.9] <b>B: RR</b> Ja gern, mein Name ist Robin Rosengrün und ich bin Low Code Enabler bei der EnBW AG in Baden-Württemberg, in Stuttgart. Und meine aktuelle Position ist, da habe ich die Aufgabe, dass ich meinen Kollegen und auch der IT-Abteilung die Low Code-Plattformen, vor allem die Microsoft Power Platform, näherbringe, Mitarbeiter schule, dann zum Beispiel Umsetzbarkeit von Projekten bewerte und da einfach, also allgemein auf breiter Front, unterstütze. Vom Hintergrund her bin ich, komme ich nicht aus der IT, sondern bin Maschinenbauingenieur und habe schon verschiedene Positionen bei der EnBW gehabt, angefangen habe ich als Werkstudent und dann auch meine erste Stelle war in der Netzberechnung von Gasnetzen in und um Stuttgart und die zweite Position, das passt heute glaube ich sehr gut, war ich Projektleiter für Contractinganlagen bei uns, also zum Beispiel wir haben da ein BHKW und Kessel- und Kälteanlagen, die gehören uns, stehen beim Kunden und wir versorgen dann den Kunden mit Energie und haben dann einen langfristigen Liefervertrag einfach ausgemacht, über 10 - 15 Jahre. Und in beiden Positionen war ich auch immer dafür zuständig, so bisschen das Projektmanagement zu machen, Daten geordnet, irgendwo zu erfassen, abzuspeichern und gerade auf der zweiten Position habe ich dann so etwa vor 3 - 4 Jahren angefangen, das alles mit Low Code-Tools zu machen, also mit der Microsoft Power Platform, vor allem mit Power Apps, dann so ein Projektmanagement für uns auch umzusetzen. Für unsere 250 Projekte, die wir da betreut haben.
6	[0:02:27.9] <b>I: RS</b> Es geht ja hier in der Arbeit heute um Low Code/ No Code, könntest du mal kurz sagen, wie du das definieren würdest? Diese beiden Begriffe oder würdest du es als einen Begriff definieren?
7	[0:02:40.6] <b>B: RR</b> Ja, das ist eine sehr, sehr schwierige Definition, weil jeder eine eigene Vorstellung davon hat. Low Code/ No Code definiere ich darüber vor allem, dass im Frontendbereich wenig Code benötigt wird und dass man eher über grafische Tools sich Sachen zusammenstellt, also zum Beispiel die klassischen PowerApps/ Canvas Apps, mit denen ich ja sehr viel mache, wirkt ja er eher von der grafischen oder von der Frontend-Bearbeitung ja eher wie Powerpoint, als wie jetzt, dass ich ein Programm schreibe, also ich ziehe mir die Sachen zusammen und das verstehe ich unter Low Code/ No Code. An sich gibts meiner Meinung nach, eigentlich fast gar nicht oder ist ein irreführender Begriff, weil damit von vielen verbunden wird, dass ich dann keine IT-Kenntnisse haben müsste, das ist aber aus meiner Sicht komplett falsch, sondern ich brauche trotzdem fundierte Kenntnisse, wie ich eine Datenstruktur mir zum Beispiel erstelle oder so was und meistens, also ich habe jetzt noch keine App geschafft, die produktiv im Einsatz ist, wo ich wirklich keine Zeile Code für geschrieben habe. No Code von daher gibts für mich nicht und Low Code dann eher, dass das Interface schnell

	und einfach zusammengeklickt wird, ohne dass ich ein grafisches Interface habe und eben nicht in Texteditor-Form.
8	[0:04:28.6] <b>I: RS</b> Kennst du den Begriff Citizen Development?
9	[0:04:32.9] <b>B: RR</b> Das ist ja mein Wirkungsfeld quasi, ich bilde Citizen Developer aus, dass sie dann die Tools für sich nutzen können. Genau. Und das ist, was bei uns im Unternehmen auch hauptsächlich stattfindet. Also die Entwicklungen mit dem Low Code-Plattformen kommen aus den Bereichen raus von nicht IT'lern oder meistens nicht gelernten IT'lern oder auch ITler, die in den anderen Funktionen dort sitzen, aber nicht klassisch bei uns in der IT-Abteilung angesiedelt sind. Und die Durchdringung der IT mit den Low Code-Plattformen ist ein Thema, was wir jetzt auch noch angehen wollen, aber was bei uns noch nicht so stattfindet, das heißt, bei uns ist eigentlich das klassische Citizen Development, das sind die Leute, die Low Code nutzen.
10	[0:05:24.2] <b>I: RS</b> Also nur um das klarzustellen. Citizen Developer sind für dich Leute, die quasi eigentlich keine IT-Kenntnisse haben, die aber dazu befähigt werden, also mit solchen Plattformen umzugehen oder wie würdest du Citizen Developer mal mit 2 Sätzen definieren?
11	[0:05:41.2] <b>B: RR</b> Nicht keine IT-Erkenntnisse, sondern also, als Grunddefinition würde ich sagen, Leute, die nicht in der IT-Abteilung sitzen, sondern aus der Fachabteilung raus selber Anwendungen entwickeln.
12	[0:05:53.9] <b>I: RS</b> Ok, und welche Vorteile siehst du jetzt von den Low Code/ No Code-Plattformen direkt, wenn du jetzt darüber nachdenkst, was macht dich enthusiastisch darauf?
13	[0:06:02.2] <b>B: RR</b> Also vielleicht die, die Geschwindigkeit. Ich zeige immer mal so ein Bild bei uns, wie ein klassisches IT-Projekt aufgesetzt wird, hat man einen sehr hohen Initialaufwand, ohne dass eine Zeile Code umgesetzt ist. Also klassischerweise haben wir da einen Fachbereich mit der Anforderung, dann haben wir einen IT-Projektleiter, der das irgendwie noch mal in IT-Anforderungen übersetzt und haben dann Entwickler, intern oder extern, die dann das umsetzen. Und bis diese Struktur aufgebaut ist, die Entwicklungen gefunden sind, die Anforderungen eingegossen sind, dauert es, hat man erst einen gewissen Initialaufwand, egal jetzt, ob das Projekt agil stattfindet oder klassisch mit einem langen Lasten/ Pflichtenheft. Irgendwie man hat einen langen Initialaufwand und bei Low Code kann ich einfach direkt anfangen, meine Ideen in eine Anwendung zu übersetzen und hab sehr schnell eine gemeinsame Sicht darauf. Wir haben uns kurz vor der Aufnahme darüber unterhalten, dass jemand eine falsche Sicht auf dem Thema hatte, das finde ich bei der Softwareentwicklung auch immer ein ganz großes Thema. Dass der Fachbereich eine Vorstellung davon hat, was er will. Aber eine Sprachblockade quasi zum IT-Bereich besteht und dann in den Köpfen verschiedene Vorstellungen davon entstehen, was man eigentlich braucht und das fliegt dann erst auf, oder merkt man dann erst, wenn es dann den ersten Prototyp gibt. Und ja, dann hat man halt bei der Low Code-Entwicklung sehr schnell einen ersten Prototyp, den man dann immer weiter verfeinert und hat sehr schnell ein gemeinsames Bild davon, was eigentlich entwickelt wird und kann dann halt an vielen Stellen noch mal anpassen, was eigentlich der Bedarf ist.
14	[0:07:58.8] <b>I: RS</b> Und siehst du da auch Nachteile? Dann also, das waren ja jetzt die Vorteile. Siehst du auch negative Aspekte von No Code/ Low Code?
15	[0:08:04.8] <b>B: RR</b> Klar, also wenn es aus dem Citizen Development raus betrieben wird, dann ist natürlich die große Gefahr, dass wir Schatten-IT 2.0, nenne ich das immer, haben, also dass wir keine Governance aufgesetzt haben und das eigentlich, ja, sowas wie Excel Script ist, keiner weiß Bescheid, es gibt keine Doku zu gar nichts. Das ist ein großes Thema und natürlich, dadurch dass, also ich habe ja gesagt, das Citizen Development heißt nicht, dass die Leute keine IT-Kenntnis haben, aber oft ist es halt so, dass sie nicht von einem klassischen IT-Background kommen, sich Sachen selber beigebracht haben und dann halt

Sachen auch nicht optimal umgesetzt werden. Was sich dann in Performance äußern oder im Datenstrukturen, die alles andere als optimal sind, gerade das, was man dann merkt, wenn sich Projekte erweitern. Ja, das sehe ich als die Hauptnachteile, also einmal Governance Aspekt und zweitens schlechte, schlechte Programmierung und Design. Also Design im Sinne von Design, aber auch im Sinne von Softwarearchitektur.

- 16 [0:09:30.8] **I: RS** Wenn du das jetzt so gegeneinander mal aufwiegst, was würdest du sagen, was zum jetzigen Zeitpunkt eine No Code/ Low Code-Plattform, realistisch gesehen, erfüllen kann an Erwartung? In Unternehmen, was sind realistische Erwartungen, wir haben ja schon darüber geredet. Es gibt viele utopische Anforderungen, was sagst du, ist eine realistische Erwartung, die man wirklich heute schon an eine solche Plattform stellen kann?
- 17 [0:09:53.3] **B: RR** Sehr große, finde ich, also es hat einen sehr großen Einsatzbereich und viele Low Code-Plattformen, auch neben der Microsoft Power Plattform, sind wirklich so ausgereift, dass sie da ohne Bedenken zum Einsatz kommen können. Und decken weitreichend ab, einfach zwischen der, also auch für die persönliche Produktivität, aber ich setze das irgendwie immer an zwischen der persönlichen Excel-Datei und dem SAP ERP System. Dann, das sind die 2 Extreme, die es dann natürlich gibt und dazwischen ist ein Riesensbereich und man muss halt einfach definieren, in welchem Bereich genau das zum Einsatz kommen soll. Aber jetzt gerade, weil dein Thema ja auch das Projektmanagement ist, da Sachen, die jetzt bei uns nicht im ERP System abgebildet werden, wäre ich immer sofort dabei, mach das bitte, bitte, bevor dort anfängt, es über Excel-Dateien etc. zu machen, nutzt da ein Low Code-Tool, eine Low Code-Plattform dafür. Man hat immer ein besseres Ergebnis, als wenn man das über Kalkulationstabellen macht oder so was. Ja, sehe ich schon als extrem weiten Einsatzbereich auch, also entweder Organisation in einer Abteilung. Große Geschichte, die ich darüber hinaus auch noch sehe, ist Kommunikation zwischen 2 oder mehreren Abteilungen. Also es gibt ja ganz klassische Prozesse, die von Abteilungen A nach B nach C und dann manchmal nach D manchmal nach E laufen oder so was, die ganz klassisch über E-Mails abgebildet werden oder, was ich noch schlimmer finde, Word Dokumente, die ausgefüllt werden müssen oder so was und dann per E-Mail weitergeschickt werden. Also, das ist auch noch ein großer, großer Einsatzbereich. Und dann muss man nur wieder die Grenze ziehen, zu, bei uns wäre es jetzt klassisch Kundendatenverarbeitung plus Abrechnung oder so was. Das gehört dann, wenn ich 1000000 Kunden habe, auf jeden Fall nicht mehr in eine Low Code-Plattform, sondern da gibt es ja zu Recht dann halt große Anbieter, die da seit Jahrzehnten dran entwickeln und daneben ausgereifte Systeme haben, die das alles schon können.
- 18 [0:12:17.5] **I: RS** Du hast ja eben schon über das Projektmanagement gesprochen. Aus Sicht des Projektmanagements jetzt gesprochen, was denkst du, welche Voraussetzungen Unternehmen generell mitbringen müssen, bevor sie überhaupt eine Low Code/ No Code- Plattform integrieren können in ihr Unternehmen, was sind erforderliche, also was sind Voraussetzungen?
- 19 [0:12:38.8] **B: RR** Also, dass diese Schatten-IT 2.0 eben nicht passiert, muss man ein gewisses Level an Governance aufbauen. Das kann ja auch ein bisschen wachsen mit der Low Code- Plattform. Aber man sollte sich von Anfang an bewusst sein, dass man so was braucht und dass jemand dafür abgestellt ist, der sich damit beschäftigt von: Kritische Konnektoren blocken oder überwachen, Neuerungen in der der Plattform dann entsprechend den Policies umsetzen/ einbinden, also da braucht man einfach jemanden, sollte man jemanden haben, der das macht. Ist das Erste, und das Zweite ist, woran es immer scheitert, es muss ein Bewusstsein für Daten geben, finde ich. Also es muss Leute geben, die Datenstrukturen in Datenbanken denken können, sonst funktioniert es nicht, weil, damit kann ich dann eben auch den Vorteil realisieren, dass ich halt Sachen nicht in einem Excel Spreadsheet abbilde, sondern in einer relationalen Datenbank.



20	[0:13:46.0] <b>I: RS</b> Würdest du auch noch Aspekte wie Unternehmensgröße oder Mindset mit als Voraussetzung sehen oder sagst du, das ist eher optional?
21	[0:13:56.2] <b>B: RR</b> Also Unternehmensgröße sehe ich überhaupt gar keine Einschränkungen. Also vom Citizen Development spielt sich dann meistens eh alles entweder innerhalb von Abteilungen ab, die dann ja meistens irgendwie eine Größe vielleicht bis hundert Leute oder so was, da funktioniert das optimal, aber auch dann darüber hinaus. Aber wir machen jetzt, zum Beispiel auch, wir bilden Prozesse ab, die alle 16000 Mitarbeiter bei uns in den Kerngesellschaften umfassen, bilden wir auch teilweise in der Power Platform ab. Also da sehe ich von bis gar keine Probleme. Und Mind Set ist natürlich, also man sollte auf der agileren Seite sein, vom Mindset her gerade auch im Citizen Development, sonst stellt man die Leute vor eine unlösbaren Aufgabe, weil man Leute, die sich weder mit Frontend-Entwicklung, noch Backend-Entwicklung, noch normalen IT-Prozessen auskennen, dann also man darf nicht die Erwartung an die stellen, dass sie dann ähnlich arbeiten können, sondern sollte ein agileres Mindset und eine gute Fehlerkultur haben, dass man den Leuten dann auch erlaubt, da zu wachsen an den Aufgaben und erst kleinere Sachen umzusetzen und dann eben auch größere Sachen umzusetzen. Der Erfolgsfaktor ist für mich auch, dass man mit Miniprojekten anfängt. Leute, die keine Erfahrung damit haben, mit Miniprojekten anfangen und dann an ihren Aufgaben wachsen, das ist eben auch so vom Mindset her dann kommt, ah ja, Low Code-Plattform, es ist alles einfach und ich will jetzt aber ein Tool, was mir ein komplettes CRM-Management, komplettes CRM mit Projektmanagement und so was alles integriert und dann ist halt der Mensch, der das zum ersten Mal macht, überwältigt einfach mit der Aufgabe.
22	[0:16:01.4] <b>I: RS</b> Meine Frage wäre jetzt auch, wie Problemstellungen in Unternehmen sein können, oder auf welche Probleme können Unternehmen stoßen, die Low Code/ No Code einführen, würdest du sagen, das ist schon eine Problemstellung oder würdest du noch weitere nennen?
23	[0:16:18.7] <b>B: RR</b> Problemstellung. Na ja,
24	[0:16:25.0] <b>I: RS</b> Oder Risiken?
25	[0:16:28.1] <b>B: RR</b> Falsche Erwartungen ist ein Problem, ein Risiko, das einfach nicht, also es gibt ja klassischerweise den Anforderer dann an Problemen, ist jetzt vielleicht auch der Chef irgendwie. Der hat dann von der Low Code-Plattform gehört und verdonnert seinen Mitarbeiter, damit was zu machen und hat einfach falsche Erwartungen, was man damit machen kann und was auch ein Mitarbeiter, der neu in so einer Rolle ist oder so etwas zum ersten Mal macht, was er damit bewirken kann.
26	[0:17:01.9] <b>I: RS</b> Und wenn du jetzt so ein Projekt startest, du möchtest No Code/ Low Code einführen. Wie würdest du das Projektteam aufbauen, würdest du die in ihren Abteilungen belassen und einen Projektleiter, der außerhalb steht, oder wie würdest du das aufbauen? Wie würdest du die Hierarchie festlegen oder was wäre die beste Hierarchie deiner Meinung nach?
27	[0:17:21.2] <b>B: RR</b> Meinst du für die Einführung einer No Code/ Low Code-Plattform?
28	[0:17:27.3] <b>I: RS</b> Ja, wenn ein Unternehmen das machen möchte, was wäre dafür die beste Organisationsform, wie sich so ein Projektteam aufstellen sollte?
29	[0:17:33.4] <b>B: RR</b> Also, es muss mindestens eine Person mit Erfahrung reinkommen. Natürlich, die das Ganze ein bisschen steuert. Ob extern oder intern ist da egal, es muss von Anfang an eine Governance Struktur mit dabei sein, also Administration von der Plattform muss es dafür geben und muss auch eine eigene Rolle sein, die nicht unbedingt, also es muss kein Entwickler sein, sondern dieser Person muss dafür eine gewisse Zeit eingeräumt

werden. Das ist gerade am Anfang, bis alle Prozesse stehen, eine zeitraubendere Geschichte und es muss ein Enabling geben. Also muss Leute wie mich geben, die einfach mit Rat und Tat zur Seite stehen, weil es sonst ein sehr zäher Prozess auch wird. Das reicht dann von: Man muss den Leuten zeigen, was sie damit machen können, als man muss Demos zeigen, was damit geht, man muss die Leute befähigen, das heißt kurze Schulung, man muss einen kurzen Draht zu dem Mensch haben können. Also ganz klassisch, manche Leute beißen sich einen Tag lang an einer Formel die Zähne aus, die ich wahrscheinlich in einer Minute runtertippen könnte. Da hilft es, also ich habe da immer ein offenes Ohr für die Leute, die können sich direkt an mich wenden und ich versuche schnell zu helfen und durch die Rolle von einem Enabler haben wir auch eine Community etabliert, was ich hier auch als Erfolgsfaktor sehen würde. Also jetzt nicht hyper offizielle Struktur da, aber dass man einfach eine Community aus den Leuten hat, die sich miteinander vernetzen, austauschen können, gegenseitig helfen gerade bei Problemen/ Herausforderungen, die spezifisch das Unternehmen betreffen. Sei es jetzt, wie der Datenschutz gehandhabt wird im Unternehmen oder wie Lizenzbeschaffung aussieht im Unternehmen oder so was oder gesperrte Konnektoren oder sowas, die eben nicht allgemeingültig sind, wo man nicht in einem Internetforum nachgucken kann. Ja, das sind eigentlich die 2 Rollen, die es bei uns gibt. Einmal Governance Administration und auf der anderen Seite ein Enabling. Inklusive auch ein gewisser Support. Das sind die 2 wichtigsten Strukturen für mich.

- 30 [0:20:19.7] **I: RS** Du hast ja, wenn du so ein Projektteam aufbaust, diese Personen einmal im Projektteam, würdest du die in ihren originären Abteilungen belassen, damit die trotzdem den Kontakt behalten zu diesen Abteilungen, aus denen, die auch Wissen mitbringen oder würdest du die komplett aus ihrem Alltagsgeschäft rausnehmen und nur für das Projekt quasi mit einsetzen, also wie würdest du da vorgehen?
- 31 [0:20:44.6] **B: RR** Es ist wichtig, dass die, also es ist bei uns so, und es ist denke ich auch wichtig, dass die Administration aus einer IT-Einheit raus passiert.
- 32 [0:21:04.9] **I: RS** Meinst du damit jetzt den Projektleiter? Mit Administration, wen meinst du damit, wenn du jetzt von klassischen Projektmanagementrollen sprechen müsstest?
- 33 [0:21:15.4] **B: RR** Also der Teil Projektgovernance Administration von der Plattform. Ist für mich wichtig, dass es in der IT-Abteilung stattfindet, einfach um engen Kontakt zur IT zu haben, um Vertrauen zur IT zu schaffen und das nicht in Konkurrenz dazu aufzubauen, sondern aus der IT raus aufzubauen, um einfach Sachen vorzubeugen. Wie: Ihr macht da euer komplett eigenes Ding, das ist nur halb gar. Die Prozesse sind völlig an allen IT-Prozessen vorbei. Wie es jetzt bei uns ist, dass meine Rolle außerhalb der IT in einem extra Digitalisierungsteam angesiedelt ist. Was natürlich ganz wichtig ist, die Leute müssen Hand in Hand zusammenarbeiten. Ich finde es gut, dass ich nicht in der IT bin, einfach dadurch, dass ich, wenn ich das Citizen Development fördern will, einfach dass ich sagen kann: Das sage ich am Anfang von jeder Veranstaltung eigentlich, die ich mache, ich bin nicht in der IT, einer wie ihr, schreibe ich meistens darunter, weil ich eben auch aus dem Fachbereich komme, einfach, dass sie sehen, es muss niemand aus der IT sein, man kann da auch selber Erfolge generieren und von daher finde ich es in der Struktur gut, wie es ist. Bei uns passt das halt auch mit meinem persönlichen Hintergrund gut zusammen, würde ich jetzt aber nicht sagen, dass es anders nicht funktioniert, nur dass der Governance-Teil der Struktur, muss meiner Meinung nach in der IT angesiedelt sein.
- 34 [0:22:53.7] **I: RS** Angenommen du führst ein Projekt ein, wo du planst, No Code/ Low Code-Plattform in deinem Unternehmen einzuführen, was ja bei euch schon geschehen ist, wie wäre deine Vorgehensweise dabei? Also wäre das eher ein klassisches Vorgehen mit Phasen oder sagst du: Phasen sind ungünstig, nur agil zählt oder was dazwischen oder etwas ganz Neues?

- 35 [0:23:20.8] **B: RR** Also, da habe ich schon mit vielen Leuten darüber gesprochen, weil also unser Ansatz war. Ja, vielleicht, kannst du Transkript kommt in den Anhang? Die Einführung ist, wir waren relativ früh mit der Microsoft 365-Einführung, da ist die Power-Plattform dabei und wir haben uns im ersten Schritt zu wenig Gedanken darüber gemacht, da war ich auch noch nicht in der Struktur und haben da eher noch einen Bereich, der eigentlich nichts mit der Power Plattform zu tun hatte, wurde das mit aufs Auge gedrückt. Die haben das gut gemacht, aber da kam keine geordnete Einführung. Leute wurden nicht aufgeklärt, was sie damit machen können und dürfen. Und es ging bei uns eher so ein bisschen, naja, es war dann auf einmal da und die Leute konnten es nutzen. Was ich gar nicht so schlimm fand, also ich bin eher der Typ, der sagt, ich stehe auch eher auf der Seite der Citizen Developer, also dass man es offen verwenden kann, dass man das aktiv den Leuten anbietet, schmackhaft macht. Dass man da nicht einen Riegel vorschiebt wie: „Du musst aber erst Kurs XY gemacht haben, bevor du darauf entwickeln darfst.“ Es gibt Leute, die lernen so, es gibt Leute, die lernen so, manche wollen gerne einen internen Kurs machen. Manche gehen über YouTube Videos, also so habe ich es auch gelernt. So biete ich es jetzt auch mittlerweile selber an, also es gibt einfach Leute, die schaffen es selber, sich da so schnell einzuarbeiten und auf einen gewissen Stand zu kommen, denen muss man dann nur noch ein bisschen die Leitplanken zeigen und die können dann selber damit arbeiten, andere brauchen halt ein bisschen mehr Unterstützung, müssen da ein bisschen mehr hingeführt werden. Also von daher denke ich, ist es wichtig, dass es wenig, also dass es keine Restriktionen gibt, das Ding zu nutzen an sich, natürlich Governance Restriktionen, was man machen darf, was man nicht machen darf, aber dass es erst mal zügig für alle verfügbar ist und dann eher über so eine Experimentierphase geht, wo die Leute das möchte dann einfach selber ausprobieren können.
- 36 [0:25:40.0] **I: RS** Worum es mir in der Frage mehr geht, ist, wie du beim Vorgehen bist, wenn du das einführst. Also was für eine Art von Projektmanagementmodell würdest du wählen? Würdest du eher so sagen: Ok, wir gehen in Schritt für Schritt 123 Phase, wo wir anfangen mit Planung, dann in die Teilplanung, dann in eine Realisierung oder sagst du: Nein, wir machen den agilen Ansatz über, zum Beispiel, weißt du, was der, was agiles Projektmanagement ist, mit Daily Scrum und so was zum Beispiel?
- 37 [0:26:12.3] **B: RR** Ja.
- 38 [0:26:13.9] **I: RS** Also in die Richtung hat die Frage mehr gezielt.
- 39 [0:26:17.3] **B: RR** Also, da ich es auch ein Projekt sehe, was immer am Laufen ist, hat halt keinen klaren Abschluss. Also, eine Low Code-Plattform an sich, ich sag nicht irgendwann, die ist eingeführt also, sondern es gibt dann immer neue Herausforderungen. Es gibt neue Funktionalitäten, die man vielleicht einführen/ schmackhaft machen will. Man hat also gerade speziell auch in der Microsoft Power Plattform, aber ich glaube auch in allen Low Code-Plattformen, hat man sehr hohe Entwicklungs-, also, die Plattform selber hat noch eine hohe Veränderungsgeschwindigkeit, wie die sich verändert. Von daher, denke ich, kommt man um einen agilen Ansatz da gar nicht drumherum, weil, Microsoft schmeißt dann einfach noch mal irgendwo einen Stock in die Speichen irgendwas ändert sich und dann musst du schnell darauf reagieren, also da kann ich nicht klassisch in 3 Phasen das durchziehen, denke ich.
- 40 [0:27:12.7] **I: RS** Würdest du vielleicht Phasen hinzufügen, als Planungssicherheit? Oder sagst du: Nein, der agile Ansatz erfüllt das voll und ganz.
- 41 [0:27:21.2] **B: RR** Ist natürlich bei dieser initialen Einführung, da müsste man schon mal in den Phasen arbeiten, denke ich, eine Governance aufbauen, Enabling Konzepte aufbauen, aber das würde ich auch nicht, also da würde ich bitte kein Projekt über 1 Jahr draus machen, sondern würde das dann eben...

- 42 [0:27:46.4] **I: RS** Also, quasi große Anteile agil, kleiner Anteil Phasen. Da hast du ja jetzt schon deine Vorteile quasi direkt gesagt, warum du den agilen Ansatz bevorzugt? Siehst du aber auch da Nachteile gegenüber vielleicht den Phasen, wo du sagst: „HMMM“ Darauf muss man ein Auge haben, wenn man mit dem Ansatz vorgeht?
- 43 [0:28:09.2] **B: RR** Ja, wenn man halt schnell durchstarten will, werden manche Governance Themen zu kurz kommen, sage ich einfach mal so. Zum Beispiel haben wir noch nicht die perfekte Lösung, wie wir das in unseren normalen Softwareentwicklungsprozess, wie die Abstimmung mit Datenschutz, Cybersecurity, Betriebsrat zu Themen ist. Also, da haben wir jetzt noch keinen zu hundert Prozent sauberen Prozess, der vollautomatisch im Hintergrund abläuft. Da wird es, glaube ich, immer dann Übergangsphasen geben, wo man dann noch offene Fragen hat und dann mit der Hand am Arm arbeitet. Zum Beispiel hat es ein bisschen gedauert, bis unsere, wenn ich Premium Features nutzen will in der Power Plattform, bis ich mir dann eine Lizenz bestellen kann, ist jetzt mittlerweile in unserem normalen Bestellsystem, wie ich mir auch einen Monitor und Laptop bestelle, da kann ich mir auch die Lizenz bestellen. Aber das war natürlich nicht von Minute eins da und ja, solche Strukturen wachsen dann halt mit der Plattform, statt dass sie von Anfang an voll perfekt abgebildet sind. Ist einfach ein Trade Off, den man dann hinnehmen muss, dafür dass man schneller unterwegs ist.
- 44 [0:29:32.2] **I: RS** Und du hast es schon gesagt, was du jetzt aus dem klassischen Management nehmen würdest, so ein bisschen die Phasen von Anfang für die Governance System. Was siehst du als die besten Methoden aus dem agilen Management an, wenn bei der Umsetzung, also, was hat sich aus deiner Erfahrung raus sehr gut bewährt, wenn man so ein Projekt durchführt? In den agilen Anteilen jetzt.
- 45 [0:29:53.0] **B: RR** Ja, wir arbeiten weder fest nach dem agilen, einfach, weil das Projekt noch so klein ist, also wir sind im Wesentlichen 3 - 4 Leute, die sich darum kümmern ohne festen Scrum Master oder so was dahinter und dann auch in den verschiedenen Abteilungen und ein festes Projekt klingen zu sein, das vorneweg als Disclaimer. Aber, wir haben unsere Weekly's, wo wir die Themen besprechen, dann bisschen langfristige Planung, wo wir hinwollen, wo wir zusammenarbeiten, aber nutzen da jetzt auch keine festen Tools aus dem agilen Baukasten.
- 46 [0:30:40.6] **I: RS** Also auch nicht so etwas wie Kanban, oder?
- 47 [0:30:43.5] **B: RR** Nee nicht mal das.
- 48 [0:30:46.2] **I: RS** Mich würde jetzt aber jetzt natürlich noch ganz besonders interessieren, wie ihr mit Änderungswünschen umgeht, die jetzt an das Projektteam gestellt werden oder die aus dem Projektteam selber kommen. Also gerade, wenn du sagst, ihr habt Weekly's, besprecht ihr die dann und setzt sie um oder, also, wie wird generell mit den Änderungswünschen aus dem Projekt umgegangen und an das Projekt von außerhalb?
- 49 [0:31:10.3] **B: RR** Die Änderungswünsche sind immer, also, das ist meistens nichts, was in 1 Woche umzusetzen ist, ist das Thema, weil wir uns da tief in IT-Strukturen befinden, jetzt zum Beispiel auf unsere Bestellsysteme so eine Lizenz drauf zu bekommen. Das ist wieder ein größeres Projekt. Das läuft dann über Wochen, viel Abstimmungsbedarf mit anderen Abteilungen, oder das andere Beispiel, was ich genannt hatte, auch mit Datenschutz. Alles gilt zusammenzuarbeiten, das ist auch nichts, was in 1 Woche geht. Das dauert und so, ja, lässt sich das auch schlecht in Sprints gießen. Wenn wir aber, ja, wenn wir dann tatsächlich auch Sachen entwickeln, wir entwickeln auch Sachen auf der Low Code-Plattform, für das Low Code-Plattform-Management oder so bietet auch Microsoft da schon Tools an, dann versuchen wir da, bis es mal initial implementiert ist, schon auch in einer Sprint-Logik so ein bisschen zu arbeiten, also definieren von Arbeitspaketen und dann 2-Wochen-Rhythmus oder so was, da dann Fortschritte zu sehen, wieder neu releasen, erste Features zu releasen, aber

machen wir nicht, brauchen wir nicht durchgehend, dafür ist unsere Arbeit auch zu unterschiedlich, also dass wir in dem Team auch zu unterschiedliche Aufgaben haben, von Governance hat, sage ich jetzt einfach mal auch sehr viele. Die haben ein Postfach und arbeiten dann quasi Tickets ab von Sachen, also ein großer Teil, den die da haben, wo man jetzt auch keine agile Logik für braucht. Ja, und ich plane einfach viele, viele Schulungen oder so was mit meinen Kollegen. Wo es auch wieder sehr viel Tagesgeschäft einfach ist. Mittlerweile sind wir jetzt in der Phase angekommen.

- 50 [0:33:16.1] **I: RS** Eine Frage, die ich mir jetzt gerade noch stelle. Du hast ja vorhin gesagt, ihr habt nicht so einen festen Scrum Master, welche wichtigen Rollen habt ihr? Also, vorhin hast du gesagt, das lief nicht von Anfang an gut, wenn du jetzt, sagen wir mal, du könntest zurückspringen und sagen, mit all dem Wissen, was du jetzt hast, was wären Rollen oder Kompetenzen gewesen im Projekt, wo du gesagt hättest: Das hätte ich gerne gehabt. Das wäre schön, wenn das da gewesen wäre, den hätte ich gerne gehabt und diese Kompetenz. Also, wenn du jetzt wirklich so eine Idealvorstellung formulieren könntest aus der Erfahrung heraus.
- 51 [0:33:47.9] **B: RR** Also schon einen Product Owner, Low Code-Plattform hätte ich gerne, hatten wir nicht, hätte ich gerne, der über alle Themen so ein bisschen darüber guckt und danach guckt, dass die Fäden alle richtig zusammen laufen.
- 52 [0:34:06.8] **I: RS** Und wie macht ihr das dann mit dieser Scrum-Master-Rolle. Du hast gesagt, ihr habt keinen festen Scrum-Master. Wechselt der dann bei euch? Bloß damit ich das auch kurz verstehe.
- 53 [0:34:19.8] **B: RR** Mit der Hand am Arm, selber organisiert.
- 54 [0:34:24.9] **I: RS** Was mich ja auch besonders interessiert, ist, wenn du das im Projekt dann mal eingeführt hast, ist es ja meistens relativ klein. Wie skalierst du das nach oben, in die große Organisation rein? Vor allem bei EnBW, ist ja auch eine große Firma, das stelle ich mir besonders schwierig vor. Wie skalierst du das?
- 55 [0:34:50.4] **B: RR** Das ist eine gute Frage, große Herausforderung. Wir versuchen nicht durch Manpower zu skalieren, sondern durch Automatisierung. Also, wie das Beispiel, ich muss nicht von Hand Lizenzen vergeben, sondern richte dafür feste Prozesse ein. Also für die Governancethemen richte ich feste Prozesse ein, die mir erlauben, das effizienter zu machen, dass ich trotzdem mit, ich glaube, wir haben gerade 150 % Governance, also 1,5 Personen die, die Governance machen, dass die das sowohl für 100 Menschen geschafft haben, aber durch weitere Professionalisierung auch für 2000 aktive Maker schaffen. Und genauso im Enabling, dass ich mir Strukturen suche, die ich anfangs für einen geringen Personenanteil, auch mit einem hohen manuellen Anteil, dann eben gemacht hab, aber dass wir auch Strukturen haben, die sich sofort an alle richten. Ein Beispiel dafür ist bei uns zum Beispiel, dass wir nicht mehr monatlich feste Kurse für PowerApps/ PowerAutomate geben, sondern dass ich einen Videoselfservice-Kurs an irgendeinem Punkt gemacht habe, den die Leute dann selber durchklicken können, das ich dann nicht sagen muss, „Melde dich zum nächsten Kurs an“, sondern „Hier ist der Selfservice Kurs, kannst starten“ und ja, das nimmt natürlich Arbeit ab und ja, dem Kurs ist es egal, ob er 1-mal im Monat oder 1000-mal im Monat geklickt wird. Dass man halt als manueller Trainer oder als persönlicher Trainer nicht mehr schaffen kann. So versuchen wir unsere Strukturen weiter zu professionalisieren. Natürlich schon personell ein Stück weit zu wachsen, aber nicht annähernd mit der Anzahl der Nutzer.
- 56 [0:36:52.2] **I: RS** Weil du es gerade erwähnt hast. Zum Beispiel Nutzer befähigen über diesen Kurs. Würdest du Mitarbeiterbefähigung nur über solche Online-Kurse machen und wenn ja, wie gehst du da mit Problemen um, auf die Nutzer stoßen? Also, wenn jemand eine Frage

hat, weil er aus dem Video etwas nicht richtig verstanden hat. Also Befähigung von Mitarbeitern als Oberthema. Wie gehst du damit um?

- 57 [0:37:17.2] **B: RR** Ja also ich gehe grundsätzlich mit der Einstellung ran Low Code-Plattformen: nicht jeder muss die Low Code-Plattform nutzen und es ist nicht mein Ziel, dass jeder Mitarbeiter anfängt, die Low Code-Plattform als Developer zu nutzen. Als Anwender muss er sie dann natürlich nutzen können, aber das liegt ja dann an der Anwendung. Aber als Developer brauchst du schon einen speziellen Typ Mensch, also ich suche mir die Leute, die bisher auch Lust hatten, in Excel ihre komplexeren Eigenkreationen zu machen und vielleicht da auch ein bisschen Scripting keine Angst hatten bisher, das sind eigentlich die Leute, die Spaß und Freude daran haben, für die es eine schöne Abwechslung im Arbeitsalltag ist und auch einmal etwas anderes zu machen, so ein bisschen zu entwickeln. Die versuche ich aktiv zu finden, anzusprechen und auf diese Low Code-Plattform zu kriegen und das ist normalerweise auch der Typ, der mit solchen Kursen klarkommt. Wenn die Leute nicht mit den Kursen klarkommen, dann ist das meistens auch so, dass für die Leute wahrscheinlich, die finden keine Freude in der Low Code-Entwicklung und dann wird es für alle Seiten zäh. Also sowohl für denjenigen, der das lernen möchte, als auch für denjenigen, der irgendwie erwartet, dass eine fertige App rauskommt und dann ist das nicht effizient und macht eigentlich alle nur unglücklich. Daher sehe ich, dass es schon, dass das die richtigen Tools sind und ich versuche die Leute möglichst früh selber ins googeln zu bringen. Sage ich jetzt mal ganz blöd, also entwickeln heißt halt in vielen Fällen googeln, sich durch Foren, Blogs etc. besten Lösung kopieren und adaptieren und Leute, die das nicht können, werden nicht glücklich werden mit dieser Art der Aufgabe. So merken die Leute selber, also ich empfehle auch Leuten, die das nicht wissen, ob sie das mögen, diesen Kurs zu machen, danach merken sie vielleicht irgendwie, das hat mir jetzt wenig Spaß gemacht, dann haben die da 3 Stunden investiert und können danach sagen: „Ich weiß, dass es nichts für mich ist“, das ist auch eine sehr gute Erkenntnis, denke ich, aber daher funktionieren solche Sachen sehr gut. Und was auch, hatte ich jetzt vorher nicht erwähnt, dass auch ein Mittel der Wahl bei mir ist, ist einfach, ich mache viele Veranstaltungen, die einfach offen für jeden sind, muss man dann von der Moderation halt ein bisschen anpassen, ob da jetzt 100 Menschen dabei sind oder 10 Menschen, wie viel direkten Kontakt man hat und zusätzlich versuche ich alles, oder ziemlich viele Veranstaltungen, auch einfach aufzunehmen und wieder zur Verfügung zu stellen, dass die Leute diese dann asynchron gucken können. Ist für mich 5 Minuten Mehraufwand. Für den ersten, der es anschaut, hat es sich eigentlich schon gelohnt, den Aufwand dann zu betreiben. Und natürlich auch für Leute, die noch mal, also statt nachzufragen, weil wenn irgendwas war, können sie einfach noch mal nachschauen, in die Aufzeichnung, das ist etwas, was mir wieder Arbeit abnimmt. Ja, das so grob mein Gedankengang zu Skalierung, Enabling, wie macht man das am besten. Aber ab irgendeinem Punkt muss man dann schon ein bisschen wachsen, weil, es gibt dann halt komplexe Anwendungsfälle, wo es einfach gut ist, noch mal Qualitätsprüfung zu haben, mal persönlich unterstützen zu können, einfach bei komplexeren Themen, einfach mal eine Abteilung zu beraten oder so was, was geht, was nicht geht. Bevor die Leute anfangen und losrennen und versuchen was zu entwickeln, versuche ich immer eine Empfehlung abzugeben, ja funktioniert sehr gut, sehr schnell damit. Oder ich kann auch meistens sagen: Ja, das ist eine Problematik, die würde ich dir auf keinen Fall als erstes Projekt zu machen empfehlen und wenn, dann wird es auch länger dauern, hört sich vielleicht im ersten Moment einfach an, aber mit der Erfahrung sieht man dann natürlich direkt die Probleme und da brauchen wir natürlich schon ein Stück weit eine persönliche Betreuung, spart dann aber hinten raus viel, viel Zeit und Frust.
- 58 [0:41:40.2] **I: RS** Wenn du sagst, es ist im Unternehmen. Was wünschst du dir außerhalb des Projektteams jetzt für, sagen wir mal, wichtige Rollen oder was bräuchtest du da vom Unternehmen außerhalb des Projektteams? Um das alles realisieren zu können, so was ist da, was ist da wichtig? Wo du sagst: „Das weiß ich schon aus Erfahrung, dass die

Abteilungsleiter zum Beispiel alle mitmachen“, oder also wer sind die wichtigen Stakeholder für das Thema?

- 59 [0:42:09.1] **B: RR** Ich habe das Thema endlich geschafft, in unserer IT-Strategie zu platzieren, auch dass da namentlich genannt wird. Das macht es natürlich nicht zum Selbstläufer, sondern da muss man noch viel Arbeit reinstecken. Also, was ich mir noch wünsche ist, dass das Thema wirklich auch in der IT ankommt, da als valider Lösungsweg gesehen wird. Auch vielleicht hin, das hört sich jetzt noch nach Microsoft Marketing -Sprech an, aber ich finde es auch wirklich eine gute Vision, also die, ich spreche immer von Fusion Teams. Dass Teile von der Entwicklung vielleicht von einem professionellen ITler umgesetzt werden, andere Teile darauf aufbauend, dann vielleicht von einem Citizen Developer umgesetzt werden. Also das ist meine Vision, wo wir hinwollen, und dazu braucht man auch einfach eine Organisationseinheit, die entwickeln kann und dafür Kapazitäten hat, wie auch immer das ausgestaltet ist. Interne Entwickler würde ich mir wünschen. Auf externe Entwickler läuft es meistens raus, aber als feste Einheit sage ich mal, die auch Sachen umsetzen kann.
- 60 [0:43:32.0] **I: RS** Also quasi schon Vorgesetzte im Unternehmen siehst du als wichtig an und aber auch in der eigenen Struktur, dass da Verständnis da ist, von den Leuten, die jetzt nicht Teil des Projektteams sind. Verstehe ich dich richtig?
- 61 [0:43:47.1] **B: RR** Also ja, das war jetzt gerade noch mehr so eine Ergänzung zu, wie sieht das Projektteam aus. Also ob das jetzt Teil von diesem Projektteam wäre. Du hast eher noch so auf die Organisation ein bisschen mehr abgezielt.
- 62 [0:44:02.3] **I: RS** Naja, wer so wichtig ist, heißt es ja. Oft scheitern Projekte daran, dass zum Beispiel, die Vorgesetzten nicht dahinterstehen oder das von anderen Abteilungen, die, von denen man Ressourcen braucht, dass die diese einfach nicht zur Verfügung stellen, weil die das nicht geplant haben und auch nicht möchten. Darum geht es mir. Was würdest du als wichtig sehen, wenn du außerhalb des Projektes jetzt mal gucken würdest? Wer ist wichtig, um den Erfolg des Projektes auch zu sichern?
- 63 [0:44:26.1] **B: RR** Also, dass allgemein ein besseres Verständnis für Daten herrscht in dem Unternehmen, also das wird ja propagiert, seit Jahren schon, in jedem Unternehmen wahrscheinlich, Daten sind das neue Gold. Ich will jetzt gar nicht so sehr in die Richtung blasen und weiß auch nicht, ob das für unser Unternehmen so sehr stimmt, wie das propagiert wird. Aber, dass Prozesse mehr auch von den Daten her gedacht werden und dass da einfach ein Verständnis für herrscht. Also, dass in jeder Abteilung irgendwo das Verständnis für herrscht, was für Daten produzieren wir und wie können wir diese verwenden, welche davon können wir strukturieren, welche lohnt es sich zu strukturieren, um, zum Beispiel, herrscht gar kein Verständnis dafür, dass man mit E-Mail-Strömen aufhören soll. Und vielleicht ist viel besser auf ein Ticketsystem, bei welchem Thema auch immer, umzustellen, weil einfach gar nicht gesehen wird, dass eigentlich immer ein Datenstrom, mit jeder E-Mail enthält die gleichen 10 Informationen, die sich super machen würden in einer in der Datenbank strukturiert, in den Prozess dann auch wieder. Schick ich halt die E-Mail an den nächsten, der muss das bearbeiten, aber die Person, die das geschickt hat, weiß dann gar nicht mehr, wie der Status ist, also das ist so ein Klassiker, den wir an ganz vielen Stellen haben. Weil einfach niemand das Verständnis dafür hat, dass da eigentlich ein Datenstrom abläuft. Dass es eigentlich nichts anderes wäre als der Zeile in der Datenbank und den Prozess für alle Seiten einfach transparenter macht und erst, wenn ich dieses Verständnis habe, dann habe ich ja auch die Möglichkeit, die Low Code Tools einzusetzen. Also das ist so ganz oft auch was, was mir entgegengebracht wird, ich würde das sehr gern verwenden, aber ich weiß nicht wofür. Wenn ich mit Leuten, also egal jeder Abteilung, nach 5 Minuten Gespräch, habe ich in der Regel 10 Ideen, was man damit machen kann, ja und andere Leute aus den Abteilungen rauskommen, die selber auf keine dieser Ideen, sagen dann „Ah ja, das stimmt ja, alles gut“ aber ja, das kommt halt mit einem gewissen Verständnis für Daten/ Datenstrukturen, kommt das dann

halt erst mal so was sieht. Das ist für mich auch ein Erfolgsfaktor, dass das funktionieren kann. Dann halt allgemein, das muss von der Führungskraft bis zum letzten Mitarbeiter, muss ich das einfach wahrscheinlich in jedem Unternehmen dieser Welt, außer vielleicht bei Twitter und Facebook, aber muss da einfach die Awareness für Daten geschafft werden...**[Tonabbruch]**... Also Awareness für Daten erhöhen, nicht unbedingt nur raus aus dem Thema, dass ich mit diesen Daten Geld machen kann in irgendeiner Form, sondern einfach, dass ich meine Prozesse so stark optimieren kann dadurch, dass ich, dass ich einfach die besser strukturiere und in den Datenbanken überführe.

- 64 [0:48:04.5] **I: RS** Und wenn du so ein Projekt jetzt hast? Was würdest du sagen, sind Qualitätskriterien oder woran würdest du Erfolg von dem Projekt messen? Also woran würdest du sagen, wenn ich mir das angucke, weiß ich, das war ein gutes Projekt oder ist ungünstig gelaufen, also was die Qualitätskriterien/ Erfolgskriterien?
- 65 [0:48:24.7] **B: RR** Jetzt wieder Projekte im Sinne von Einführung einer Low Code-Plattform?
- 66 [0:48:28.8] **I: RS** Ja.
- 67 [0:48:30.0] **B: RR** Also was wir monitoren auf hoher Flugebene ist natürlich die Anzahl der Personen, die die Plattform nutzen. Anzahl der aktiv genutzten Anwendungen lässt sich durch Governance ja auch einfach ermitteln. Also das sind Daten, die erfasst werden von der Power Platform jetzt im Speziellen, aber wahrscheinlich auch von allen anderen. Aber, dann auch tatsächlich, dann gehts dann eher in die qualitativen Ziele, dass wirklich auch sich das weiterentwickelt, dass Anwendungen mit immer größerem Impact auch auf die Plattform gehoben werden. Also dass man auch wirklich dann irgendwann so sicher ist, dass man anfängt, geschäftskritischere Anwendungen darauf zu heben. Genau das lässt sich natürlich nur schwer, dann irgendwie messen. Plus, dass man halt merkt, dass man nicht die Schatten IT 2.0 aufgebaut hat dadurch, sondern dass die Governance funktioniert, also funktioniert schon dadurch, dass man sieht, man kann solche Zahlen, wie welche Anwendungen es gibt, aber auch, dass man dann halt irgendwann die Strukturen geschaffen hat, die man braucht. Dass man merkt es kommen nicht jeden Tag Tickets rein zu Themen, der Owner von der App hat das Unternehmen verlassen. Kannst du mich als Owner einsetzen, sondern dass man merkt, man hat Strukturen aufgebaut für sowas, dass das gar nicht mehr passiert ohne Awareness für solche Themen da ist.
- 68 [0:50:13.3] **I: RS** Dann bewegen wir uns jetzt sogar schon langsam auf das Ende vom Interview zu. Hast du noch Anmerkungen jetzt, wo du sagst, danach hast du jetzt nicht unbedingt gefragt, oder das ist eine gute Best Practice, die ich jetzt gerne noch teilen würde, wo ich sage, das wäre schon schön, wenn das mit drin wäre. Was sich für mich als erfolgversprechend herausgestellt hat.
- 69 [0:50:37.0] **B: RR** Also für uns war Erfolgsfaktor, glaube ich, dass wir, hatte ich vorher glaube ich auch schon gesagt, dass wir von allen Seiten kommen, also dass wir jemand haben, der auch von den Citizen Developern kommt, dass es nicht ein Projekt ist, was für die Leute wie von der IT übergestülpt wirkt. Und vor allem, dass auch von der bei uns, von den Personen gerade von der IT mitgetragen wird, also das es nicht so ein unliebsames Projekt ist, was man halt macht, weil Low Code-Plattformen hip sind, sondern was von quasi allen Seiten mitgetragen wird und dann sich, da arbeiten wir noch ein bisschen dran, und dann tatsächlich auch in der IT, da ein Bewusstsein verwurzelt ist, dass es kein Gegenmodell zur klassischen IT ist, sondern dass das halt einfach ein weiterer Pfeil im Köcher ist, dass man einfach neue Möglichkeiten dadurch hat. Für alle, von allen Richtungen, für professionelle Entwicklung, für kleinere Entwicklung, für gemeinsame Projekte zwischen IT und Fachbereichen. Ja, das ist so, finde ich, für mich das Wichtigste. Die anderen Sachen habe ich schon gesagt.



70 [0:52:12.8] **I: RS** Dann danke ich dir ganz sehr für das Interview und ich würde jetzt die Aufnahmen beenden.

71 [0:52:17.9] **Ende**

## Einwilligungserklärung Interview

Ich erkläre hiermit mein Einverständnis zur Nutzung der personenbezogenen Daten, die im Rahmen des folgenden Gesprächs erhoben wurden:

Datum des Interviews	23.06.2022
Namen der zu interviewenden Person	Robin Rosengrün
Name der interviewenden Person	Rocco Schenk
Kurzbeschreibung des Forschungsprojekts	Einführung von No Code/ Low Code aus der Sicht des Projektmanagements
Durchführende Hochschule	Hochschule Mittweida – University of Applied Science

Die Daten werden im Rahmen eines mündlichen Gesprächs erhoben, das mit einem Aufnahmegerät und Videoaufnahmegerät aufgezeichnet wurde. Zum Zwecke der Datenanalyse werden die mündlich erhobenen Daten verschriftlicht (Transkription), wobei die Daten nicht anonymisiert werden.

Vorhandene Kontaktdaten, der interviewten Person, werden aus Dokumentationsgründen in einem separaten Schriftstück lediglich den Gutachter\*innen der wissenschaftlichen Ausarbeitung zur Verfügung gestellt. Nach dem Abschluss des Projekts werden diese Daten gelöscht.

Der Speicherung der personenbezogenen Daten zu Dokumentationszwecken kann durch die interviewte Person jederzeit widersprochen werden. Die Teilnahme an dem Gespräch erfolgt freiwillig. Das Gespräch kann zu jedem Zeitpunkt abgebrochen werden. Das Einverständnis zur Aufzeichnung und Weiterverwendung der Daten kann jederzeit widerrufen werden.

Robin Rosengrün

\_\_\_\_\_  
Vorname und Name in Druckbuchstaben



\_\_\_\_\_  
Unterschrift

13.07.2022, Sindelfingen

\_\_\_\_\_  
Datum, Ort

## Anlagen 4, Interview EB03

RS: Rocco Schenk

FG: Fabian Gackstätter

1	[0:00:00.0] <b>Start (FG - Fabian Gackstatter/ RS - Rocco Schenk)</b>
2	[0:00:04.7] <b>I: RS</b> So dann ist die Aufnahme jetzt gestartet. Bist du mit der Aufzeichnung des Interviews einverstanden?
3	[0:00:10.8] <b>B: FG</b> Bin ich mit einverstanden, ja.
4	[0:00:13.9] <b>I: RS</b> Dann möchte ich dich ganz herzlich begrüßen bei dem Interview und würde dich zum Anfang gerne bitten, dass du dich einfach mal kurz vorstellst. Wer du bist? Woher du kommst?
5	[0:00:20.7] <b>B: FG</b> Ok, Fabian Gackstätter ist mein Name. Ich bin Solution Architect für Dynamics 365 und Power Platform bei Microsoft. In der Rolle, bei Microsoft, jetzt seit 7 1/2 Jahren ungefähr und habe davor, also ich bin jetzt seit knapp 20 Jahren dabei, um Dynamics und CRM-Lösungen generell einzuführen bei Kunden, angefangen als technischer Consultant bis hin zum Business Consultant bis hin zum Architekten. In der Rolle fühle ich mich auch ganz wohl. Studiert habe ich internationale Wirtschaftsinformatik, um ebenso die Informationsschnittstelle zwischen dem klassischen IT-Bereich und dem Kundenbereich oder dem Businessbereich herstellen zu können, so als Übersetzer dazwischen, Probleme zu analysieren und von einem Team an das andere zu kommunizieren. Und habe jetzt vor ein paar Jahren auch meinen MBA gemacht, um einfach da den Businessanteil zu verstärken und da auch auf einem anderen Level die entsprechende Auskunftsfähigkeit. Dass ich auch mal vorrechnen kann, warum machen wir das denn. Nicht um die IT zu bespaßen, sondern, was ist denn so ein Return on Investment für Business. Das ist die Geschichte und da hat sich so ein MBA, ich sage mal, man lernt eine neue Sprache.
6	[0:01:00.7] <b>I: RS</b> Dann wäre quasi meine erste Frage. Das Thema ist ja die Einführung von No Code/ Low Code. Wie definierst du No Code/ Low Code für dich?
7	[0:01:11.9] <b>B: FG</b> No Code/ Low Code für mich ist, sage ich mal, der Einstieg in Applikationsentwicklungen, für die ich selber kein Entwickler sein muss, also ein Entwickler, der klassischerweise mit einem Visual Studio arbeitet. No Code/ Low Code kommt von der anderen Seite. Leute, die sich mit PowerPoint auskennen, Leute, die sich mit Excel auskennen, die dann tatsächlich mit den entsprechenden Editoren oder Wizards sich einfache Applikationen, einfache Automationen, zusammenklicken können, ohne eine Zeile Code schreiben zu müssen. Das ist für mich No Code/ Low Code.
8	[0:01:43.5] <b>I: RS</b> Ok, hast du schonmal vom Citizen Development gehört?
9	[0:01:47.2] <b>B: FG</b> Klar, Citizen Developer ist eine der Stützen, eine meiner Hauptanliegen, tatsächlich auch bei unseren Kunden einzuführen im Unternehmen, weil, die Entwicklung macht ja klassischerweise bei den meisten Kunden nur einen sehr kleinen Anteil des gesamten Personals aus. Citizen Developer kann potenziell jeder Mitarbeiter werden, weil jeder Mitarbeiter, wenn ich ihm die richtigen No Code/ Low Code-Tools an die Hand gebe, kann er damit seine eigenen Applikationen bauen. Seine eigenen Flows bauen, eigene Automationen, ohne sich jetzt großartig in Entwicklungsthemen einarbeiten zu müssen. Also Citizen Developer sind für uns eine ganz wichtige Zielgruppe.

- 10 [0:02:20.3] **I: RS** Und mit den beiden Sachen, kannst du mal Vorteile der Plattform und vom Citizen Development nennen?
- 11 [0:02:27.1] **B: FG** Die Plattform selber, also die Microsoft Power Plattform als solches, ist eben unsere Sammlung, unser Einstieg in das Thema Citizen Development. Mit der Applikationsentwicklung, mit Power Automate als Workflow Engine, mit Power BI als Businessanalytics Tool, mit Virtual Agents und dem AI Builder, um auch tatsächlich komplexere Chatlösungen und auch die AI an den Citizen Developer heranzuführen. Der Vorteil des Citizen Developers generell ist einfach das Potenzial, das ich innerhalb eines Unternehmens habe, weil wie gesagt, die IT ist in der Regel ein eher kleiner Bestandteil eines jeden Unternehmens und kann dadurch auch sehr schnell zum Bottleneck werden, wenn es darum geht, neue Applikationen einzuführen, Regress aus dem Business zu bedienen, weil der klassische Entwicklungsprozess eben etwas komplexer gestrickt ist, mit den verschiedenen Qualitätsstufen, die durchlaufen werden müssen. Mit den verschiedenen Anforderungen. Ich muss ein Design erstellen. Ich muss einen technischen Blueprint erstellen, das muss genehmigt werden. Der Citizen Developer muss sich um solche Sachen in der Regel nicht kümmern. Der Citizen Developer kommt. Er hat eine Idee, dass er sagt: "Ich habe hier einen papierbasierten Prozess, den würde ich gerne digitalisieren." Oder: "Wir haben hier eine uralte Applikation, die eigentlich schon nicht mehr funktioniert. Die macht nicht viel, das könnte ich mit einem modernen Tool eigentlich selber ersetzen, diese Geschichte" und mit den entsprechenden Tools kann ein Citizen Developer dann innerhalb von ein paar Stunden sich seine eigenen Applikationen zusammenbauen, ohne den kompletten Entwicklungszyklus durchlaufen zu müssen, den klassische Enterprise-Applikationen einfach haben. Und dadurch, dass prinzipiell jeder zu einem Citizen Developer werden kann, habe ich entsprechend auf einmal viel mehr Potenzial, technische Lösungen zu implementieren, ohne meine IT als Bottleneck im Hintergrund zu haben.
- 12 [0:04:11.0] **I: RS** Und siehst du da aber auch Nachteile?
- 13 [0:04:13.7] **B: FG** Die Nachteile hast du natürlich auch, wenn ich das Ganze nicht kontrolliert mache. Ich gebe ja dem Citizen Developer eine gewisse Fähigkeit an die Hand, eigene Applikationen zu bauen, auf die Daten des Unternehmens zuzugreifen. Wenn ich da nicht die richtigen Richtlinien platziere, sprich, auf welche Daten kann denn ein Citizen Developer zugreifen? Wie kann er die verteilen? Wie kann er die verarbeiten? Habe ich natürlich das Risiko, dass diese Daten irgendwo hinkommen, wo sie nicht hingehören, zum Beispiel. Oder falls meine Supportorganisation nicht darauf vorbereitet ist, dass da jetzt auf einmal statt zehn Entwicklern hundert Entwickler in meinem System unterwegs sind, die dann im Zweifelsfall geflutet werden von Serviceanfragen, von Supportanfragen, auf die sie nicht vorbereitet sind, habe ich natürlich auch ein Problem. Also Citizen Developer funktionieren eigentlich nur, wenn ich das in einer kontrollierten Offensive beschreibe. Sprich, ich brauche eine solide Grundlage. Ich muss die Plattform sauber definiert haben. Ich muss die richtigen Guidelines und policies etabliert haben und ich muss aber auch meine Supportorganisation darauf vorbereitet haben und auch meine Changemanagement Organisationen, dass da jetzt auf einmal potenziell 90 % mehr Entwickler im Haus unterwegs sind, mit den entsprechenden Bedürfnissen. Wenn ich das tatsächlich abgedeckt habe, dann kann ich das Risiko durch so einen Citizen Developer auch drastisch reduzieren.
- 14 [0:05:29.2] **I: RS** Mit den Vor- und Nachteilen jetzt im Hinterkopf, würde mich jetzt interessieren, es gibt ja um das Thema aktuell viel Hype. Was sind denn realistische Erwartungen, die man wirklich an No Code/ Low Code-Plattformen generell denn jetzt wirklich schon stellen kann als Unternehmen und wo dann eher mehr Hoffnung als Realität dabei ist?
- 15 [0:05:51.6] **B: FG** Also schon mal, was ich realistisch tatsächlich einmal erreichen kann ist, dass wesentlich mehr Ideen umgesetzt werden, die aus sämtlichen Businessströmen kommen können. Die im normalen, im klassischen Unternehmen, sage ich mal, irgendwo in der

IT im Zweifelsfall hängen bleiben und niemals das Tageslicht erblicken. Der Citizen Developer hat eben selber die Möglichkeit, gerade kleinere Sachen umzusetzen, mit der sich eine normale IT überhaupt nicht beschäftigen würde. Also wenn es jetzt wirklich nur darum geht, ich habe hier einen Papierprozess. Ich habe hier ein Formular, was jeden Tag ausgefüllt werden muss. Da sind fünf Informationen drauf und wenn ich die ausgefüllt habe, scanne ich das Ding ein und schicke es an eine E-Mail-Adresse. Da wird dir niemals eine interne IT eine Applikation dafür bauen, weil du dann immer so weit unten im Stapel des Notwendigen landen wirst, dass es wahrscheinlich niemals so weit kommt. Die interne IT kümmert sich um die SAPs dieser Welt, um die Secret Server dieser Welt, um die Dynamics Solutions dieser Welt, aber gerade die kleinen Prozesse, die bleiben relativ häufig liegen, wenn ich dann nicht einen dedizierten Entwickler habe, der sich eben um diese kleinen Themen kümmert. Der Citizen Developer, der kann tatsächlich selber diese Sachen bauen. Die können dann im Zweifelsfall von der IT noch einmal gereviewt werden, bevor sie dann zum produktiven Einsatz kommen, nehmen aber den Großteil der Arbeit der IT einfach ab. Das heißt, die Chance, solche kleinen Applikationen tatsächlich zum Leben zu erwecken, ist wesentlich größer. Man darf jetzt natürlich nicht erwarten, dass ich dadurch meine komplette interne IT einsparen kann, weil, ich habe ja jetzt alles Citizen Developer. Also der klassische Citizen Developer fängt jetzt nicht an, das nächste SAP ERP-System zu bauen. Da muss man auch ganz klar trennen, was ist denn wirklich No Code/ Low Code? Was sind die Einsatzgebiete dafür? Was sind auch die realistischen Ergebnisse gegenüber der klassischen IT? Und dann habe ich halt den ganzen Zwischenraum dazwischen. Also es gibt ja nicht nur den Citizen Developer am einen Ende und den Pro Developer am anderen. Ich habe ja auch den Gesamtbereich in der Mitte. Den sogenannten Hybrid Approach oder Fusion Approach, wo ich tatsächlich sagen kann, hier arbeiten jetzt Citizen Developer und Pro Developer zusammen. Das heißt, der Citizen Developer hat vielleicht eine gute Idee für eine Applikation. Er hat auch schon mal das Design erstellt. Er hat mit dem No Code/ Low Code-Tool das UI erstellt, tut sich jetzt aber schwer, die Businesslogik zu implementieren, für die dann No Code/ Low Code irgendwann an ihre Grenzen stößt. Oder eben auch, wo dann Datentöpfe angezapft werden sollen, die dem klassischen Citizen Developer nicht zur Verfügung stehen und da kommt dann das Zusammenspiel mit dem Pro Entwickler dazu, der da tatsächlich unterstützt, diese Businesslogik zu bauen, der den Citizen Developer auch unterstützt bei einem Applikation Lifecycle Management. Das heißt, die Applikation, die der Citizen Developer auf seinem eigenen Environment entwickelt hat, die dann transportiert wird auf ein Testsystem, die dann transportiert wird auf ein Produktivsystem, zum Beispiel. Und das sind, zum Beispiel, zehn Szenarien, die ich mit der Power Platform komplett abdecken kann. Von dem Citizen Developer, der sich mit Canvas Apps auf SharePoint Basis, mit denen fängt er mal an, bis zum Pro Developer, der den kompletten Stack von Dynamics Power Platform, Azure zur Verfügung hat und alles, was dazwischen passiert. Und das muss ich halt als Unternehmen auch sauber definieren. Wo möchte ich denn, dass meine Leute ihre Zeit, ihre Ressourcen verbringen? Sind das wirklich die klassischen Citizen Developer, wo ich sage, ihr habt SharePoint und Canvas-Apps? Das ist eure Spielwiese oder erweitere ich dieses Spektrum, weil ich sage: "Ich habe durchaus technisch affine User, auch in meinem Business, denen gebe ich auch mehr Teile zur Verfügung und stelle ihnen auch mehr Datentöpfe zur Verfügung, um wirklich eigene Applikationen zu bauen, die für ein ganzes Team oder eine ganze Abteilung dann noch zum Einsatz kommen können."

- 16 [0:09:15.1] **I: RS** Du hast ja schon angesprochen, es kommt zu einer Entlastung der IT-Abteilung. Es gibt ja derzeit ein großes Problem auch mit Fachkräften, auch im IT-Bereich. Bist du der Meinung, dass dieses Thema No Code/ Low Code und Citizen Development den Fachkräftemangel in der IT beheben kann? Das der dem zumindest Abhilfe schaffen kann, in irgendeiner Form?

- 17 [0:09:41.3] **B: FG** Also ich denke, er wird ihn nicht komplett beheben können, denn die Anforderungen an einen IT-Mitarbeiter sind ja in der Regel mehr als nur eine reine Entwicklungsaufgabe. Also wenn ich mir angucke, was sind denn meine Fachkräfte in der IT? Das sind die Entwickler auf der einen Seite, das sind aber auch die Leute wie Quality Assurance Administrator, das sind Leute wie Datenbank-Administratoren, SAP-Administratoren. Das sind ja jetzt nicht die Rollen, die ein Citizen Developer übernehmen wird in meiner internen IT. Wo der Citizen Developer die interne IT definitiv entlassen wird, ist es, wenn es wirklich darum geht, Innovationen umzusetzen, Ideen umzusetzen, die von außen kommen, die von innen kommen. Die tatsächlich in Applikationen übersetzt werden müssen, wo in der Regel erst mal der klassische IT-Entwickler in der Pflicht steht. Den kann ich auf alle Fälle durch die Citizen Developer entlasten, weil auf einmal sein Stack, was er abarbeiten muss, sind wirklich die Dynamics, die SAP-Lösungen dieser Welt, aber eben nicht mehr den Speiseplan für die Kantine zu programmieren, den kann auch ein Citizen Developer zusammenstellen. Oder das Weihnachtswichtel-Spiel oder den Approval-Prozess für einen Urlaubsantrag. Alles Themen, die heute erst mal in der IT landen, bei irgendeinem Entwickler auf dem Tisch. Einen Großteil davon kann ich tatsächlich an die Citizen Developer geben, weil sie eben von der Komplexität her relativ klein gehalten sind und kann damit zumindest den Fachkräftemangel in der Entwicklung entgegenwirken. Wie gesagt, die restlichen IT-Themen werden auch weiter klassische IT-Themen bleiben. Also der Pfad ist nicht, dass ein Citizen Developer zum SAP-Admin wird und da eine Rolle besetzt. Aber die Entwicklung auf alle Fälle.
- 18 [0:11:16.9] **I: RS** Als kurzen Spoiler. Alle Fragen, die ich dir jetzt quasi stelle, gehen immer von dem Ursprungsszenario aus: Ein Unternehmen möchte gerne eine No Code/ Low Code-Plattform bei sich einführen.
- 19 [0:11:28.1] **B: FG** Ja ok.
- 20 [0:11:30.3] **I: RS** Bevor so ein Unternehmen ja diese Plattform einführen kann bei sich, ist es natürlich auch spannend, was denn Voraussetzungen sind, damit es das überhaupt einführen kann. Also da spielen so Sachen eine Rolle für mich wie Mindset. Braucht das Unternehmen eine bestimmte Größe? Muss es eine gewisse Manpower dahinter haben? Oder müssen die Mitarbeiter eine bestimmte Befähigung auch haben, im Vorhinein, bevor das losgehen kann?
- 21 [0:11:53.1] **B: FG** Ok, also als Grundlage, das Mindset muss natürlich passen. Also das sich ein Unternehmen wirklich darauf einstellt: "Wir möchten No Code/ Low Code und das Thema Citizen Development einsetzen." Das ist halt ein drastisches Umdenken, im Sinne von: Ich habe eine interne IT-Abteilung, die sich um alle IT-Themen kümmert. Bei denen, das breche ich auf. Alles was jetzt wirklich an IT-Anfragen reinkommt, muss ich eben unter dem Gesichtspunkt betrachten: Ist das nach wie vor ein Thema für meine klassische IT? Oder sind das tatsächlich Themen, die auch Citizen Developer erledigen können? In der Regel kommt tatsächlich der meiste Input für diesen Stream von den Citizen Developern selber, in dem sie sagen: "Wir haben hier Ideen, wir haben hier Probleme im Unternehmen, die relativ einfach zu adressieren wären, mit denen sich die interne IT aber nicht beschäftigt." Also, ich sage mal, das Mindset muss auf alle Fälle passen. Das Vertrauen gegenüber den Mitarbeitern muss auch da sein, dass man sie tatsächlich jetzt an Themen ranlässt, die klassischerweise jetzt nicht bei ihnen gelegen haben. Das setzt natürlich auch voraus, das ich intern die richtigen Weichen stelle. Also ein Betriebsrat, zum Beispiel, muss auch involviert sein. Wir haben Kunden, wo der Betriebsrat eben so weit weg davon ist, dass der sagt: "Alles was als Applikation im Unternehmen ausgerollt oder entwickelt wird, muss durch einen Betriebsrat genehmigt werden." Was für einen klassischen IT-Bereich natürlich ok ist. Was aber das Citizen Development krass ausbremsen würde, weil, wenn jetzt jeder anfängt, Applikationen zu bauen und jede einzelne Applikation vom Betriebsrat genehmigt werden muss, ist dein Betriebsrat auf einmal der totale Flaschenhals, weil, dafür sind die ja nicht ausgelegt, auf einmal

von hunderttausend Leuten hunderttausend Applikationen zu sichten und zu genehmigen, zum Beispiel. Es reicht nicht nur, einfach die technische Basis zu schaffen und zu sagen: "Wir haben jetzt die Plattform gekauft. Wir haben die scharf geschaltet. Hier sind die Lizenzen, fangt an zu bauen." Sondern ich muss auch intern vom Mindset her und auch vom Changemanagement mich als Unternehmen darauf einstellen, dass die richtigen Richtlinien gesetzt sind, dass ein Betriebsrat involviert ist, dass auch die richtige Kommunikation und das richtige Training für meine User zur Verfügung steht. Weil, es nutzt nichts, einfach nur die Plattform hinzustellen und dann sagen: "Seht zu, wie ihr klarkommt. Also hier sind von mir aus noch 2 Microsoft-Links, da könnt ihr euch mal Trainings anschauen." Sondern ich muss mir wirklich Gedanken machen, wie fördere ich denn auch meine Citizen Developer, im Sinne von Communities, im Sinne von Champions, die diese Communities unterstützen und auch im Sinne von: wie kann denn die interne IT dabei unterstützen. Also, wie können die, zum Beispiel, Trainings organisieren. Wie können die Community Meetings übernehmen. Wie können die Fragen beantworten, die die Citizen Developer an sie stellen. Also diese Einsatzplanung muss ich entsprechend auch anpassen. Und dann muss ich natürlich auch die technische Grundlage sauber definiert haben. Also da kommen dann die Nicht-Entwickler aus meiner zentralen IT wieder zum Tragen, sprich die Power Platform-Administratoren, ein Active Directory-Administrator, der jetzt auch die richtigen Rechte tatsächlich freischalten kann und der auch das ganze Thema monitoring dann überwacht. Ich will ja als Unternehmen wissen, was machen denn meine Citizen Developer überhaupt im System. Bauen die Applikationen? Bauen die Automationen? Nutzen die tatsächlich auch nur die Tools, die eigentlich dafür vorgesehen sind oder haben die aus Versehen irgendwelche Sachen offengelassen? Dass ich zum einen das monitoren kann, zum anderen aber auch darauf reagieren kann. Was passiert denn, wenn der Rocco jetzt eine Applikation gebaut hat und teilt die mit hunderttausend seiner Kollegen. Ist das so vorgesehen? Wollen wir das überhaupt? Und wie unterstützen wir Rocco dabei, wenn seine Applikation jetzt wirklich gut ist? Wie verknüpfe ich Rocco mit unserem zentralen Support? Weil, wenn ich jetzt eine Applikation baue und teile die mit meinem Team, mit fünf anderen Leuten, die kriege ich selber nach wie vor unterstützt. Wenn ich jetzt eine Applikation baue, die morgen von hunderttausend Leuten bedient wird, dann werde ich wahrscheinlich unter der Last der Issues und feature Request und Anfragen zusammenbrechen und kann meinen eigenen Tagesjob eigentlich nicht mehr machen. Also, über genau solche Themen muss ich mir Gedanken machen. Was sind denn die Citizen Developer-Applikationen? Wie, in welchem Rahmen sollen die sich bewegen? Und wann wird aus einer Citizen Developer-Applikation tatsächlich eine businesskritische, die im ganzen Unternehmen eingesetzt wird? Und das sind dann alles solche Gedanken, die mache ich mir idealer Weise im Vorfeld. Ich muss leider dazu sagen, die Hälfte unserer Kunden fängt erstmal damit an, dass sie sagt: "Wir haben jetzt die Lizenzen gekauft. Wir bauen jetzt mal eine Applikation." Und nach einem halben Jahr ist dann: entweder brennt die Hütte, weil die interne IT dann keinerlei Kontrolle mehr darüber hat, die weiß nicht, wo sie gucken muss. Sie weiß nicht, wie sie darauf reagieren soll und stellt im Zweifelsfall die Plattform wieder ein. Oder es passiert, dass eben die User nicht abgeholt werden. Es wird kein Training zur Verfügung gestellt. Es gibt keinen Feedback-Mechanismus, es gibt keine Trainings und die Plattform hebt eben nicht ab und es werden keine Applikationen gebaut und die Lizenzen gammeln einfach vor sich hin. Also, das sind so die zwei Szenarien, die wir sehen, wenn ein Kunde eben ohne die saubere Basis loslegt. Auf der anderen Seite, mit kontrollierter Offensive, kriege ich genau diese Richtlinien platziert, so dass ich definieren kann: Welche Szenarien habe ich denn? Was sind klassische Entwickler-Applikationen? Was sind klassische Citizen Developer? Wie sieht die Zone dazwischen aus? Wie sieht auch eine Progression aus von einer Applikation, die ursprünglich mal ein User für sich gebaut hat? Die dann aber so gut war, dass sie irgendwann zur Enterprise-Applikation geworden ist. Wie sieht dieser Lebenszyklus aus? Und wenn ich die Sachen alle sauber definiert habe, dann funktioniert so eine Plattform auch.

- 22 [0:17:29.9] **I: RS** Und wie siehst du das von den Faktoren Geld, Manpower, Unternehmensgröße? Sind die irgendwie relevant bzw. siehst du noch Faktoren, die ich jetzt nicht genannt haben könnte?
- 23 [0:17:42.3] **B: FG** Also, ich sage mal, die Manpower an sich spielt, glaube ich, die geringste Rolle an der Stelle. Also es gibt keine spezifische Unternehmensgröße, wo ich sagen würde: "Wenn ihr weniger als X-hundert Leute seid, dann lohnt es sich eigentlich nicht." Weil Citizen Developer, wie gesagt, kann jeder sein. Das kann ich in einem Unternehmen mit zehn Mann anfangen. Das machen wir aber auch bei Konzernen mit fünfhunderttausend Leuten. Also die ganze Spanne ist da. Ich muss mir natürlich die Zeit dafür nehmen. Also das ist kein Selbstläufer, wenn ich jetzt eine No Code/ Low Code-Plattform einführe und der Schwerpunkt liegt jetzt wirklich auf Plattform. Es geht ja jetzt nicht um eine einzelne Applikation, die ich einstelle, wo ich sage: "Ok, ich habe jetzt hier das neue System implementiert. Dafür brauche ich jetzt zwei Admins, die sich darum kümmern." Wenn ich eine Plattform zur Verfügung stelle, da sind eben viel breitere Themen dran, um die ich mich kümmern muss. Das heißt, ich brauche eine entsprechend große Admintruppe, die tatsächlich diese ganzen Aktivitäten der User managen kann. Ich brauche aber auch jetzt eine entsprechende Changemanagementorganisation, die tatsächlich damit umgehen kann. Also in den Implementierungen, die wir sehen, ist die technische Zurverfügungstellung der Plattform maximal 20 % vom gesamten Volumen, was so ein Projekt ausmacht und die restlichen 80 % sind tatsächlich, wie bereite ich denn meine Organisation darauf vor, das jetzt tatsächlich umzusetzen, das anzunehmen, die Leute zu trainieren, die Leute aber auch bei der Stange zu halten. Mit vernünftigen Informationen zu füllen, mit den entsprechenden Aktivitäten, wie einem Hackathon, wie mal eine Expertenrunde mit Walk-in-Clinic und allem. Diese Sachen, die ich zur Verfügung stellen muss. Also da muss man als Unternehmen natürlich auch Geld in die Hand nehmen, weil diese Rollen müssen gestacked werden. Das sind in der Regel keine Sachen, die ich zu einem anderen Job nebenher machen kann. Und damit tun wir uns auch am Anfang schwer, wenn ich jetzt so ein Team zusammenstelle auf Kundenseite und sozusagen: "Wen brauch ich denn jetzt? Ok, ich brauche die drei Admins für den technischen Teil und dann brauche ich aber noch die zehn Leute, die sich um das Thema Changemanagement und Service Organisation kümmern." Wo die meisten Kunden sagen: "Ich dachte, das wäre ein IT-Thema." Und das ist so eine Plattformscheidung halt nicht. Die betrifft im Zweifelsfall das ganze Unternehmen und genau so muss ich die auch aufhängen. Dass es eben nicht mit den drei Admins getan ist, die das jetzt noch nebenhermachen, sondern es ist ja ein organisationsweiter Change, den ich da einführe und den muss ich auch mit der entsprechenden Manpower unterfüttern. Sonst, ich bin noch am überlegen, ob ich noch... Also das war jetzt Geld, Manpower, was man braucht und, wie gesagt, von der Organisationsstruktur muss halt auch das Mindset da sein und auch das Vertrauen seinen Mitarbeitern gegenüber, weil, ansonsten lasse ich halt viel Potenzial von so einer No Code/ Low Code- Plattform liegen. Wir haben Kunden, die es tatsächlich einführen, aber dann nur mit tatsächlichen Entwicklern bestücken. Das heißt, die ganzen Citizen Developer bleiben nach wie vor außen vor. Was natürlich dann viel Potenzial von so einer Plattform wieder verschenkt. Zum einen dem Misstrauen geschuldet, dass man sagt: "Ich traue das meinen Mitarbeitern einfach nicht zu." Oder: "Wir können die nicht in den entsprechenden Kontrollmechanismus hineinbringen, wie ich es mit meiner zentralen IT machen kann." oder weil man sagt: "Ich habe einfach nicht die Manpower, um das zu betreuen, die Geschichte. Wir haben nicht genügend Leute für Projekte, dann können wir jetzt nicht nebenher noch eine neue Plattform betreiben." Dann lass ich halt viel Potenzial liegen, weil, wie gesagt, es geht nicht darum, ein neues Office Paket einzuführen, was ich mit ein, zwei Mann betreiben kann und tragen, sondern No Code/ Low Code-Plattform heißt halt eben auch, ganz neue Aufgabengebiete, die es so vorher nicht gab.



- 24 [0:21:23.4] **I: RS** Bei der Einführung, siehst du da auch Risiken, Hürden oder Problemstellungen, auf die Firmen stoßen könnten, abgesehen von denen, die du jetzt schon genannt hast, die du besonders herausstellen möchtest?
- 25 [0:21:36.2] **B: FG** Also, die zwei größten Risiken, die ich eigentlich sehe bei der Einführung für uns, denen wir uns normalerweise gegenübersehen, ist zum einen mangelndes Personal, weil gerade solche Center of Excellence-Implementierungen, die wir am Anfang machen, da ist eben wichtig, dass der Kunde da mitarbeitet. Das ist viel Knowledge-Transfer von unserer Seite, das heißt, da müssen dann aber auch die entsprechenden Kundenressourcen verfügbar sein, präsent sein und mitarbeiten, damit wirklich da was Brauchbares bei rauskommt. Und es ist eben auch wichtig, dass die Kunden verstehen, dass wir ihnen nicht vorkauen können: "So müsst ihr eine No Code/ Low Code-Plattform aufsetzen." Wir haben zwar entsprechende Best Practices, aber jedes Unternehmen tickt eben anders und unser Job ist es dann, die Plattform so anzupassen und den Kunden durch den richtigen Entscheidungsprozess durchzuführen, dass am Ende die Plattform dabei rauskommt. Also so als Beispiel, wir haben als Kunden auf der einen Seite eine SAP, wo quasi 90 % der Mitarbeiter Entwickler sind und ich habe auf der anderen Seite Kunden, das sind Ministerien, die eine Plattform einführen, wo ein Bruchteil der Leute Entwickler sind. Das heißt, da jetzt zu vermuten, ich platziere genau ein und dieselbe Lösung mit denselben Policies, mit denselben Richtlinien und die funktioniert bei beiden Unternehmen, ist unrealistisch und das müssen die Kunden halt auch verinnerlichen, dass wir ihnen dabei helfen, die Plattform und das ganze Tooling drum herum auf ihre Bedürfnisse zuzuschneiden, damit sie ihre Prozesse unterstützen, damit sie ihre Personalstruktur auch unterstützen. Und das kann ich Ihnen halt nicht vorgeben. Also, das ist das Problem, dass die Kunden das personell meistens unterschätzen. Ein anderes Problem, was wir manchmal haben, dass wir uns relativ schnell in einer internen IT-Bürokratie verlieren, weil eben eine Power Platform Silos aufbricht. So eine Power Platform betrifft nicht nur einzelne Teams. Das heißt, ich setze mich nicht nur mit einem Exchange Admin zusammen oder ich setze mich nicht nur mit einem Dynamics Admin zusammen oder mit einem SAP Admin zusammen. Eine Plattform heißt, ich betreffe die alle, das heißt, in vielen Fällen müssen die sich zum ersten Mal an einen Tisch setzen und gemeinsam über Regeln definieren, die sie bisher schön in ihrem Silo für sich verwalten konnten. Das funktioniert halt mit einer Plattform nicht und da haben manche Unternehmen dann ein Problem, zum einen die richtigen Leute zu identifizieren. Wen brauche ich denn da überhaupt? Und die dann auch zur Verfügung zu stellen. Und dann auch gemeinsam mal eine Entscheidung zu treffen, die nicht sechs Monate irgendwo in einem Bürokratiedschungel untergeht. Gerade Banken sind da ein schwieriges Thema, da tatsächlich mal alle Kunden an einen Tisch zu bekommen, die richtigen Stakeholder, die dann auch gemeinsam eine Entscheidung treffen können, die eben nicht zwei Jahre dann durch interne Gremien durchlaufen muss. Das sind so meine zwei größten Hürden. Zum einen die interne Komplexität des Kunden und zum anderen die nicht vorhandenen Ressourcen, die eigentlich gebraucht werden. Die Technik ist in der Regel das geringste Problem. Die kriegen wir immer gebacken, aber die richtigen Leute im richtigen Zeitraum zu bekommen, ist schwierig.
- 26 [0:24:36.6] **I: RS** Du hast ganz kurz zum Anfang das Center of Excellence erwähnt. Kannst du ganz kurz sagen, was das ist?
- 27 [0:24:42.0] **B: FG** Center of Excellence für uns ist unsere Organisationsstruktur, die wir bei unseren Kunden etablieren möchten. Ein Center of Excellence für die Power Platform zum Beispiel. Die entsprechend die Governance orchestriert. Die entsprechend die Tools innehat, die zur Verfügung gestellt werden. Die aber auch die Themen Changemanagement und Service Support zum Beispiel betreuen. Personell, was muss ich da an Prozessen auf die Beine stellen? Was muss ich an Tools zur Verfügung stellen? Was sind die Trainings? Was sind die Dokumente, die ich zur Verfügung stellen muss? Und wie könnten jetzt solche Lebenszyklen

von Entwicklern und von Applikationen aussehen? All diese Themen fassen wir normalerweise zusammen unter einem Center of Excellence.

- 28 [0:25:20.7] **I: RS** Danke noch mal für die Klarstellung. Wenn du in einem Unternehmen bist und du sollst jetzt quasi dein Projektteam zusammenstellen. Welche Projektorganisationsform findest du am erfolgversprechendsten für die Einführung von No Code/ Low Code?
- 29 [0:25:39.5] **B: FG** Also, wir setzen uns normalerweise zusammen, ein Projektteam auf unserer Seite, mit einem insgesamt Architekten. Also jemand wie ich, der die Fäden zusammenhält und dann habe ich verschiedene Consultants und weitere Architekten dabei, die die einzelnen Teilströme abdecken. Also für uns ist eine Center of Excellence oder eine Power Platform-Einführung in 5 Bereiche unterteilt. Einmal das Thema Governance an sich. Also wie sieht die Projektstruktur aus und auch auf Kundenseite. Dann habe ich den String des Platform enablements. Also wie muss die Plattform tatsächlich organisiert sein? Welche Rollen brauche ich? Welche technischen Capabilities müssen gegeben sein? Welche Policies müssen im Unternehmen eingeschaltet sein? Dann habe ich meinen String des Developer enablements und hier trainiere ich jetzt tatsächlich die Kundenressourcen und auch die Citizen Developer, bereite Applikationen vor. Dann der Bereich des Changemanagements, also wie muss sich die Organisation vorbereiten, im Sinne von Training? Und als Letztes den Bereich die Supportorganisation. Wie verknüpfe ich jetzt meine neue Plattform, meine Citizen Developer, mit den entsprechenden internen Supportstrukturen. Das sind so die fünf Arme, die wir normalerweise haben und die sind dann gestaffed auf unserer Seite mit den entsprechenden SME's, sprich Entwicklung, also Entwickler, die den Citizen Developer Stream unterstützen. Funktionale Berater, die beim Thema Plattform enablement unterstützen und ACM und MSM Consultants, die eben Changemanagement und Service Organisation unterstützen. Es ist schon ein relativ großes Team, wenn wir wirklich die komplette Plattform implementieren.
- 30 [0:27:17.4] **I: RS** Und würdest du von Kundenseite die Mitarbeiter, die dem Projektteam angehören, aus ihren originären Abteilungen herauslösen? Oder nur zu 50 % herauslösen, damit sie trotzdem noch die Basis haben? Wie händelst du das?
- 31 [0:27:32.5] **B: FG** Teils, teils. Also manche Leute sollten sich wirklich Vollzeit dann auf die Plattform konzentrieren. Gerade was das Plattform Enablement angeht, die Admins, die wir brauchen. Die werden in der Regel stärker beansprucht. In den anderen Streams ist es unterschiedlich. Also an mancher Stelle brauche ich einfach nur die Expertise auf Kundenseite, um Entscheidungen zu treffen. Das kann ich natürlich nebenhermachen. Auf der anderen Seite, in Bereichen wie Changemanagement, wo es dann wirklich darum geht, Communitys zu betreuen, Trainings zu organisieren. Da ist es in der Regel auch über 50 % Leistung, die ich von einem Mitarbeiter brauche und da macht es in der Regel schon Sinn, ihn aus seiner aktuellen Rolle herauszulösen oder im Rahmen seiner aktuellen Rolle für das Thema Plattform zu dedizieren. Also wenn ich ein internes Changemanagementteam habe, dann muss ich ihn ja jetzt da nicht herausnehmen. Ich muss nur sicherstellen, dass er tatsächlich dann die 50 % bis 100 % Kapazität hat, um das Thema Changemanagement für die Power Plattform zu betreiben.
- 32 [0:28:28.1] **I: RS** Und du hast dein Team jetzt aufgestellt. Welche projektmanagementtechnische Vorgehensweise bevorzugst du? Es zielt auf die klassischen Wasserfallmodelle oder auf einen agilen Ansatz oder einen hybriden oder hast du ein ganz neues Modell, von dem ich noch nichts weiß?
- 33 [0:28:48.9] **B: FG** Also, wir sind normalerweise agil unterwegs. Das heißt, wir fangen an mit einem Sprint Null, in dem wir erst mal eine Reihe von Workshops durchführen, um den Kunden auf ein gewisses Niveau aufzuschauen und nach diesem Sprint Null planen wir dann wirklich Entwicklungssprints, um zu sehen: "Ok, das sind jetzt die Arbeitspakete, die wir

tatsächlich implementieren möchten, in dem und dem Zeitraum." Auch immer dem Umstand geschuldet, wie viel Mitarbeit habe ich jetzt auf Kundenseite. Also, habe ich tatsächlich dedizierte Ressourcen, die mir zu 100 % zur Verfügung stehen. Das heißt, ich kann auch wirklich jeden Tag Arbeitspakete abarbeiten. Oder habe ich eben nur Leute, die maximal 50 % zur Verfügung stehen, dann brauche ich natürlich entsprechend länger, um die Pakete umzusetzen. Aber wir versuchen es normalerweise schon in einer agilen Methode umzusetzen, einfach auch, weil die Pakete sehr unterschiedlich sind und die jetzt in einem Wasserfall anzubringen, wäre sehr schwierig.

- 34 [0:29:40.2] **I: RS** Also rein agil oder sagst du, du willst auch Elemente aus dem Wasserfallmodell haben? Keine Ahnung, Phasen, weil die einen Vorteil bringen.
- 35 [0:29:49.1] **B: FG** Eigentlich nicht. Also, wie gesagt, der einzige Wasserfallanteil ist am Anfang so diese Introworkshops, die wir mit dem Kunden machen, über zwei bis drei Wochen, um einfach die richtigen Basisvoraussetzungen zu schaffen und danach geht es aber in den agilen Modus über, wenn wir definieren: "Ok, das sind jetzt die Applikationen, die wir bauen möchten, als Teil der Einführung. Das sind die Organisationsstrukturen, die wir bauen möchten. Das ist das Tooling, das wir selber anfassen müssen." Und das ist aber ein rein agiles Vorgehen an der Stelle.
- 36 [0:30:16.9] **I: RS** Und kannst du mal die Vorteile nennen, warum du dieses agile Vorgehen den anderen gegenüber bevorzugst?
- 37 [0:30:23.3] **B: FG** Weil die Bestandteile zum einen recht unabhängig voneinander sind. Also das Thema Changemanagement und Platform enablement, zum Beispiel, kann völlig parallel laufen. Es ist von unterschiedlichen Stakeholdern und Mitarbeitern betroffen, die auch mit unterschiedlichen Geschwindigkeiten unterwegs sind. Das heißt, wenn ich die jetzt alle in ein Wasserfall versuchen würde reinzupacken. Der Wasserfall ist immer nur so schnell wie das langsamste Teammitglied und in der agilen Methode habe ich eben die Chance, Sachen auch vernünftig zu parallelisieren und zu sagen: "Ok, im Entwickler-Stream bauen wir jetzt diese zehn Applikationen. Das ist unser Entwicklungsstream und im Thema Changemanagement, das sind die drei Communitys, die wir aufsetzen, die parallel dazu laufen." Also die agile Methode bietet sich einfach hier eher an, weil ich schneller zu meinen Arbeitspaketen komme und die auch unabhängig von den anderen Bereichen entwickeln kann.
- 38 [0:31:11.3] **I: RS** Und wenn du mit dem Unternehmen in diese agile Methodik einsteigst. Gibt es auch Hürden oder Sachen, auf die man als Unternehmen achten sollte? Also Nachteile, die man eventuell ausgleichen muss?
- 39 [0:31:25.6] **B: FG** Ja, ist jetzt aber weniger der Power Platform geschuldet als der agilen Methodik insgesamt. Wenn ich jetzt in ein Unternehmen komme, die mit der agilen Methode bisher noch nie zusammengearbeitet haben, die tun sich natürlich erst mal schwer, weil ich dann erst mal eine agile Projektvorgehensweise mit dem Kunden klären muss, bevor es überhaupt an die Inhalte gehen kann. Also, das ist so die Hürde. Ist aber losgelöst von dem Thema Power Platform an sich. Das Gleiche hätte ich, wenn es jetzt um eine Dynamics Einführung geht, zum Beispiel. Im schlimmsten Fall geht man dann halt auch wirklich auf ein Wasserfallmodell über, wenn der Kunde sagt: "Agil ist definitiv nicht unseres. Wir brauchen den kompletten Plan von vorn bis hinten." Dann machen wir das auch. Also, wir bestehen nicht auf einer agilen Methode. Es ist halt unsere Empfehlung das zu tun, weil wir damit in der Regel bessere Ergebnisse erzielen, wenn der Kunde den Weg aber so nicht mitgehen kann, dann machen wir es auch ganz normal im Wasserfall. Die Geschichte dauert in der Regel länger und ist dann halt auch schwieriger für den Kunden zu greifen, weil es dann halt auch einfach länger dauert, bis ich mal was in der Hand habe hinterher.

40	[0:32:30.9] <b>I: RS</b> Und welche Methoden aus dem agilen Methodensatz, aus den Tools, welche benutzt ihr da so?
41	[0:32:38.2] <b>B: FG</b> Also wir haben in Microsoft intern unsere eigene Methodik, die sehr stark an Scrum angelehnt ist.
42	[0:32:44.0] <b>I: RS</b> Benutzt ihr noch andere oder ist das wirklich so diese Scrum Logik?
43	[0:32:50.4] <b>B: FG</b> Scrum Methodik ist die Basis für uns.
44	[0:32:52.4] <b>I: RS</b> Also Kanban nutzt ihr zum Beispiel nicht?
45	[0:32:54.3] <b>B: FG</b> Benutzen wir im Sinne der Tools, wird Kanban eingesetzt. Aber, wie gesagt, wir haben eine Scrum ähnliche Methodik, unsere Surestep Methodik, die größtenteils auf Scrum aufsetzt, die aber zum Beispiel auch Kanban Boards im ( <b>unverständlich</b> ) umsetzt. Also wir versuchen da so ein bisschen, das praktischste aus beiden Welten oder aus mehreren Welten zusammenzusuchen, was sich in unseren Implementierungsprojekten bewährt hat. Spiegelt sich dann auch in den Artefakten wider, die erstellt werden. Spiegelt sich auch in den Rollen wider, die wir besetzen innerhalb dieser Projekte. Also kein hundertprozentiges Scrum, aber jemand, der Scrum beherrscht, wird sich bei uns sehr schnell zu Hause fühlen.
46	[0:33:32.0] <b>I: RS</b> Du hast ja jetzt schon Rollen angesprochen. Es ist ja ein Scrum-ähnliches. Hast du dann trotzdem diese gleichen Rollen: Scrum Master, Product Owner, das Developmentteam?
47	[0:33:48.8] <b>B: FG</b> Ja die nutzen wir auch.
48	[0:33:50.2] <b>I: RS</b> Gibt es weitere Rollen, die du sagst, die sind wichtig innerhalb des Teams?
49	[0:33:53.8] <b>B: FG</b> Was bei uns halt wichtig ist, sind solche Leute wie ein Changemanagement- Koordinator, zum Beispiel, der eine eigene Rolle ist. Dann der Architekt selber, der irgendwo zwischen den Stühlen sitzt, der Teil der Entwicklung dieses Teams ist, aber auch sehr eng mit einem Project Manager und dem Product Owner zusammensitzt, einfach im Sinne von Übersetzung. Von was der Product Owner gerne hätte bis zu dem, was beim technischen Team dann hinterher ankommt, da unterstützt der Architekt halt sehr stark bei uns.
50	[0:34:24.9] <b>I: RS</b> Es kommt ja öfters auch vor, dass Änderungswünsche an das Projekt herangetragen werden. Das kann von innerhalb oder von außerhalb des Projekts erfolgen. Wenn du möchtest, kannst du das auch aufgliedern, wenn da ein Unterschied besteht für dich. Wie geht ihr damit um?
51	[0:34:45.6] <b>B: FG</b> Also, wenn wir mit klassischen, mit agilem Projekt vorgehen, ist jede Änderung, die reinkommt, wird auch so übers Changemanagement abgewickelt, dass wir sagen: "Die kommt ins Backlog mit rein." Und dann wird diese Änderung gegen das bisherige Backlog gemessen. Also wenn die wichtig ist, dann rutscht die entsprechend weiter nach vorn, dann fällt halt hinten irgendwas anderes runter in der Geschichte. Ganz klassisches Changemanagement. Was ein bisschen schwierig wird in einer Plattform-Einführung in dem Stream ist, dass ich eben nicht nur Applikationen baue, sondern ich habe ja mehrere Streams, die parallel agieren, wenn ich da irgendwelche Changes habe, die mehrere Streams betreffen, ist die Koordination natürlich teilweise tricky. Wenn jetzt, zum Beispiel, irgendwelche Regulatoren sich ändern, die jetzt eben nicht nur meine Entwicklung betreffen, sondern alle fünf Streams die ich habe, muss ich halt gucken, wie kann ich den jetzt sauber einbauen, dass der Change auch tatsächlich dann irgendwie parallel in verschiedenen Streams umgesetzt wird. Ansonsten nützt es mir nix, wenn meine Applikationen das jetzt schon umgesetzt haben, mein Changemanagement aber frühestens in sechs Monaten dazu kommt. Also die

	Priorisierung muss dann halt Stream-übergreifend erfolgen, wenn der Change das benötigt. Das macht es dann etwas komplexer, wäre aber in jedem normalen Umsetzungsprojekt auch so.
52	[0:36:09.3] <b>I: RS</b> Also nur noch mal kurz, um es zusammenzufassen. Es ist die ganz klassische Scrumlogik, wie ihr damit umgeht.
53	[0:36:11.9] <b>B: FG</b> Ja
54	[0:36:12.5] <b>I: RS</b> Innerhalb von Unternehmen gibt es ja nicht nur wichtige Rollen im Projekt, sondern auch außerhalb des Projektes. Leute, die man abholen muss. Welche wichtigen Stakeholder siehst du da im Unternehmen?
55	[0:36:24.0] <b>B: FG</b> Also, ganz wichtig, gerade bei der Power Platform. Die meisten Unternehmen denken, das ist erst mal nur ein reines technisches Thema. Das heißt, es hängt irgendwo unter meinem CTO, seiner Organisation, seiner Admins und der Rest vom Unternehmen hat damit nichts zu tun. Falsch. Ich habe ja gesagt, Plattform betrifft am Ende des Tages alle. Das heißt, ich brauche da auch einen Changemanagementlead auf Kundenseite, der dann das Thema Training beinhaltet. Ich muss meinen Betriebsrat mit abholen, der in den meisten Projekten klassischerweise nicht beteiligt ist, aber dadurch, dass jetzt meine Plattform prinzipiell jeden Mitarbeiter betrifft, muss der Betriebsrat halt involviert sein, muss die Entscheidungen auch mitgehen, muss sich auch darauf einstellen, was heißt das denn im Sinne von: Da kommen jetzt Änderungen auf uns zu. Sachen, die wir vielleicht in der klassischen IT-Entwicklung gemacht haben, funktionieren jetzt mit einer No Code/ Low Code-Plattform ganz anders, wie das Beispiel was ich vorhin gebracht habe mit Unternehmen, die sagen: "Jede Applikation, die ausgerollt wird, muss durch den Betriebsrat genehmigt werden." Ist für eine No Code/ Low Code-Plattform mit Citizen Developern ein Todesstoß an der Stelle, wenn ein Betriebsrat da nicht sauber abgeholt wird, weil, das kann nicht funktionieren. Die muss ich auf alle Fälle mit reinnehmen und auch meine Supportstruktur muss mit eingebunden werden, weil jetzt eine Plattform auszurollen, heißt halt nicht nur: "Wir rollen ein neues Office Paket aus." Das heißt: "Hier sind die zehn neuen FAQs, die die Supportorganisation beinhalten muss." Sondern ich habe hier auf einmal eine riesige Entwicklungsgruppe sitzen, die sich dem Thema IT auseinandersetzt, die ganz andere Fragen haben werden als der klassische Anwendungsnutzer. Also, der anruft und sagt: "Hier mein Excel spinnt. Könnt ihr mir das mal neu installieren?" Du wirst auf einmal auch User haben, die dann Entwicklungsfragen stellen. Wie baue ich eine Canvas-App? Wie baue ich eine Model Driven-App? Wie komme ich an diesen Connector ran? Das heißt, entweder baue ich eine komplett separate Supportorganisation auf, die sich dann um diese Themen auch dediziert kümmern kann. Oder ich bohre entsprechend meine bestehende Supportorganisation so weit auf, dass die diesen zusätzlichen Demand auch handeln können. Und wie gesagt, dafür haben wir ja in der Implementierung einen komplett eigenen Stream, der nichts andere macht, als eine Supportorganisationen aufzuschauen und zu integrieren, die ich ja so in einem klassischen Entwicklungsprojekt nicht habe. Also da nutze ich meine existierende Supportorganisation und sage halt: "Ok, statt Siebel benutzt du jetzt heute Dynamics." Ich trainiere die Leute, um damit sie Anwenderfragen beantworten können. Aber eine Plattform ist ja ein ganzes anderes Piece, was ich da vorstelle. Dementsprechend muss ich auch ganz andere Streams und ganz andere Teams auf Kundenseite mit einbeziehen, die ich in einem klassischen Projekt nicht habe. Es verlässt halt die klassische IT. Die Plattform betrifft am Ende des Tages im Zweifelsfall jeden Einzelnen.
56	[0:39:08.7] <b>I: RS</b> Du hast jetzt quasi schon bisschen angedeutet, das fängt ja meistens in einer kleinen Abteilung bei vielen Unternehmen an, wo die diese Plattform erstmalig implementieren. Quasi die Prototypenabteilung und irgendwann wird das aber ja hoffentlich in das ganze Unternehmen skaliert. Wie gestaltest du die Skalierung in das Unternehmen hinein, dann später?

- 57 [0:39:33.4] **B: FG** Da kommt eben unser Center of Excellence ins Spiel. Dass wir sagen, wir schaffen jetzt hier die saubere Grundlage zum einen von den Prozessen her, damit vom Prozess her klar ist, ein User möchte eine Applikation bauen, das sind die Schritte, die er tun muss. Und das funktioniert mit zehn Usern, das funktioniert mit hunderttausend Usern. Dass der Prozess sauber etabliert ist und zum anderen aber auch vom Thema Plattform enablement. Dass die richtigen Umgebungen zur Verfügung gestellt werden, dass das Thema Lizenzmanagement adressiert ist, dass das Thema Monitoring und Reaction sauber definiert ist, wenn ein User das und das macht, was er nicht tun sollte, was passiert dann? Geht dann erst mal eine Mail nur an den Admin raus oder schreitet die Plattform selber ein und unterbindet gewissermaßen Aktivitäten. Die Grundlagen müssen halt sauber definiert werden und dann skaliert so ein Center of Excellence auch sauber oder eine Plattformeinführung. Und deswegen ist halt wichtig und dafür haben wir dedizierte Angebote, die den Kunden eben genau dabei unterstützen, diese Grundlage zu schaffen, damit Skalierung eben hinterher funktioniert.
- 58 [0:40:36.9] **I: RS** Und wie befähigt ihr dann die Mitarbeiter während dieser Skalierung? Oder schon im Einführungsprojekt direkt?
- 59 [0:40:44.2] **B: FG** Dafür haben wir ebendiesen dedizierten Stream, der sagt: "Wir schlaun jetzt mal Entwickler auf." Das können interne Entwickler sein. Das können aber auch Citizen Developer sein, die dann als Multiplikatoren funktionieren. Also idealerweise identifizieren wir im Vorfeld die Champions beim Kunden. Was wären denn unsere Vorreiter? Klassischerweise die Leute, die die besten Excelsheets bauen, wo die komplexesten Formeln drin sind oder die Leute, die PowerPoints bauen und Animationen und allen Features, die wirklich die Tools selber ausreizen. Das sind die Champions, die wir initial identifizieren möchten, wo wir sagen: "Ok, wir geben euch jetzt mal ein Bootcamp für Power Platform, wie baut ihr denn Applikationen. Und ihr tragt dieses Wissen dann in eure Teams weiter. Dass ihr dann entsprechend die zweite Generation, dritte Generation an Citizen Developern ausbilden könnt." Und auf der anderen Seite auch als Champions in der Community agiert, die da sagen: "Wir stellen für euch die entsprechenden Informationen zusammen. Wir hosten alle zwei Monate einen Hackathon, wo wir gemeinsam mit den restlichen Mitarbeitern Applikationen bauen, die dann auch solche Walk-in-Clinics zum Beispiel betreiben, dass man sagt: "Jeden Donnerstag zwischen 12 und 14 Uhr ist Sprechstunde. Alle Citizen Developer können sich in ein Teammeeting einwählen und die Champions sind da, um Rede und Antwort zu stehen." Also wir arbeiten da viel mit Multiplikatoren, weil, du wirst nie ein Enduser-Training für ein ganzes Unternehmen organisieren können. Ist auch nicht erforderlich, weil nicht jeder User wird zum Citizen Developer werden an der Stelle. Und, wie gesagt, wir versuchen da den sauberen Grundstein zu legen mit den Champions. Die zu befähigen, damit sie es entsprechend weitertragen können und auch als Vorbilder agieren können.
- 60 [0:42:24.7] **I: RS** Du hast vorhin auch Communitys und Support durch die IT angesprochen. Wie spielt das damit rein?
- 61 [0:42:31.4] **B: FG** Die Communitys sind insofern wichtig, weil sie die erste Anlaufstelle für die User sein sollten. Dass eben ein User, der eine Frage hat oder Probleme hat, nicht als erstes zum Telefon greift und den internen Support anruft, die ihm im Zweifelsfall sowieso nicht helfen können, sondern dass es wirklich interne Communitys gibt oder externe Communitys auch, in denen dann Fragen beantwortet werden können, von einem Citizen Developer zum anderen. Wenn es um dedizierte Implementierung geht, wenn es einfach um Best Practices geht. Oder auch neue Leute, die fragen: "Wie fange ich denn am besten an? Was wären denn für mich die besten Einstiegspunkte? Was sind denn so ein paar Einsteigertipps, zum Beispiel?" Und die sind in der Regel selbst organisiert. Das heißt, die werden initial zur Verfügung gestellt von einer Plattform her. Die Inhalte kommen dann aber durch die User selber und können im Zweifelsfall auch durch die interne IT unterstützt werden. Das heißt, dass die

zum Beispiel als Moderatoren agieren oder auch als Champions. Die sagen, als Teil meines Developerjobs baue ich auch ab und zu einfach mal No Code/ Low Code-Applikationen und stelle die zur Verfügung. Packe die auch in einen entsprechenden Applikationskatalog und hoste dann auch mal eine Session, wo ich zeige, wie habe ich denn diese Applikation tatsächlich gebaut, um neuen Citizen Developern oder auch bestehenden Citizen Developern einfach neue Techniken beizubringen.

- 62 [0:43:47.7] **I: RS** Du hattest auch ganz kurz, also ich habe das wie eine Art Helpdesk durch die IT aufgefasst. Sind das dann trotzdem auch Ansprechpartner, die bei Fragen ansprechbar sind? Oder sind das nur die Champions?
- 63 [0:44:02.6] **B: FG** Das kommt auf die Unternehmensgröße und die Struktur darauf an, würde ich sagen, ob ich tatsächlich genügend Leute habe in meiner internen IT oder andersherum, habe ich tatsächlich genügend Champions, dass sich das Thema quasi selbst organisiert? Dass die User sich untereinander selber helfen, oder ist es von der Kultur eher so, es wird alles von der internen IT vorangetriebene. Die agieren als die Ansprechpartner und die User konsumieren nur an dieser Geschichte. Das ist so ein bisschen Unternehmenskultur, was dahintersteckt. Wir sehen beides. Also manche Unternehmen, wo die interne IT am Anfang das Forum zur Verfügung stellt und dann zehn Champions nominiert und die schmeißen den Laden dann und holen entsprechend die User ran. Wir haben aber auch Kunden, wo wirklich ein zentrales IT-Forum existiert. Die dann wie ein verlängerter Arm funktionieren, die dann Citizen Developern Fragen beantworten. Ist natürlich nicht ideal, weil, dadurch wird die IT schnell wieder zum Flaschenhals und wenn die Fragen nicht rechtzeitig beantwortet werden können, sinkt natürlich die Motivation auf Seiten der Citizen Developer. Also wir sehen in den Unternehmen, wo wirklich da eine starke Community ist, die sich selber mit Wissen versorgt, da ist die Adaption von der Plattform auch wesentlich höher, weil den Leuten halt geholfen wird, weil da viel eher ein Austausch stattfindet als jetzt bei den Unternehmen, wo es klassisch durch die interne IT verwaltet wird.
- 64 [0:45:27.4] **I: RS** Bei so einem Projekt gibt es ja ziemlich viele Gefahren, dass etwas schief läuft. Wie sicherst du die Qualität des Projekts?
- 65 [0:45:42.4] **B: FG** Ist jetzt auch unabhängig von Power Platform-Projekten, würde ich sagen. Wir haben regelmäßige Statuschecks. Wir haben auch die entsprechenden Quality Gates, wo wir sagen: "Ok, das und das sind jetzt eigentlich die Punkte, die wir adressiert haben sollen." Regelmäßige Feedbackrunden mit dem Kunden: Ist das jetzt auch verstanden? Sind die entsprechenden Arbeitspakete auch implementiert? Haben wir unsere Checkliste, die wir am Anfang setzen, haben wir die tatsächlich abgearbeitet? Sehen wir, dass der Kunde jetzt in der Lage ist, diese Plattform selber zu betreiben? Oder sehen wir, die haben zwar in den Workshops mit dringesessen aber sich nur, wie soll ich sagen, bespaßen lassen, aber es nicht wirklich verinnerlicht, worauf es tatsächlich ankommt. Und das ist halt eben wichtig. Gerade so in der Anfangsphase, wo wir viel Wissen mitbringen und das auch an den Kunden übergeben, dass das auch wirklich verinnerlicht wird. Dass der Kunde auch in der Lage ist, dieses Wissen dann zu transportieren und für sich umzusetzen. Und dafür muss man regelmäßige Statuschecks auch einfach machen und auch mal Rückfragen stellen. Nach dem Motto: "Zeigt mir mal, was ihr jetzt mitgenommen habt. Wie habt ihr das jetzt implementiert?" Dass wir auch konstantes Review machen auf die Themen, die dann vom Kunden implementiert werden. Um zu sehen, folgen die auch wirklich den Best Practices oder hat man hier tatsächlich noch irgendwas übersehen. So ein Beispiel wäre eine Environmentstrategie zum Beispiel. Wie muss ich denn jetzt meine Environments aufsetzen. Da haben wir verschiedene Szenarien, die wir mit dem Kunden durchspielen. Der Kunde nimmt das dann mit, kommt mit einem eigenen Konzept wieder und wir führen dann ein entsprechendes Review durch. Zum einen auf der konzeptionellen Ebene und dann, wenn es auch tatsächlich implementiert ist, dass wir auch da noch mal drauf schauen. Sind wirklich die richtigen Drähte miteinander

verdrahtet. Sind da noch irgendwo Schlupflöcher offen, die dann wieder zu Problemen führen könnten.

- 66 [0:47:26.8] **I: RS** Du hast es schon leicht angeschnitten. Thema Erfolgsmessung. Woran macht ihr den Erfolg eines Projekts fest? Einerseits, dass die Plattform genutzt wird, hast du ja gerade schon gesagt. Gibt es noch andere Faktoren?
- 67 [0:47:43.4] **B: FG** Ja, es gibt verschiedene KPI's, die wir nutzen. Zum einen, wie viele Applikationen werden denn überhaupt gebaut? Werden die dann tatsächlich auch genutzt? Weil, wir sehen auch bei manchen Kunden, da werden am Anfang da tausende Applikationen gebaut, die dann nach drei Monaten noch als Zombies rum agieren, die aber keinen produktiven Einsatz haben. Wir messen zum einen, wie viel Applikationen gibt es denn? Mit wie vielen Usern sind die geteilt? Wie aktiv werden die genutzt? Wie viele Aktionen werden generiert darüber? Dann in den Communitys selber, wie viele Mitglieder hat denn meine Community? Wie viele Postings gibts denn dann? Wie sieht eine Antwortfrequenz aus, wenn eine Frage gestellt wird? Kommt da innerhalb von einer Woche eine Antwort, kommen da zehn Antworten zurück? Also, es gibt schon verschiedene Indikatoren, die wir auch aus der Plattform selber auslesen können. Direkt, bevor wir irgendwelche User mal befragen müssen, inwieweit ist denn die Plattform angekommen und natürlich auch das Aufkommen von Supporttickets, um zu sehen, funktioniert auch ein Supporthandling sauber. Rufen die Leute tatsächlich alle beim Helpdesk an oder organisieren sie sich wirklich in der Community, die dafür zur Verfügung gestellt wurde? Wie viele Leute partizipieren an einem Hackathon, wenn wir ihn zur Verfügung stellen. Oder in einer Walk-in-Clinic, wie viele Leute kommen tatsächlich und haben Fragen oder haben Lösungen? Und dann später, wenn die Plattform etwas gereift ist bei einem Kunden. Was wir normalerweise auch etablieren, ist ein Applikationskatalog, in dem dann Applikationen, die von Usern gebaut werden, auch anderen Usern zur Verfügung gestellt werden, also ein App-Store und auch ein Store für einzelne Komponenten. Und auch daran kann ich sehen, wie ist denn die Adaptation von meiner Plattform? Habe ich da nach sechs Monaten, sind da zwei Applikationen in meinem App-Store oder sind da zweihundert drin, die die Leute gebaut haben und auch die einzelnen Komponenten. Wie viele sind das? Wie ist die Qualität von den Komponenten? Es gibt schon relativ viele KPI's, die man abfragen kann. Eben nicht nur, wie viele User haben denn eine Lizenz bekommen, zum Beispiel.
- 68 [0:49:37.4] **I: RS** Bloß für das Transkript eine Rückfrage. Kannst Du noch mal kurz definieren, was ein Hackathon ist?
- 69 [0:49:44.1] **B: FG** Ein Hackathon ist für uns, dass wir sagen: "Wir organisieren jetzt mal ein - wie beschreibe ich das - wie einen Flashmob." Also ich stelle den zur Verfügung. Wir kommen mit einem Team, fünf, sechs Leute, mit Entwicklern, mit SME's und geben dann quasi ein offenes Mikrofon für die Leute, die sagen: "Ok ich habe da eine Idee, die würde ich gerne umsetzen. Können wir das heute zusammen machen?" Und dann werden entsprechende Teams geformt, je nachdem, wie viele Leute zur Verfügung stehen, um dann eben diese Ideen innerhalb von einem Tag umzusetzen und zu einem Prototyp-Level zu bringen. Also, wenn jetzt ein User sagt: "Ich hätte hier eine Idee für einen internen Prozess, ich schaff das aber alleine nicht." Dann wird eben ein Team geformt, die dann gemeinsam anfangen, diese Applikation zu designen, zu implementieren, die dann durch unsere Techniker und SME's unterstützt wird im Laufe des Tages und am Ende des Tages gibt es dann eine Abschlusspräsentation, wo dann jedes Team entsprechend seinen Prototyp vorstellt, was sie denn heute im Laufe des Tages gebaut haben. Das ist halt eine super Gelegenheit, um zum einen Wissen zu vermitteln, aber auch um einfach Netzwerke zu bilden, dass sich Leute mal austauschen können, dass Ideen ausgetauscht werden, so diese gegenseitige Befruchtung. Also das Team A, die hatten da eine super Idee, die könnten wir in unserer Abteilung auch gebrauchen. Da hat noch nie einer dran gedacht. Ich bau das jetzt auch und dafür vernetze ich mich mal mit den Jungs aus Team A, die das da jetzt im Hackathon schonmal vorgebaut



haben. Also das ist so der Sinn von einem Hackathon, um Wissen zu vermitteln und auch die Community voranzubringen, um zu zeigen, ihr seid da nicht alleine als Citizen Developer, da gibt es unheimlich viele Leute, die das auch gerne machen. Und das ist halt auch so eine KPI, wo ich sehen kann: Wie viele Leute melden sich denn an für so einen Hackathon? Sind das Zehn? Sind das nur meine zehn Champions, die ich vielleicht habe oder kommen da zweihundert Leute, die im Zweifelsfall nur zugucken wollen. Die mal sehen: "Hier ich könnte mir das gut vorstellen. Kann ich in so einem Team einfach mal zwei bis drei Stunden zugucken, wie baut ihr denn eine App?"

- 70 [0:51:41.0] **I: RS** Dann sind wir jetzt quasi auch schon am Ende von unserem Interview. Mich würde jetzt noch interessieren: Hast du Anmerkungen zum Interview? Best Practices to share? Oder Sachen, die jetzt du persönlich als wichtig empfindest, die ich aber jetzt vielleicht nicht im Interview gefragt habe?
- 71 [0:51:55.8] **B: FG** Also, ich sage mal, du hast ja eine Frage gestellt zur Erwartungshaltung, die ein Kunde hat. Das ist halt sehr wichtig, dass der Kunde wirklich weiß, worauf er sich denn da einlässt. Wie gesagt, eine No Code/ Low Code-Plattform ist eben nicht nur: Ich führe jetzt ein neues Tool ein ins System, sondern das ist schon eine, das geht schon ziemlich an die Basis von einem Unternehmen und das unterschätzen eben viele Kunden. Und wenn sie sich dann darauf Committen, dass sie wirklich sich darauf fokussieren, wie kriege ich denn jetzt die Technik zum Laufen. Weil es in der Regel von der IT mal vorangetrieben wird, so ein Thema, und dann halt auch so mit Scheuklappen draufgeguckt wird: Ich muss nur die Technik ans Laufen kriegen. Der Rest passiert dann schon irgendwie. Das ist halt gefährlich, so eine Geschichte. Also da muss ich auch meine Kunden immer wieder drauf hinweisen: Ihr dürft es nicht nur auf die Technik reduzieren. Die Technik macht maximal 25 % aus von der ganzen Geschichte. Der Rest ist, wie eure Unternehmensstruktur dieses Thema absorbieren kann und tatsächlich umsetzen kann. Wenn ihr nur die Technik zum Laufen kriegt, werdet ihr an den anderen Themen scheitern, wenn ihr das so unterschätzt. Das würde ich so mitgeben. Das ist einfach wichtig, so als Best Practice und da ruhig auch drauf rumreiten. So als Mantra: Jungs, es ist nicht nur die Technik! Power Platform betrifft alles. Power Platform rennt Silos ein, weil, ich kann auch nicht mehr alleine für mich alles entscheiden, wenn es zum Beispiel um Thema Policies für Connectoren geht. Mit welchen Datenstrukturen darf ich denn kommunizieren? Betrifft alles, was auf der Plattform stattfindet und eben nicht nur das Team das jetzt Connectoren baut, zum Beispiel und solche Sachen.
- 72 [0:53:38.7] **I: RS** Vielen Dank für das Interview und ich würde dann jetzt die Aufnahmen beenden.
- 73 [0:53:42.9] **Ende**

## Einwilligungserklärung Interview

Ich erkläre hiermit mein Einverständnis zur Nutzung der personenbezogenen Daten, die im Rahmen des folgenden Gesprächs erhoben wurden:

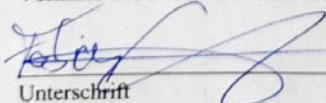
Datum des Interviews	15.07.2022
Namen der zu interviewenden Person	Fabian Gackstätter
Name der interviewenden Person	Rocco Schenk
Kurzbeschreibung des Forschungsprojekts	Einführung von No Code/ Low Code aus Sicht des Projektmanagements
Durchführende Hochschule	Hochschule Mittweida – University of Applied Science

Die Daten werden im Rahmen eines mündlichen Gesprächs erhoben, das mit einem Aufnahmegerät und Videoaufnahmegerät aufgezeichnet wurde. Zum Zwecke der Datenanalyse werden die mündlich erhobenen Daten verschriftlicht (Transkription), wobei die Daten nicht anonymisiert werden.

Vorhandene Kontaktdaten, der interviewten Person, werden aus Dokumentationsgründen in einem separaten Schriftstück lediglich den Gutachter\*innen der wissenschaftlichen Ausarbeitung zur Verfügung gestellt. Nach dem Abschluss des Projekts werden diese Daten gelöscht.

Der Speicherung der personenbezogenen Daten zu Dokumentationszwecken kann durch die interviewte Person jederzeit widersprochen werden. Die Teilnahme an dem Gespräch erfolgt freiwillig. Das Gespräch kann zu jedem Zeitpunkt abgebrochen werden. Das Einverständnis zur Aufzeichnung und Weiterverwendung der Daten kann jederzeit widerrufen werden.

Fabian Gackstätter  
Vorname und Name in Druckbuchstaben

  
Unterschrift

14.07.22, Atzelgitz  
Datum, Ort

## Anlagen 5, Interview EB04

RS: Rocco Schenk

SG: Simone Göttl

1	[0:00:00.0] <b>Start (SG - Simone Göttl/ RS - Rocco Schenk)</b>
2	[0:00:09.4] <b>I: RS</b> Dann hat die Aufnahme jetzt begonnen. Ich begrüße Sie ganz herzlich. Möchte Sie fragen: Sind Sie mit der Aufnahme einverstanden?
3	[0:00:17.1] <b>B: SG</b> Ja, das passt für mich.
4	[0:00:19.1] <b>I: RS</b> Zu Ihren personenbezogenen Daten. Ich werde das Interview am Ende transkribieren und das kommt dann in die Anlage meiner Masterarbeit. Jegliche personenbezogenen Daten, also an Kontaktdaten, die ich von Ihnen habe, lösche ich. Sie werden aber nicht anonymisiert innerhalb des Interviews. Sind Sie damit auch einverstanden?
5	[0:00:39.2] <b>B: SG</b> Das passt für mich.
6	[0:00:41.1] <b>I: RS</b> Sehr schön, dann würde ich gerne damit einsteigen, dass Sie sich mal ganz kurz vorstellen, wie so Ihre Erfahrungen sind? Also auch im Projektmanagementbereich, auch mit Low Code vielleicht in Verbindung.
7	[0:00:51.9] <b>B: SG</b> Gut, also mein Name ist Simone Göttl. Ich bin Unternehmensberaterin. mache das schon seit mittlerweile fast 13 Jahren fest und habe mich die letzten Jahre auf New Work-Themen fokussiert. New Work dann insbesondere im Kontext des digitalen Arbeitsplatzes, der dort ist. Was sich dann einerseits auszeichnet durch die Einführung von Kollaborationsplattformen, andererseits durch die Einführung und den Ausbau und Rollout von No Code/ Low Code-Plattformen. Und das mache ich insbesondere in der Finanzindustrie, also bei Banken und Versicherungen und habe da jetzt die letzten anderthalb, zwei Jahre umfangreiche Expertise erhalten. Muss man natürlich auch sagen, durch Corona ein bisschen befördert.
8	[0:01:31.1] <b>I: RS</b> Stimmt, das war wahrscheinlich einer der ultimativen Kickstarter für No Code/ Low Code. Zum Anfang, als leichten Einstieg. Wie würden Sie denn No Code/ Low Code definieren? Jetzt aus Ihrem Standpunkt her gesehen.
9	[0:01:47.3] <b>B: SG</b> No Code/ Low Code, für mich ist das ein Bausteinsystem. Das heißt, ein Bausteinkastensystem, wo ich auch, also ich als Endnutzer, also wirklich auch kein ITler, sondern ein Endanwender, ein Fachbereichsmitarbeiter, jemand, der auch keine IT-Ausbildung hat, sehr schnell und sehr einfach Anwendungen und Applikationen zusammenbauen kann, ohne Vorwissen zu haben, also für mich ist es ein Bausteinkastenprinzip, was ich schnell nutzen kann für meine Bedürfnisse.
10	[0:02:15.2] <b>I: RS</b> Sind Sie da schon in Berührung mit dem Konzept Citizen Development gekommen?
11	[0:02:20.7] <b>B: SG</b> Ja, Citizen Developer ist ja dann die Person, also gerade was ich gesagt habe, aus meiner Sicht, wo ich jetzt unterwegs bin, bei Banken und Versicherungen, der Fachbereichsanwender oder der Non-ITler, der quasi am Ende sagt: "Ich finde das eigentlich ganz cool und ich möchte mir mal selber helfen und brauche ein, ich nenne es jetzt wirklich, Baukastenprinzip, um mir selber Anwendungen, Apps, Applikationen, Workflows

zusammenstellen zu können." Also ja, schon in Berührung gekommen, ist aber an vielen Punkten auch noch ein bisschen, den Begriff muss man noch mal erklären, wenn jemand sich noch nicht darauf beschränkt hat. Das habe ich auch schon vielfach gelernt.

- 12 [0:02:58.9] **I: RS** Ok, Sie haben ja jetzt schon einen Vorteil genannt. Können sie noch weitere Vorteile nennen von No Code/ Low Code und von Citizen Development?
- 13 [0:03:11.1] **B: SG** Also, wie gesagt die Einfachheit, also das es auch Nicht-Programmierer nutzen können und das ist natürlich dann ein großer Vorteil für eine IT, also gerade bei großen Konzernen oder bei Banken und Versicherungen. Dass IT-Entwicklungen häufig langwierig dauern, zu lang für einen Endnutzer, dass es auch sehr kostspielig ist, plus, das sehen wir ja auch alle, dass gerade Entwickler oder Cloud-Entwickler auch nicht immer verfügbar sind und der Markt ziemlich leergefegt ist von diesen Fachkräften. Das heißt, es besteht auch häufig einfach gar keine Möglichkeit mehr, so schnell auf Anwendungen, Apps als IT reagieren zu können und deshalb hat sich dieser Markt so entwickelt. Das ist definitiv dort auch ein großer Vorteil für No Code/ Low Code, um das lösen zu können.
- 14 [0:03:59.4] **I: RS** Sie haben ja jetzt schon die Fachkräfte angesprochen. Können Sie mal kurz sagen, aus Ihrer Perspektive, es gibt in der IT-Branche ja einen Fachkräftemangel und auch in den IT-Abteilungen, können No Code/ Low Code da Abhilfe schaffen, können die das beheben? Also wie ist das aus Ihrer Perspektive?
- 15 [0:04:22.3] **B: SG** Also, ich glaube man muss ein bisschen differenzieren, was man mit No Code/ Low Code nicht machen kann, ich sage mal, ein SAP-System oder andere im Bankensystem, Kernbankensystem abzulösen. Wenn es aber darum geht, sich einerseits Helferlein im Alltag zu bauen, vielleicht auch, um Sachen zu automatisieren, die man tagtäglich macht oder um fachbereichsspezifische Prozesse geht, dann ist das, glaube ich schon etwas, wo die Endentwickler auch Vorteile erhalten können. Das heißt, dafür eignet sich es definitiv und ein Vorteil ist natürlich dort dann auch, also jeder, der schon mal ein IT-Projekt gemacht hat, ein ganz klassisches, man schreibt ganz viele Anforderungen, gibt die an die IT und bekommt aber nicht das zurück, was man sich eigentlich gewünscht hat. Wenn ich aber jetzt ein Fachbereich oder selber die Person bin, die die Anforderung definiert und es umsetze, dann geht es auch schneller und es ist am Ende auch besser, weil es zu meinen Anforderungen passt.
- 16 [0:05:16.4] **I: RS** Jetzt haben Sie ja schon sehr viele Vorteile natürlich gesagt, aber das wird ja wahrscheinlich auch Nachteile haben. Können Sie die vielleicht mal wiedergeben, wie die aus Ihrer Perspektive sind?
- 17 [0:05:26.0] **B: SG** Ich bin eigentlich kein großer Fan vom Nachteile finden, ist immer schwierig. Was ein bisschen Hürden sind, ist natürlich, dort die Leitplanken innerhalb eines Unternehmens zu setzen. Das heißt, einerseits sicherzustellen, wir hatten das vorhin auch, Datenschutz, dass das erfüllt ist. Zugriffsberechtigungen sicherzustellen, wer darf welche Daten sehen? Wie kann ich aber auch diese No Code/ Low Code-Plattform an die bestehenden Bestandssysteme abbilden? Wo brauche ich APIs? Und natürlich auch die ganze Hürde, was ist eigentlich mein Rahmenkonstrukt? Meine Governance? Was erlaube ich überhaupt, um das machen zu können? Das sind natürlich Punkte, das sind jetzt keine Nachteile, aber das sind Hürden. Was aber auch ein Nachteil sein kann und deshalb, wenn so eine Plattform nicht gemanagt ist, No Code/ Low Code-Plattform ist, dass man immer mehr Schatten-IT bekommt. Das heißt, wenn ich nicht als IT zum Beispiel schaue, welche Apps werden entwickelt und die auch einem Lebenszyklus unterlasse, habe ich irgendwann ein Riesenwildwuchs und das ist definitiv ein Nachteil und ich laufe natürlich auch Gefahr, das ist auch so, dass ich an der einen oder anderen Stelle, wenn der derjenige, der das als Fachbereich entwickelt, nicht richtig geschult ist oder nicht richtig ausgebildet ist, dass es am Ende doch länger dauert.

- 18 [0:06:44.8] **I: RS** Das Schulungsproblem, sehen Sie das als Teil der Governance oder sagen Sie: "Nein, das ist tatsächlich auch mehr ausgegliedert?"
- 19 [0:06:56.2] **B: SG** Also es gibt verschiedene Formate. Es gibt dann so Center of Excellence, wo man so was verankern kann. Das heißt, einerseits die Governance verankert, also Prozesse, was darf ich, was darf ich nicht? Was ist erlaubt? Wie erfolgen die Prozesse für Freigaben? Welche Apps darf ich überhaupt programmieren? Da findet man häufig eine gewisse Expertise, dass man sagt: "Man hat so Leuchtturmmitarbeiter, die da natürlich auch helfen können." Nichtsdestotrotz würde ich bei einem Rollout immer die Mitarbeiter mitnehmen. Das heißt, deshalb sage ich auch bewusst mitnehmen, für mich ist das nicht eine einmalige Schulung, sondern für mich ist das eine fortwährende Enablement, Befähigung, was durch Trainings passieren kann, durch einzelne Learningnuggets, YouTube-Videos, aber auch vielleicht durch Communitys. Und deshalb ist es nicht nur Aufgabe des Center of Excellence, sondern muss man einfach auch schauen, wer am Ende auch der Product Owner ist und dass der sein Produkt natürlich durch so einen Konzern und Haus trägt.
- 20 [0:07:50.6] **I: RS** Wir haben ja jetzt schon viel darüber gesprochen, welche guten Aspekte No Code/ Low Code, Citizen Development hat. Wir haben auch darüber gesprochen, worauf man aufpassen sollte. Aus heutiger Sicht, wirklich Stand heute gesehen, was ist aus Ihrer Sicht eine realistische Erwartung, die ich an eine No Code/ Low Code-Plattform wirklich stellen kann, als Unternehmen, wenn ich das jetzt einführen möchte?
- 21 [0:08:13.1] **B: SG** Eine Erwartungshaltung ist, also wenn ich aus der Erfahrung spreche, die Hersteller, ich will nicht sagen, dass sie immer mehr versprechen, als am Ende möglich ist, aber es ist ein Werbeversprechen, was teilweise einfacher klingt, als es wirklich ist. Was ich viel in der Praxis sehe ist, dass es dann doch nicht so einfach ist. Und es gibt auch Hersteller, die bieten App in a Day an. Ja, ich kann eine App in einem Tag bauen, das heißt aber nicht, dass ich jeden Use Case in einem Tag bauen kann. Und das heißt, es ist ganz klar, was ich erwarten muss, es ist ein Erwartungsmanagement und das, was mir vielleicht versprochen wird, sollte ich gründlich hinterfragen, im Sinne, ist es wirklich so einfach und so leicht und sollte ich mich tiefergehend damit auseinandersetzen. Das heißt, auch wirklich Prototypen bauen oder Proof of Concept Phase machen und dort verschiedene Fachbereiche mit einbeziehen. Und nicht nur auf so App in a Day, was so ein vorgefertigter Tag ist, alles total einfach klingt. Also ich bleibe jetzt mal bei einem großen Möbelhersteller, wo wir das alle kennen. Das klingt immer so schön und so toll und so schön und dann sitzt man dann doch zu Hause und stellt fest: "Wo ist denn diese verdammte eine Schraube hin?"
- 22 [0:09:23.9] **I: RS** Alle weiteren Fragen erfolgen von jetzt ab immer unter der Prämisse, dass ein Unternehmen No Code/ Low Code gerne bei sich einführen möchte und das ist quasi immer unsere Ausgangssituation. Sie haben ja jetzt schon angeschnitten, dass es gewisse Voraussetzungen gibt, u.a. dass man halt nicht an einem Tag quasi die Mitarbeiter geschult hat und die das dann sofort können. Sehen Sie noch andere Voraussetzungen, die ein Unternehmen vielleicht mitbringen muss, abgesehen von einer funktionierenden Governance, die förderlich sind oder die auf jeden Fall da sein müssen, damit es erfolgreich eingeführt werden kann? Dabei denke ich so an Themen wie Mindset, wie Unternehmensgröße, braucht es ein bestimmtes Kapital oder Manpower eventuell? Also wie sind da Ihre Erfahrungen?
- 23 [0:10:24.4] **B: SG** Also ich habe das schon bei Unternehmen oder Banken unterschiedlichster Art gesehen. Von: Bank hat 800 Mitarbeiter oder hat 10000 Mitarbeiter, weil, man kann es für beide einführen, das funktioniert. Ich habe es auch schon gesehen von der Zielgruppe her, dass es Vereine genutzt haben, um ihre Mitgliedsanträge zu verwalten und das effizienter zu gestalten. So, das zeigt halt eigentlich, dass das Leistungsspektrum auch schon ziemlich groß ist, also von klein bis groß. Was es aber bedarf ist, wie Sie es gesagt haben, auch der Mindset, aber das Mindset, glaube ich, kommt von 2 Seiten. Einerseits von der Belegschaft selber, dass man dort auch wirklich eine Akzeptanz fördert und dort auch deutlich

macht, was die Vorteile sind oder in gewisser Weise, man wird nicht alle überzeugen, gerade bei großen Unternehmen, sondern ein paar Leuchttürme hat, die dafür stehen und die anderen mitziehen. Das Zweite ist aber, was ganz wichtig ist, das Commitment von oben. Was meine ich damit? Es muss natürlich der Vorstand bzw. ein IT-Vorstand auch für diese Transformation stehen und auch sagen, das ist etwas, was wir jetzt machen und es darf ihm nicht aufdoktriniert sein. Das heißt, am besten, was heute gut funktioniert und was ich schon gesehen habe, ist das einzubetten in eine digitale Agenda oder in eine grundsätzliche Digitalisierungsstrategie. Und wenn es da mit drin ist und verankert ist, dann funktioniert das auch gut.

- 24 [0:11:47.1] **I: RS** Mit grundsätzlicher Digitalisierungsstrategie meinen Sie, dass generell im Unternehmen digitalisiert wird und das quasi als ein Aspekt mit gefördert wird?
- 25 [0:11:57.4] **B: SG** Genau. Also es gibt, manche nennen es die IT-Strategie, manche nennen es digitale Transformationsstrategie, und da sind dann halt Punkte drin. Ganz klar, was wir häufig sehen, gibt es eine Cloudstrategie. Möchte man mit Systemen in die Cloud gehen, aber auch möchten wir weiterhin bei einzelnen Häusern, die ich gesehen habe, steht das da explizit drin. Wir verwenden No Code/ Low Code oder man formuliert es anders: "Wir helfen dem Fachbereich, sich selbst zu helfen, ohne Entwickler zu werden." Also das ist schon, dass es dort verankert ist und das ist ja eine Marschroute und da sollte es auf alle Fälle auch verankert sein, dass das Commitment von oben da ist.
- 26 [0:12:33.7] **I: RS** Wir haben ja jetzt schon über Voraussetzungen gesprochen. Sehen Sie auch Problemstellungen oder, wie bei den Nachteilen, Hürden, welche Unternehmen beachten sollten, wenn sie jetzt in der Einführungsphase sind von einer No Code/ Low Code-Plattform, auf die Sie da explizit achten sollten? Wo häufig drüber gestolpert wird, zum Beispiel?
- 27 [0:12:56.3] **B: SG** Also häufiges Thema ist natürlich, was darf ich in dieser Plattform? Das heißt eigentlich auch, welche Funktionen schalte ich frei an dem Punkt und wie löse ich diesen Sicherheitsaspekt. Was meine ich, zum Beispiel, damit. Also bei Teams, wo wir jetzt drüber reden, da gibt es ja zum Beispiel so eine Funktion, die heißt Transkript, und wenn man die anstellt, dann können Sie meinen Wortlaut, wie so ein bisschen im Fernsehen, mitlezen. Das ist ein total cooles Feature, ist aber aus Datenschutzgesichtspunkten total fragwürdig, weil die Daten, zumindest ist so mein letzter Stand, dort häufig halt immer nach Amerika gehen. Das heißt, das wäre aus EU-Sicht für einen Datenschutzkunden gar nicht zulässig. Das heißt, es gibt aber auch in diesen No Code/ Low Code- Plattformen Funktionalitäten, die vielleicht, aus welchen Gründen auch immer, nicht in Europa gespeichert werden. Das muss man sich anschauen. Plus, was ein zweiter Punkt ist, man muss natürlich auch die ganzen Security-Aspekte mit betrachten. Hat jetzt ein Hersteller Zugriff auf meine Daten, weil ich im Vertrag stehen habe, dass ich einen Wartungsvertrag habe, dass er Zugriff haben kann und dann ist natürlich die Frage, was sind das für Daten, die ich hier in dieser No Code/ Low Code-Plattform speichere. Sind das öffentlich verfügbare Informationen? Also, ich sage mal so, Informationen, die normalerweise auch auf einer Website wären oder in der Presse, ist das kein Problem. Ist es aber etwas, was gewissen Vertraulichkeitsgesichtspunkten unterliegt, habe ich natürlich schon eine Herausforderung zu schauen, wie schütze ich die und was wende ich für Security-Maßnahmen an, damit die nicht einfach abfließen können, und sei es nur der Hersteller, der sich das anschaut. Und das ist schon ein wichtiger Punkt, wo gerade Banken und Versicherungen natürlich auch viel mit draufschauen. Also für welche Daten gebe ich das überhaupt frei und auch die Informationssicherheit, die da ein großes Auge darauf hat. Und ich habe dann halt die Möglichkeit, das zu lösen mit technischen Maßnahmen, dass ich mehr verschlüssele, vielleicht auch gar nicht ablege und Sachen auch nicht zulasse. Oder ich habe natürlich auch immer die Möglichkeit, dass ich den Mitarbeitern in Form von Arbeitsanweisungen und Hilfestellungen gewisse Vorgaben auferlege. Aber das sind halt Punkte, Menschen sind da natürlich nicht so sicher wie die Technik vielleicht an der einen oder anderen Stelle. Und das sind definitiv gerade, die ich sehe, eine der großen

	Hürden und wo man sich natürlich dann auch einfach, wenn man in einem regulierten Umfeld unterwegs ist, aber auch andere Konzerne, natürlich relativ schwer damit tut und ein bisschen vorsichtiger ist.
28	[0:15:27.1] <b>I: RS</b> Sie haben ja jetzt viel über Datenschutz und Cybersecurity gesagt. Sehen Sie noch andere Probleme, was auf ein Unternehmen, das No Code/ Low Code einführt, auf dieses zukommen könnte?
29	[0:15:41.2] <b>B: SG</b> Also die Frage ist auch immer, was ist denn für mich die richtige Plattform. Also was meine ich? Wir haben ja jetzt sehr viel allgemein über No Code/ Low Code-Plattform gesprochen und da gibt es ja verschiedene Hersteller, Microsoft, Mendix, Pega, da gibt es eine Unzahl von und die haben natürlich alle ihre Vor- und Nachteile. Sei es preislich, sei es von den Funktionen, von der Usability und ich bin jetzt bei Kunden auch schon darüber gestolpert. Also ich habe jetzt einen Kunden zum Beispiel gehabt, der auch gesagt hat: "Naja, die Plattform, die ich habe, ist das überhaupt noch die Richtige für mich? Oder bräuchte ich nicht eine andere?" Und das sind natürlich auch Entscheidungen, die man berücksichtigen muss. Welche nutze ich jetzt und aus welchen Gründen und wie gehe ich damit um, wenn die Plattformen sich weiterentwickeln und plötzlich die Plattform, die ich auch gewählt habe, nicht mehr so tauglich ist für meine Punkte. Habe ich dann Exit-Optionen? Lass ich es zu, dass es mehr als eine gibt? Kann ich mir das kostenmäßig leisten? Und das sind natürlich die Punkte, wo auch viele, dieser Auswahlprozess an sich, wenn ich nicht sage, ich habe einen Favoriten, aus welchen Gründen auch immer, das sollte auch gut überlegt sein.
30	[0:16:52.5] <b>I: RS</b> Angenommen, wir haben jetzt die perfekte Plattform gefunden für unser Beispielunternehmen und das möchte die jetzt einführen. Welche Projektorganisationsform würden Sie als am geeignetsten zur Einführung einer solchen Plattform ansehen? Also würden Sie die Mitarbeiter, zum Beispiel, herauslösen aus Ihren Fachabteilungen, die am Projekt beteiligt sind? Oder sagen Sie: "Nein, es ist wichtig, dass die darin bleiben." Wie ist die Hierarchie aufgebaut?
31	[0:17:23.5] <b>B: SG</b> Also was wir jetzt die letzten Jahre auch für Software-Einführungsprojekte, und das ist ja am Ende ein Software-Einführungsprojekt, oder solche Themen gesehen haben, ist einmal die Unterscheidung zwischen klassischem Projektmanagement und agilem Projektmanagement. Was wir jetzt auch häufig schon sehen bei den Unternehmen, dass die auch grundsätzlich ihre IT- Changeprojekte nicht mehr wasserfallartig orchestrieren. Das heißt ( <b>unverständlich</b> ). Es gibt klarere Roadmap, es gibt ein Fachkonzept, dann gibt es ein EV-Konzept und dann wird das umgesetzt, sondern dass da auch agile Methoden, wie Scrum oder wie Spotify oder wie das SAFe Framework angewendet werden. Und das ist das Erste, das aus meiner Hypothese auch besser geeignet ist für diese Plattform, auch wenn manche Ziele relativ eindeutig sind und wir nicht wie bei einer normalen App sind, Anforderungen sind unklar. Aber das ist etwas, was auf alle Fälle, was ich machen würde, von der Projektorganisation ist, glaube ich, dass auch, was wir sehen, was ein Erfolgsfaktor ist. Ich würde auch nicht mehr von einem Projektleiter sprechen, sondern von einem Product Owner für diese Plattform und das ist im Idealfall der, der sie auch später dann in der Linie, wenn es eingeführt ist, übernimmt. Klar ist es immer wichtig, darüber eine Art Stakeholder-Steuerungskreis zu haben, Sponsoren, die auch Entscheidungen treffen können, das heißt, das sind dann entweder Bereichsleiter, Vorstände oder wie auch immer das in dem Unternehmen aufgesetzt ist, um auch Entscheidungen herbeiführen zu können. Und vom grundsätzlichen Team, die dann in dem Scrum Team sind, zur Einführung, ist immer so ein bisschen die Frage, wie man aufgestellt ist. Was meine ich damit? Es macht vielleicht auch manchmal keinen Sinn, einen Fachbereichsmitarbeiter für eine Einführung komplett aus seiner Umgebung herauszuholen, weil er dann den Bezug dazu vielleicht verliert, andererseits sagt man, wenn man agil unterwegs ist: Mitarbeiter sollen voll committed sein, das heißt, die sollen eigentlich zu 100 % in einem Scrum-Team arbeiten. Das heißt, ich würde das immer so schneiden,

dass es ein Scrum-Team gibt, was sich grundsätzlich um die IT kümmert, um die Governanceeinführung, um die technische Umsetzung, ich hätte dann aber hintendran ein Team, das quasi auch einen ersten MVP, einen ersten Prototyp für den Fachbereich mit dem Fachbereich entwickelt. Und das ist ein wichtiger Punkt und von der Organisation her definitiv auch an ein Backlog zu denken und auch das wirklich auch klar zu definieren: Wann ist Definition of Done? Bis wann soll was erledigt sein? Und auch wirklich dort das mit aufzunehmen. Das wären so wichtige Eckpfeiler vom Projektmanagement. Also wirklich nicht klassisch, sondern agil, aber so agil adaptiert, wie es auch für Unternehmung oder den Konzern passt, weil, Scrum gibt Leitplanken, zum Beispiel, aber es ist ja nicht so, dass man es unbedingt so machen muss, sondern es gibt dann auch bestimmte Kulturen, Werte oder Organisationsformen, die das einfach in einer großen Unternehmung vielleicht nicht anders zulassen.

- 32 [0:20:31.9] **I: RS** Meine Frage wäre jetzt nur, wenn Sie quasi ein Scrum-Kernteam haben und aber trotzdem Mitarbeiter in einzelnen Abteilungen, die quasi dort trotzdem ihre Fachexpertise mit rausziehen, kann es ja zu dem Problem kommen, dass manche Mitarbeiter sich eher zu ihrem Linienvorgesetzten zugehörig fühlen als zu dem eigentlichen Projekt. Sehen Sie die Gefahr auch oder ist das eher etwas, das händelbar ist?
- 33 [0:21:05.0] **B: SG** Ich glaube, das ist immer händelbar, weil es auch aus Erfahrung, es ist ja in anderen Projekten auch nicht anders gewesen. Ob man es jetzt agil oder was weiß ich, sondern es war schon immer so, dass es ein Projekt gibt und ich glaube, was dort wichtig ist, Zugehörigkeit ist immer die Frage: Fühle ich mich dem Projekt zugehörig oder meiner Heimat, wo ich aufgehängt bin? Und ich glaube, man kann zwei Identitäten haben und sollte es auch, weil, das Projekt ist ja irgendwann ausgelaufen und irgendwann fertig, da ist die Plattform zumindest eingeführt, da ist ja meine Identität in gewisser Weise weg. Deshalb, glaube ich, sehe ich da keine Konflikte und wenn ein Mitarbeiter dort Konflikte hat zwischen Projekt und Linie, weil jetzt vielleicht 50/50, 60/40 aufgeteilt ist, ist das natürlich auch ein klarer Punkt, den man an eine Projektleitung oder an einen Product Owner bzw. an einen Teamleiter adressieren muss, dass es dort Konflikte gibt und diese auflösen können muss. Aber von der Identität her, es sei denn das ist ein Mitarbeiter, der komplett immer nur Projekte arbeitet, also wie so eine Art inhouse Consulting, der wird sich da schon wohlfühlen. Ich würde einen Mitarbeiter nicht bis ins letzte rausholen, gerade dann, wenn es um Low Code/ No Code- Anwendungen geht und vielleicht er dann natürlich die frohe Botschaft in den Fachbereich zurücktragen kann, wie toll das alles ist.
- 34 [0:22:30.8] **I: RS** Das ist immer das Beste, wenn so was passiert. Sie haben ja jetzt schon gesagt, dass Sie das agile Vorgehensmodell klar favorisieren. Würden Sie das bei der Einführung als reinrassiges agiles Vorgehen nehmen oder würden Sie trotzdem noch sagen, ein paar Sachen aus dem traditionellen Weg, aus dem Wasserfall, zum Beispiel, Phasen, können schon von Vorteil sein? Oder sagen Sie: "Nö, agil passt." Und warum Sie das direkt bevorzugen würden, würde mich auch noch interessieren.
- 35 [0:23:03.7] **B: SG** Also ich glaube, das ist, zu mindestens bei den Mandanten, bei denen ich jetzt unterwegs bin, ist es immer ein bisschen Mischform. Weil, was meine ich damit, wenn wir jetzt bei Banken sind, gibt es zum Beispiel eine ING diba, die hat sich als erste komplett agilisiert. Das sehen wir auch, aber andere haben das noch nicht gemacht. ING ist auch in der Presse, da kann man auch alles drüber erzählen und das ist auch alles fein. Andere haben das noch nicht gemacht und ich glaube, es sind sowieso schon immer Mischformen, die aktiv werden. Also rein agil macht dann Sinn, auch wenn ich wirklich das Ziel noch nicht konkret vor Augen habe und sage, ich brauche eine App oder eine Anwendung für Problem XY und dann immer wieder auch häppchenweise an den Kunden, also wie ein Start-up, an den Kunden eine App oder eine Anwendung hinausgeht. So, aber wenn wir jetzt von einem reinen Einführungsprojekt sprechen, von einer Software, dann ist das Ziel ja relativ definiert. Ich



kann mir ja überlegen, welche Features und Module ich als Erstes löse und deshalb wäre für mich das dann schon agil, aber es gibt halt Elemente darin, die einfach auch in gewisser Weise wahrscheinlich nicht agil sind. Das fängt bei irgendwelchen Statusberichten an. Das fängt damit an, wer hat jetzt Einfluss auf so ein Projekt? Ja und wie werden auch Entscheidungen getroffen letztlich? Also ich glaube, daher ist es wahrscheinlich schon eine Mischform, aber ich würde die agilen Elemente nehmen, weil das einfach auch dazu führt, dass ich auch peu à peu einzelne Features und Module einführen kann und nicht alles als Big Bang machen muss, das heißt, der Mitarbeiter, der Endnutzer, kann es schneller nutzen.

- 36 [0:24:37.7] **I: RS** Nur für mich noch mal zum Verständnis. Was würden Sie, wenn Sie sagen, es ist eine Mischform meistens, aber ganz klarer Schwerpunkt liegt auf agil, aber was würden Sie sagen, abgesehen von Dokumentationsfragen, würden Sie sagen, es gibt noch andere Sachen aus dem traditionellen Projektmanagement, die da häufig mit vorkommen? Gibt es da Sachen, die sich wiederholen, die wiederholend vorkommen?
- 37 [0:25:06.6] **B: SG** Was man auch noch sieht. Also man muss ja jetzt überlegen, ob man einen Projektstatusbericht macht, wenn man zum Beispiel in Tools wie Jira ein Backlog pflegt für so ein Einführungsprojekt, dann steckt ja dort die Wahrheit drin und ich kann mir dort anzeigen lassen, an welchen Punkten welche Sachen im Backlog sind, welche Sachen im aktuellen Sprintbacklog sind und krieg ja auch dann über gewisse Kennzahlen, die so ein Scrum-Team hat, wie viele Feature Points, habe ich jetzt zum Beispiel im letzten Sprint geschafft und so weiter, kriege ich ja Input darüber, wie es läuft. Aber was man bei großen Unternehmen mit viel Geld sehen, das ist noch klassisch verankert, da will irgendjemand, der dann alle Projekte zusammen bündelt, eine Ampel sehen. Da ist natürlich die Frage, also Statusampeln, ist das jetzt agil oder nicht? Da kann man immer drüber streiten, aber da werden andere Kennziffern gemacht. Da wird auch erwartet, dass ein Projektfortschritt gemessen wird, in 85 % ist fertig und solche Sachen. Also das sieht man schon noch, dass dann halt einfach alte Strukturen dazu führen, dass man sich alter Elemente, gerade von Statusberichten, und so was bedient.
- 38 [0:26:18.5] **I: RS** Sie werden merken, ich komme jetzt zu der gleichen Frage wie vorher schon. Sehen Sie Hürden, wenn Sie bei diesem hybriden Schwerpunkt agilen Vorgehen unterwegs sind? Sehen Sie da Hürden, auf die Unternehmen stoßen könnten? Also sehen Sie da Hürden?
- 39 [0:26:36.2] **B: SG** Na gut, das kommt darauf an. Also wenn eine Unternehmung schon mehrere agile Projekte gefahren hat und weiß, wie es geht, was meine ich damit, die Mitarbeiter haben schon Erfahrung damit, dann sehe ich da keine Hürden. Aber wenn das sozusagen als Vehikel benutzt werden würde, um eines der ersten agilen Projekte zu nutzen, dann ist das schon kompliziert, weil, es kennt noch keiner so richtig die Rollen. Scrum Master, was macht der eigentlich den ganzen Tag? Für was ist der eigentlich zuständig? Ach so, der soll so Daily moderieren, aber da brauch ich doch keinen für. Also Daily, wieso soll ich jeden Morgen sagen, welche Sachen ich mache und wieso soll ich mich jetzt committen, was ich in zwei- oder vierwöchigen Sprints erledige. Und das sind, glaube ich, Punkte, das ist natürlich eine Hürde, die dann definitiv mit reinspielt, wie weit ist dieses Mindset dafür und was natürlich auch eine Hürde sein kann, ist immer bisschen der Punkt, dass Agilität so ein bisschen manchmal als Flexibilität wahrgenommen wird. Das heißt, dass vielleicht gewisse Stakeholder oder Personen auch das Backlog stark beeinflussen möchten und auch vielleicht die Priorisierung des Backlogs dabei. Was natürlich nicht immer so an der Stelle vorgesehen ist, wenn man es einmal abgestimmt hat. Und das sind so ein bisschen die Hürden, die man schon so sieht und die früher in einem klassischen Prinzip, da weiß halt jeder, was er zu tun hat, was er sein lassen darf und muss oder was er machen darf.
- 40 [0:28:08.9] **I: RS** Sie haben ja jetzt schon einige agile Methodiken genannt. Können Sie mal konkret sagen, welche Methodiken Sie da ganz klar bevorzugen bei so einem

Einführungsprojekt? Also Scrum habe ich schon rausgehört, ist wahrscheinlich bei Ihnen ein favorisiertes Modell. Gibt es noch weitere, also Kanban etc.?

- 41 [0:28:28.6] **B: SG** Kanbanboards nutzen an der Stelle ist natürlich auch irgendwie immer. Ich habe die Erfahrung, zumindest dort, wo ich unterwegs bin, dass Scrum irgendwie so eine gängige Praxis ist. Es ist so ein bisschen auch in Mode gekommen und auch, von dem Framework her, relativ einfach. Das ist so, ich nenne es mal so, dass es so klein ist, also klein ist relativ, aber es ist kein Riesen- Einführungsprojekt, wo ich mehrere hundert Mann habe, dann würde ich zu so einem skalierten Framework tendieren, wie SAFe, oder man könnte auch, wenn es mehrere Teams sind, an ein Scrum of Scrum denken oder ein Nexus. Die Erfahrung zeigt aber, zumindestens dort, wo wir unterwegs sind, dass viele Scrum als a und o nutzen oder in abgewandelter Form stattfindet. Das ist so ein bisschen Mode.
- 42 [0:29:23.9] **I: RS** Sie haben ja quasi auch schon angesprochen, dass es zu Problemen kommen kann, wenn viele Änderungswünsche an das Projekt gestellt werden. Auch aus dem Projekt können ja Änderungswünsche kommen. Wie würden Sie mit diesen Änderungswünschen prinzipiell umgehen, wenn jetzt von außen einmal an das Projekt Änderungswünsche gestellt, aber auch, wenn von innerhalb des Projektes selbst ein Änderungswunsch kommt, weil entdeckt wurde, dass irgendetwas nicht so funktioniert, wie man sich das erhofft hat?
- 43 [0:29:52.7] **B: SG** Also erstens: in Backlog aufnehmen. Das ist, glaube ich, der erste Punkt und dann ist natürlich ein zweiter Punkt. Ich habe mein Backlog ja vorab priorisiert bzw. priorisiere das immer wieder innerhalb eines Sprints und dann ist natürlich der Punkt, ist es etwas, was eine hohe Priorisierung hat, weil es ein wichtiges, ein kritisches Item ist, weil es vielleicht auch für den Endkunden, für den Nutzer, wichtig ist? Und das ist natürlich ein Punkt, immer wieder zu priorisieren, die Punkte aufzuwerten und das auch dem Scrum-Team dann dort auch zu überlassen, was in das Backlog kommt und dort mit den Stakeholdern auszutauschen. Was natürlich nicht passieren soll, ist, dass jemand einfach reingrätscht und sagt, das ist jetzt mein Killerfeature und den ganzen Plan durcheinander bringt. Das ist eigentlich nicht vorgesehen und wer ist da in der Rolle? Der Scrum Master ist da definitiv in der Rolle, die Methodiken einzuhalten und dafür zu sorgen, dass das Sprintteam auch adäquat arbeiten kann. So, das ist ein wichtiger Punkt und dort das dann auch zu leben. Wir wissen alle, dass irgendwie Theorien gut funktionieren, wenn jetzt so eine, ich bleibe mal beim Datenschutz, wenn eine EU DSGVO erst nicht aktiv ist und während so einem Einführungsprojekt reinfliegt, weil es plötzlich wichtig ist, dann hat man natürlich ein Risiko, dass man etwas nicht rechtlich Konformes einführen würde und das ist natürlich eine andere Priorisierung vielleicht, als wenn irgendwie ein Feature, ob eine Funktionalität, wie kann ich jetzt mobil arbeiten oder nicht mobil arbeiten. Kann ich Externe hierzu einladen oder nicht, das ist was ganz anderes, das hat eine ganz andere Kritikalität.
- 44 [0:31:40.7] **I: RS** Sie haben ja jetzt schon über Scrum Master, Product Owner, gesprochen, wenn Sie an wichtige Rollen des Projektes denken, sind das die klassischen Rollen, die man aus dem Scrum, zum Beispiel, ganz häufig hört oder sagen Sie, es gibt auch Rollen, die Sie als weniger wichtig dabei empfinden und andere, die müssen drin sein, ohne scheitert das Projekt?
- 45 [0:32:07.9] **B: SG** Also was ganz wichtig ist. Ich weiß nicht, ob es Rollen oder als Position noch im Unternehmen ist. Das sind ja IT-Einführungsprojekte, das heißt, es ist einerseits wichtig, so eine Compliance Abteilung für das Thema Datenschutz mit einzubeziehen, die Informationssicherheit, was auch zulässig ist und, nicht zu guter Letzt, das sollte man in Konzernen nie vergessen, das ist der Betriebsrat oder die Personalversammlung. Das heißt, all das beeinflusst ja Mitarbeiter und da hat der Betriebsrat auch ein Mitspracherecht und auch solche Tools müssen in so eine Konzern- oder einer Betriebsvereinbarung aufgenommen werden. Das heißt, diese Personen oder Personengruppen mit zu involvieren von Tag eins, ist unheimlich wichtig, weil, am Ende können das die Show Stopper sein, dass ein Tool nicht

- ausgerollt wird. Habe ich auch schon erlebt. Es war bei einer MS Teams- Einführung bei einem Kunden hat sich der Rollout zwei Wochen verzögert, weil es keine Freigabe des Betriebsrats gab. Und das ist, glaube ich, etwas, was außerhalb dessen ist, was man vielleicht im Framework explizit sieht. Es ist vielleicht ein Stakeholder, aber ein Stakeholder, der halt nicht als Stakeholder bezeichnet werden sollte, sondern eine sehr wichtige Rolle einnimmt.
- 46 [0:33:22.6] **I: RS** Das wäre quasi auch meine nächste Frage gewesen, welche wichtigen Stakeholder Sie im Projekt sehen. Bloß aber noch mal zur Klarstellung, aber innerhalb des Projektes sehen Sie die ganz klassischen Scrumrollen wie Scrum Master, Product Owner, das Development-Team, einen Sponsor, das sind so die Rollen, die Sie sehen?
- 47 [0:33:41.8] **B: SG** Genau.
- 48 [0:33:43.7] **I: RS** Abgesehen von der Compliance Abteilung und dem Betriebsrat, sehen Sie noch andere wichtige Stakeholder im Unternehmen, die auf jeden Fall in diesem Projekt, zumindest im Informationsfluss, eingebunden sein sollten?
- 49 [0:33:57.2] **B: SG** Also grundsätzlich ist es immer die Frage, wie man es dann aufsetzt und wer in so einem Einführungsteam ist. Am Ende geht es ja darum, dass die Mitarbeiter, je nachdem fünftausend, zehntausend, vielleicht auch nur fünfhundert Mitarbeiter, die das Tool nutzen können, das heißt, es ist ja auch in gewisser Weise ein Stakeholderkreis und den muss ich ja auch bespielen oder die auf die Reise mitnehmen. Was dann passiert, wenn ich die nicht mit auf die Reise nehme, dann sage ich an einem Tag, ich schalte jetzt den Schalter um und das Tool, es gibt die No Code/ Low Code-Plattform, aber keiner davon weiß oder was man damit machen kann und ich finde einfach, das ist ja schon ein Veränderungsprozess, ein Change, das heißt, es sind auch die Fachbereiche bzw. auch sämtliche Einheiten des Unternehmens in gewisser Form mitzunehmen. Das kann über Leuchttürme da rein passieren, aber auch über grundsätzliche Kommunikationsmaßnahmen vorab und das ist, glaube ich, ein Stakeholder, den man häufig unterschätzt, weil, am Ende müssen die Mitarbeiter es ja auch nutzen.
- 50 [0:34:57.2] **I: RS** Da stimme ich Ihnen total zu. Wenn wir gerade dabei sind. Wir sind jetzt bei einem mittleren bis wirklich großen Konzern. Dann führen die ja nicht auf einen Schlag irgendwas ein, sondern das wird ja meistens in einer Pilotabteilung, in einer Prototypenabteilung, eingeführt. Wie würden Sie die Skalierung in den ganzen Konzern oder in das größere Unternehmen gestalten, wenn das jetzt erfolgreich in dieser einen Abteilung eingeführt worden ist? Wie würden Sie dann die Skalierung hoch in den Konzern machen?
- 51 [0:35:32.0] **B: SG** In jeder Abteilung gibt es, manche nennen es Champions, manche nennen es Leuchttürme, ein oder zwei oder auch drei, je nachdem wie groß die Abteilung oder der Bereich ist, Leuchttürme identifizieren, die auch die frohe Botschaft vorantragen und auch wissen, was diese Tools können und dort sozusagen Botschafter sind, das ist glaube ich eins. Das andere ist natürlich, ganz klar, Intranetseiten zu haben und darüber zu kommunizieren. Also Intranet ist jetzt vielleicht das eine in der Unternehmenskommunikation. Es gibt ja auch viele Unternehmen, die dann auch wie eine Art eigene interne App haben für die Unternehmenskommunikation, das da zu verkünden. Aber was wir auch festgestellt haben, was sehr wichtig für eine Einführung ist, das Stichwort, was man unter Communities festmacht. Das heißt, zum Beispiel, eine Kollaborationsplattform wie Slack oder Teams oder so zu nutzen und dort eine Community für diese No Code/ Low Code-Plattform zu bilden. Das heißt, dort werden einerseits Informationen bereitgestellt, was ich schon kann, vielleicht auch erste Trainings angeboten. Andererseits schafft man es darüber auch dann, wenn Kolleginnen und Kollegen Herausforderungen haben in der Nutzung: "Ich habe mal eine Frage?" Dann sind dort ein paar hundert, die man einfach mit einem Schlag fragen kann und die darauf antworten können. Ganz einfach, wie in einer WhatsApp-Gruppe/ -Chat am Ende, und was man natürlich auch machen kann, wenn da sich ein paar aktiv in der Community anmelden. Also es

sollten nicht alle sein, aber man sollte sich dafür aktiv anmelden. Man schafft es natürlich dort dann auch vielleicht, die nächste Testgruppe für einzelne Features zu finden und denen vorab schon Sachen als nächste Pilotgruppe bereitzustellen. Also, wenn ich vielleicht eine Bot-Funktion noch nicht aktiviert habe, eine AI-Funktion noch nicht, die das dann mitgehen zu lassen. Das ist ein wichtiger Punkt, der auch sehr gut funktioniert und der, aus eigener Erfahrung ich schon festgestellt habe, dass das ziemlich cool ist.

- 52 [0:37:16.5] **I: RS** Das spielt ja auch in das Thema Befähigung der Mitarbeiter rein. Wie gesagt, die kommen nicht durch die Tür, Sie haben das ja schon angeschnitten, und wissen sofort, wie das funktioniert. Die Community und Schulungen sind so das einzige, wo Sie sagen, da kann man bei der Befähigung der Mitarbeiter ansetzen. Oder sagen Sie, da gibt es noch mehr?
- 53 [0:37:37.0] **B: SG** Gut, man kann ja überlegen, also manche dieser Hersteller bieten ja auch so Zertifizierungsplattformen an. Das ist ja auch noch mal so ein ganz nettes goody, dass man als Mitarbeiter quasi so eine Zertifizierung, also wie beim Scrum Master, gibt es eine Zertifizierung, aber für diese Plattformen auch. Das ist noch ein guter Punkt. Und sonst, wie gesagt, Communitys, ganz klassische Trainings und was man sich auch anschauen sollte, das klingt jetzt wieder aber YouTube- Videos, also es gibt ein paar Koryphäen für einzelne Plattformen und wenn die erzählen, was möglich ist, dann muss ich persönlich sagen, da bin ich mit einem 5-Minuten-Video schneller, als mich einmal selbst durchzuprobieren und das auch natürlich zu kommunizieren und sagen, das sind so die best of links.
- 54 [0:38:22.0] **I: RS** Mich würde jetzt aber auch noch interessieren: am Ende von einem Projekt kommt irgendwann der Accounter und sagt: "Na, ihr habt das jetzt eingeführt. Woran messt ihr denn bitte jetzt für mich in Kennzahlen, dass das Projekt erfolgreich war?" Oder wie würden Sie die Qualität des Projektes messen oder sicherstellen?
- 55 [0:38:52.4] **B: SG** Aus Anbietersicht also für eine No Code/ Low Cod-Plattform oder aus Unternehmenssicht, das so ein Programm einführt?
- 56 [0:38:58.0] **I: RS** Aus der Unternehmenssicht.
- 57 [0:38:59.1] **B: SG** Wann ist ein Projekt denn erfolgreich? Das ist ja eigentlich im Sinne so: in Time, in Budget, in Scope, hat man ja immer so schön gesagt, im klassischen. Zu Projektabschluss, kann man sagen, hat man all das erreicht, was man in seinem Backlog drin hatte, zu einem gewissen definierten Zeitpunkt, dann ist das abgeschlossen worden und dann bin ich auch, das muss ganz klar sagen, in dem Kostenpart drin. Das ist aber eher das Projekt, aber wenn ich jetzt messen möchte, war ich erfolgreich, diese No Code/ Low Code-Plattform im Unternehmen bereitzustellen. Also wird sie danach genutzt, dann habe ich natürlich unterschiedliche Messkriterien, also ich kann mir anschauen, wie viele Apps und Anwendungen werden angefangen, wie viele werden entwickelt und wie viele werden Live geschaltet? Wie ist grundsätzlich die Nutzungsrate und was ich natürlich auch machen kann, was ich mir definitiv mal anschauen würde, man kann das jetzt fassen unter Mitarbeiterbefragung, aber wie zufrieden sind eigentlich die Mitarbeiter mit diesem Tool, mit dieser Nutzung? Was würden Sie sich noch wünschen? Das macht man entweder unter strukturierten Interviews, in Umfragen, in, man kennt es so bisschen vom Baumarkt, wo man dann draufhaut und sagt: "Wie zufrieden waren Sie heute mit ihrem Einkauf?", das kann man auch machen. Es ist halt einfach wirklich dann daran, am Puls der Zeit zu fühlen, wie zufrieden sind die Mitarbeiter damit und daran würde ich auch einen Erfolg definieren.
- 58 [0:40:21.1] **I: RS** Dann, wir ziemlich schnell gewesen. Wir sind jetzt schon fast am Ende von unserem Interview. Mich würde jetzt noch interessieren, haben Sie noch Anmerkungen zum Interview? Sachen, die ich nicht gefragt habe, wo Sie aber sagen, das ist mir besonders

wichtig oder best practices, die Sie noch gerne teilen möchten, die jetzt auch nicht vorgekommen sind.

- 59 [0:40:42.2] **B: SG** Also ich glaube, was man einfach sich immer überlegen muss, was sind so Vor- und Nachteile von so einer Plattform. Es gibt ja verschiedene Anbieter und die haben verschiedene, ich habe wenig über Anbieter selber gesprochen, da sieht man natürlich auch Präferenzen bei Unternehmen. Der eine sagt, ich finde jetzt Microsoft ganz toll, weil, das kennt ja jeder, weil sich das einbettet in PowerPoint, in Excel. Kommt dann aber der nächste um die Ecke und sagt: "Nee, ich habe es ein bisschen komplizierter. Ich möchte gerne Pega nutzen." Oder "Nee, ich mag dann doch noch mal Appian, weil das auch mit komplexeren Workflows möglich ist und ich eigentlich ganz andere Sachen damit mache." Also von daher, ich glaube ein wichtiger Punkt ist einfach, wie mache ich so eine Anbieterauswahl und welche Tools, Prozesse, kann ich mir als Unternehmen vorstellen, zukünftig in so einer No Code/ Low Code-Plattform abzubilden. Das ist, glaube ich, noch mal ein wichtiger Punkt, den ich an der Stelle auch noch mal mitgeben würde, weil, die sind schon unterschiedlich.
- 60 [0:41:44.7] **I: RS** Dann bedanke ich mich ganz sehr und ich würde jetzt die Aufnahmen beenden.
- 61 [0:41:57.9] **Ende**

## Einwilligungserklärung Interview

Ich erkläre hiermit mein Einverständnis zur Nutzung der personenbezogenen Daten, die im Rahmen des folgenden Gesprächs erhoben wurden:

Datum des Interviews	13.07.2022
Namen der zu interviewenden Person	Simone Göttl
Name der interviewenden Person	Rocco Schenk
Kurzbeschreibung des Forschungsprojekts	Einführung von No Code/ Low Code aus Sicht des Projektmanagements
Durchführende Hochschule	Hochschule Mittweida – University of Applied Science

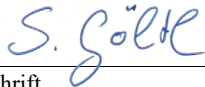
Die Daten werden im Rahmen eines mündlichen Gesprächs erhoben, das mit einem Aufnahmegerät und Videoaufnahmegerät aufgezeichnet wurde. Zum Zwecke der Datenanalyse werden die mündlich erhobenen Daten verschriftlicht (Transkription), wobei die Daten nicht anonymisiert werden.

Vorhandene Kontaktdaten, der interviewten Person, werden aus Dokumentationsgründen in einem separaten Schriftstück lediglich den Gutachter\*innen der wissenschaftlichen Ausarbeitung zur Verfügung gestellt. Nach dem Abschluss des Projekts werden diese Daten gelöscht.

Der Speicherung der personenbezogenen Daten zu Dokumentationszwecken kann durch die interviewte Person jederzeit widersprochen werden. Die Teilnahme an dem Gespräch erfolgt freiwillig. Das Gespräch kann zu jedem Zeitpunkt abgebrochen werden. Das Einverständnis zur Aufzeichnung und Weiterverwendung der Daten kann jederzeit widerrufen werden.

Simone Göttl

Vorname und Name in Druckbuchstaben



Unterschrift

Frankfurt, 13.07.2022

Datum, Ort

## Anlagen 6, Interview EB05

RS: Rocco Schenk

CR: Christian Rohrbeck

1	[0:00:00.0] <b>Start ( CR - Christian Rohrbeck/ RS - Rocco Schenk)</b>
2	[0:00:07.5] <b>I: RS</b> Ich begrüße dich ganz herzlich zum Interview. Vielen Dank, dass du dir die Zeit nimmst. Ich würde dich zuerst fragen, ob du damit einverstanden bist, dass ich dieses Interview aufnehme?
3	[0:00:18.3] <b>B: CR</b> Ja, ich bin damit einverstanden Rocco.
4	[0:00:21.6] <b>I: RS</b> Was mit deinen Daten passiert, nur als Info für dich. Ich zeichne das auf, zu Transkriptionszwecken. Die Audio- und Videodateien lösche ich dann wieder. Das Transkript kommt in den Anhang meiner Masterarbeit. Du wirst nicht anonymisiert innerhalb der Masterarbeit, da deine Expertenmeinung ja herausgestellt wird und man dich als Experten erkennen muss. Dann würde ich quasi direkt damit einsteigen wollen. Kannst du dich mal bitte kurze vorstellen, wie dein Projektmanagement- Kontext ist und auch dein Kontext zu No Code/ Low Code-Plattformen?
5	[0:00:56.1] <b>B: CR</b> Ja, sehr gerne. Vielleicht ganz kurz auch zu meinem Hintergrund. Ich habe Wirtschaftsinformatik studiert, bin Mitte der 90er-Jahre damit fertig geworden. Damals war das, ich wollte schon sagen, Master, nein, es ist ein ganz normales Diplom FH. Ich habe das an einer Hochschule gemacht, für Technik und Wirtschaft in Dresden. Ich fand damals das Studium, vielleicht ganz kurz, sehr Technik-lastig. Der BWL-Anteil war sehr gering und auch der Anteil von dem, was mein eigentliches to-do heutzutage ausmacht, war sehr gering. Es geht nämlich hier, das ist mein aktuelles Arbeitsfeld als IT-Projektmanager, in dem ich tatsächlich seit 20 Jahren aktiv bin, merke ich, es geht hauptsächlich um die Kommunikation und das Vermitteln von Wissen, oder das Bereitstellen von Wissen für die richtigen Personen, in der richtigen Menge. Und das ganze noch mit einem entspannten Hintergrund. Sodass man, ich sage mal, von der emotionalen Seite, da sich gut abgeholt fühlt, von beiden Seiten, zwischen denen man steht. So, das würde ich jetzt einfach mal als das Wichtigste meiner Arbeit begreifen. Was auf keinen Fall da zu vernachlässigen ist, ist natürlich, dass man die Sprache der Personen spricht. Also sprich, man muss auch in der Lage sein, die technischen Hintergründe zu verstehen und da auch in der Lage sein, Entscheidungen zu treffen. Das heißt, dieser Technikanteil, der kommt mir jetzt hier dann zugute, den ich im Studium so bemängelt habe. Ich muss dazu gestehen, dass ich mit Low Code/ No Code-Plattformen selber einmal in Berührung gekommen bin, da hatte ich eine Software entwickelt, die in der Lage war, Druckdaten zu layouten und in ein Formular reinzubringen. Und das war halt ein Drag-and-drop-System, wo man bestimmte Felder in bestimmte Positionen zog. Nur von meinem Verständnis her, das ist das, was ich als Low Code- Plattform interpretiere. Man kann im Prinzip mit optischen Mitteln bestimmte Algorithmen abbilden, ohne jetzt hier wirklich, ich sage jetzt mal, die Befehlszeilen, die Codewörter und sonstiges zu kennen. Ich bin also in der Lage, das abzubilden und als No Code-Plattform, da fehlt mir jetzt das wissenschaftliche Verständnis für, wie das jetzt dort definiert ist. Da müsste ich jetzt einmal kurz nachschauen, was es unterscheidet oder vielleicht

kannst du mir auch ganz kurz den wissenschaftlichen Anteil geben, dann kann ich auf jeden Fall vielleicht was dazu sagen. In meiner Vorstellung ist es dann halt, keine Ahnung, ich kann es nicht sagen, vielleicht kannst du mich da kurz aufklären.

6

[0:04:08.8] **I: RS** Also generell deine Beschreibung von No Code ist quasi das, was du schon gesagt hast. Womit du auch Berührungspunkte hattest. Die wissenschaftliche Beschreibung ist: Du ziehst einfach über grafische Bausteine, ohne Kenntnisse von Programmierung, quasi ein Programm zusammen. Und der Unterschied zu Low Code ist einfach nur der kleine Aufsatz, dass du ein paar einzelne Zeilen kennen musst, ähnlich wie Excel, ganz, ganz grob. Das ist die wissenschaftliche Beschreibung. Ich muss jetzt nachfragen, du hast jetzt auch die Expertise, dass du jetzt zur Einführung von Low Code etwas sagen kannst, zu solchen IT-Projekten?

7

[0:04:49.3] **B: CR** Gut, dass du mich das noch mal fragst. Ich habe es, wie gesagt, in meiner damaligen Tätigkeit als Entwickler selber einmal gemacht komplett. Habe also noch in Erinnerung, gutes Gefühl, wie das gewesen ist und denke, dass ich als Basis schon in Kombination mit meinem aktuellen Tätigkeitsfeld, dann die Expertise dafür habe.

8

[0:05:13.2] **I: RS** Dann wäre meine nächste Frage: Hast du den Begriff Citizen Development schon einmal gehört?

9

[0:05:21.1] **B: CR** Auch hier ist es so. Ich kenne diesen Begriff aus der Tätigkeit, die ich im Rahmen des PMI habe, aber ich weiß, dass da was ganz anderes hinter steckt, als man das jetzt auf den ersten Schlag vermuten würde.

10

[0:05:42.7] **I: RS** Dann sag einfach, wie du es kennst. Also erst mal zur Klarstellung. PMI, kannst du erst mal noch kurz sagen, was das ist?

11

[0:05:51.3] **B: CR** Ja sicher. PMI ist das Project Management Institute, das ist, ich sage mal, für uns Projektmanager gibt dieses Institut eines der großen Zertifikate aus, was sehr akzeptiert ist. Auch mittlerweile in Europa. Das kommt ursprünglich aus Amerika und der Name verrät es schon. Die sind in den USA ansässig und haben einen großen Beitrag zur Standardisierung des Projektmanagements geleistet, aber unabhängig davon gibt es natürlich auch in Deutschland und in den USA da entsprechende Pendant, sage ich mal. Ich habe mich für PMI damals entschieden, weil ich tatsächlich eine Zeit lang in Kanada war und da war das schon sehr populär, aber lang bevor das hier in Europa und Deutschland so breitflächig eingeführt wurde.

12

[0:06:59.3] **I: RS** Dann noch mal auf die Frage. Du hast eine Vorstellung von Citizen Development. Kannst du kurz sagen, wie die ist?

13

[0:07:08.4] **B: CR** Ja, also meine Vorstellung geht dahin, dass, vielleicht nur ganz kurz zu den Organisationsformen, in denen ich mich bewege. Ich bin meistens in größeren Unternehmen unterwegs. Wir haben hier in der Regel eine Matrixorganisation. Das heißt, die Unternehmen selber sind in Abteilungen organisiert und haben da Abteilungsaufgaben und sobald wir ein Projekt machen, ist es so, dass wir im Prinzip da einen Durchstich machen, durch die Abteilungen. Bestimmte Kollegen da mit Aufgaben betrauen und dadurch auch sehen, welche Abteilungen wie miteinander arbeiten. Und meine Vorstellung von dem Citizen Development ist diejenige, dass man einem Kollegen, der in einer Fachabteilung ist und an einem speziellen



Problem arbeitet, eine solche Software an die Hand gibt und dass dieser sich, ohne jetzt Fachkenntnisse von Programmierung zu haben, in der Lage sieht, einfache Algorithmen aufzusetzen, die dann Funktionen ausführen innerhalb dieses speziellen Geschäftszweiges, in dem die Kollegen da dann unterwegs sind.

14

[0:08:18.9] **I: RS** Mit dem, was wir jetzt über No Code/ Low Code und Citizen Development besprochen haben, welche Vorteile siehst denn du dann von solchen Plattformen?

15

[0:08:35.8] **B: CR** Also, ich denke mal, der absolute Vorteil, der Hauptvorteil, ist einfach, dass hier das fachliche Know-how ohne Reibungsverluste in eine IT-Lösung umgesetzt werden kann. Das muss man sagen, ist in unserem Berufsfeld das Hauptproblem, dass wir verschiedene Ansprechpartner haben, die so unterschiedliche Sprachen sprechen, dass da enorm hohe Kommunikationsreibungsverluste da sind, das entfällt. Also wir haben, derjenige der den Prozess verstanden hat, ist auch in der Lage, das abzubilden in ein System. Und natürlich könnte man jetzt auch sagen, die überlasteten IT-Leute werden da entlastet. Aber das sind alles Nebenvorteile, sage ich mal. Hauptsächlich geht es darum, dass keine Reibungsverluste entstehen.

16

[0:09:26.5] **I: RS** Und siehst du da auch Nachteile?

17

[0:09:29.7] **B: CR** Absolut. Ich merke halt immer wieder, wenn ich mit den Kollegen vom Fachbereich unterwegs bin, dass da, ich sage mal, grundsätzliches IT-Verständnis überhaupt nicht vorhanden ist. Weder wird verstanden, warum bestimmte Sachen gewartet werden müssen. Also absolute Basics, mit denen ich halt aufgewachsen bin, fehlen und das führt dann auch dazu, dass eine falsche Erwartungshaltung, zum Beispiel, an solche Software gestellt wird oder im Falle des Misslingens, sage ich mal, nicht die richtigen Schlüsse gezogen werden. Sprich, dass dann die eigentliche Umsetzung das Problem ist. Also die Transformation der fachlichen Information in die IT-Technik. Das hier an genau der Stelle dann Zeitaufwände entstehen, die bei einem ITler nicht entstehen würden. Nur der ITler hat halt das fachliche Know-how nicht. Also es wird verschoben so ein bisschen, die Problematik.

18

[0:10:42.9] **I: RS** Du hast ja gesagt, du hast schon Erfahrung mit der Einführung von einer No Code/ Low Code-Plattform gemacht.

19

[0:10:53.3] **B: CR** Korrekt

20

[0:10:54.9] **I: RS** Was sind so realistische Erwartungen, die man heutzutage wirklich daran stellen könnte an solche No Code/ Low Code-Plattformen, auch aus der Erfahrung, die du damit gemacht hast? Und was sind utopische Gedanken, wo einfach nur der Hype dabei ist?

21

[0:11:12.2] **B: CR** Ja also, realistisch ist auf jeden Fall erstmal, sich vom Scope her, dessen was diese Anwendung kann, ein bisschen, ich sage mal, eine einfache Erwartungshaltung zu haben. Ich glaube nicht, dass solche Plattformen geeignet sind, hochkomplexe Prozesse abzubilden, sondern eher für einfache Hilfs-, na ja, für Hilfsaufgaben ist jetzt wieder untertrieben, für einfachere Aufgaben gedacht ist und das halt, tja, was sie realistisch erfüllen können. Sie können realistisch helfen, so eine Art Halbautomatisierung durchzuführen. Dass man im Prinzip einen Assistenten sich baut, aber trotzdem die ausführende Gewalt, sage ich mal, beim Mitarbeiter verbleibt. Sie sind, was den Hype angeht, ist vielleicht der Hype dahin gehend, dass

man sich vorstellt, dass man halt komplette IT-Prozesse oder IT-Software-Lösungen ablösen kann damit und daran glaube ich persönlich nicht.

22

[0:12:23.9] **I: RS** Ok, du hast ja vorhin schon gesagt, es entlastet auch die IT-Abteilungen. Also es gibt ja ein gewissen Fachkräftemangel, auch in der IT. Würdest du sagen, das kann dem helfen, dass dieser Fachkräftemangel in den Firmen weniger schlimm ist?

23

[0:12:39.2] **B: CR** Also ich hole mal anders aus an der Stelle. Mal angenommen, man hat ein kleines Dorf und eine Durchgangsstraße, dann staut sich da der Verkehr, viele Leute sind verärgert und man beschließt, eine Umgehungsstraße zu bauen. Jetzt hätte man die Erwartung, dass die Umgehungsstraße dann den Verkehr aufnimmt und die Durchgangsstraße leer ist. Aber es führt dazu, dass insgesamt mehr Verkehr unterwegs ist, weil der ja flüssiger läuft jetzt. Was wieder zu Staus und Verstopfungen führt. Ich kann mir vorstellen, dass das dieses grundsätzliche Problem des Fachkräftemangels nicht oder nur teilweise tangiert oder entlastet. Also ich glaube nicht, dass das hier dazu geeignet ist, denn an dem einen oder anderen Punkt wird derjenige, der aus dem Fachbereich heraus eine solche Anwendung bedient, trotz allem auch Rückfragen haben. Es würde also eher dazu führen, dass mehr IT-Lösungen unterwegs sind und die mehr Human Resources an sich brauchen und am Ende keine Entlastung mit sich bringt, was das Thema angeht.

24

[0:14:00.2] **I: RS** Also du würdest eher sagen, dass durch die Einführung, dass die Fachabteilungen selbst für sich aus ihrem Fachwissen heraus, kleine Anwendungen bauen können, dass dadurch ein gesteigertes Anfrageaufkommen für die IT kommt und quasi keine Entlastung, dass halt weniger Anfragen oder große Anfragen an die IT gestellt werden?

25

[0:14:24.5] **B: CR** Ich bin davon überzeugt, dass die ein oder andere Systemanfrage wegfällt, aber ich glaube eher, dass der Support, den die IT zu leisten hat, also dass im Prinzip sich dadurch die Aufwände von Projektdurchführung eher in Richtung Support verschieben, weil diese Kollegen dann gecocht werden müssen in gewisser Weise, um einfach dann auch fertig zu werden. Also ich würde jetzt nicht sagen, dass es von der absoluten Gesamtheit mehr wird, aber es wird auch nicht spürbar zu einer Entlastung beiführen.

26

[0:15:05.2] **I: RS** Ich stelle dir nachher noch mal eine Frage zu dem Thema. Lass uns erst mal weitergehen. Wir sind jetzt an dem Punkt, wo quasi jede Frage, die ich ab jetzt stelle, bezieht sich immer darauf, wenn eine Firma eine No Code/ Low Code-Plattform einführen möchte, wie dort deine Erfahrungen sind und wie dort dein Wissen ist, immer auf dich persönlich bezogen. Ich erwarte dabei nicht, dass du alles weißt, sondern einfach nur deine persönlichen Erfahrungen, aber immer auf dieses Einführungsprojekt einer No Code/ Low Code-Plattform bezogen. Welche Voraussetzungen müssten denn deutsche Unternehmen erfüllen, zum Beispiel, um erfolgreich eine No Code/ Low Code- Plattform zu integrieren, einzuführen in ihr Unternehmen. Dabei meine ich so Sachen wie Mindset, Unternehmensgröße eventuell, muss Kapital dahinterstehen, Manpower, wie muss die Befähigung eventuell der Mitarbeiter sein?

27

[0:16:11.4] **B: CR** Also ich glaube, was ich halt dadurch, dass ich auch freiberuflich unterwegs bin und schon die eine oder andere Firma gesehen habe, ist erst mal aus meiner Sicht die Firmenkultur. Hat man in der Firma eine Willkommenskultur, ist man offen für neue Dinge, das färbt spürbar ab auf die Mitarbeiter, die da vor Ort unterwegs sind. Und mal, gesetzt den Fall, es handelt sich um eine Firma, die wirklich gut mit ihren Mitarbeitern umgeht, sehr offen ist und diese Form des Know-how-Aufbaus auch unterstützt, dann wären erst mal da die wichtigsten Einstiegsgrößen getan. Ich muss auch sagen, in meiner Erfahrung, dass Kollegen ab einem

gewissen Alter, ich möchte jetzt gar nicht so sehr auf einzelne, aber so rein vom Durchschnitt her, kommt man ab einem gewissen Alter so in eine Idee rein, dass man sagt: "Ich hab das schon immer so gemacht" bzw. ich erlebe das sehr oft, dass ich diese Antwort bekomme, dass man das schon immer so gemacht hat und dass dadurch das Verständnis für neue Voraussetzungen fehlt. Also, Alter spielt durchaus hier eine Rolle. Aber wie gesagt, das möchte ich gar nicht zu sehr pauschalisieren. Es gibt natürlich immer Ausnahmen. Ja, es gibt viele Details, die für ein erfolgreiches Projektteam sprechen. Das kann man auch verallgemeinern, egal, um welche Art von Projekt es sich handelt. Persönlich bin ich auch der Meinung, dass, zum Beispiel, eine Geschlechtermischung gut ist. Ich merke das immer wieder, wenn wir irgendwie Frauen im Team haben, dass dann doch noch mal eine andere Stimmung da ist, die auch mehr auch auf dem Fokus ist. Das tut gut. Das halt einfach da der Projektansatz der Richtige ist, sei es jetzt agil oder Waterfall. Je nachdem, was da halt dann sinnvoll ist, wie das Teams drauf ist, eignen sich da verschiedenste Methoden. Was auch gut ist und das merke ich halt in größeren Unternehmen, die eine relativ hierarchische Einteilung haben, wenn die Abteilungsleiter voll und ganz dahinterstehen. Das größte Problem, was ich zum Beispiel im Moment habe, ist einfach das, dass wenn wir in unserer Matrixorganisation, in der ich gerade unterwegs bin, vom Projekt her, ich sage mal, quer durch Abteilungen stechen und uns Mitarbeiter zu 50 % ausleihen für ein Projekt, wird dieser Mitarbeiter im Zweifel immer seinem Linienvorgesetzten den Vorrang geben, das heißt, ich als Projektmanager ziehe immer den Kürzeren. Das heißt, ganz wichtig ist hier, ich nenne es mal, das Top-Management Commitment oder das Management Commitment, dass die Abteilungsleiter ihre Leute dafür auch das machen lassen und nicht für ihre originären Aufgaben abziehen. Verständnis und die richtige Umgebung, das richtige Umfeld, sind entscheidend. Jetzt noch mal kurz zurückgefragt, ich habe mich, glaube ich, ein bisschen verloren. Habe ich deine Frage damit beantwortet oder ist da jetzt noch Punkt offen?

28

[0:19:42.6] **I: RS** Ich würde das jetzt mal unter dem großen Topic Mindset einordnen.

29

[0:19:49.7] **B: CR** Ok

30

[0:19:50.4] **I: RS** Mich würde noch interessieren, ob du andere Größen siehst, wie Unternehmensgröße, Kapital, Manpower oder die, ob die Mitarbeiter eine bestimmte Vorbildung haben müssen, also eine Befähigung bereits mitbringen müssen? Oder ob du sagst: "Das sind jetzt nicht so wichtige Sachen. Mindset ist das wichtigste aus meiner Perspektive."

31

[0:20:01.6] **B: CR** Also, fangen wir vielleicht mal mit dem Kapital an. Also, prinzipiell sollte von der obersten Unternehmensführung, egal welche Größe, da gesagt werden, das ist eine Investition, dafür stellen wir die finanziellen Mittel bereit. Also das ist eine Basisvoraussetzung, damit da einfach kein Druck herrscht, weil, man kann einfach nicht auch entwickeln, wenn man in einem Sparzwang ist. Das ist einfach auch eine Sache, Projekte sind auch so eine Sache, wo Geld verbrannt wird. Das muss klar sein. So, dann Manpower: Ich glaube, es ist auf jeden Fall gut, wenn man jemanden, der einen solches No Code/ Low Code-System kennt, mit zu den Kollegen setzt, die das am Ende realisieren sollen. Das muss gar keine Schulung sein, aber man kann einfach sich in der Abteilung aufhalten, als Ansprechpartner für diejenigen, die es zu tun haben. Und also ich sage mal, was auf jeden Fall, aus meiner Sicht wichtig ist, um es realistisch zum Funktionieren zu bringen. Ein Coach, der die Entwicklung die ersten Wochen und Monate über hin begleitet, bis so viel Know-how aufgebaut ist, dass man das selber machen kann. Und zwar nicht in kurzen knappen Schulungen mit Frontalbeschallung, sondern in einer entspannten Art und Weise nebenher. Der sich da nebenbei mit reinsetzt. Kostet mehr, ist klar, aber an der Stelle absolut wertvoll, weil, mit einer Schulung ist es nicht getan. Und die Größe des Unternehmens spielt aus meiner Sicht gar nicht so sehr eine Rolle. Auch große

Unternehmen sind in der Lage, sich abteilungstechnisch so klein herunterzumachen, dass man nur noch den Abteilungsfokus hat und das drum rum verschwindet. Es kommt rein auf den Willen für die Innovation an.

32

[0:22:10.8] **I: RS** Siehst du bei der Einführung einer No Code/ Low Code-Lösung auch bestimmte Risiken und Probleme, auf die man da ein bisschen ein Auge haben sollte?

33

[0:22:26.3] **B: CR** Absolut. Ich habe es sehr oft erlebt und fast in jedem Unternehmen. Das halt auch das, was mich sehr überrascht hat, weil ich das von einem theoretischen Ansatz nie gedacht habe. Die meisten Unternehmen kennen Ihre eigenen Prozesse nicht. Die machen was und dann sagen die, ich sage jetzt mal überspitzt und sehr salopp: "Ich krieg was von dem und geb es dem." Aber den gesamten Prozess, mal analysiert oder vielleicht sogar grafisch, wie du das jetzt hier gemacht hast mit deinem Entscheidungsbaum, dass einfach mal die Unternehmensprozesse graphisch aufgezeichnet werden, das existiert fast nie. Das ist gewachsen und wird so gemacht zum Teil, weil es schon immer so gemacht wurde. Sobald man versucht jetzt, mit einer Softwarelösung, egal welcher Art, Standardisierung hineinzubringen und zu verstehen, was passiert denn eigentlich in unserem Abteilungsprozess, dann stößt man auf Grenzen. Das Problem ist also gar nicht so sehr die Umsetzung der Softwarelösung, sondern das, was davor gemacht werden muss. Nämlich die Aufbereitung der Prozesse und auch der Datenquellen, sage ich mal, damit man auch versteht, was machen wir hier eigentlich, von wo nach wo wandern unsere Daten und wie verarbeiten wir die. Und diese Vorarbeit wird aus meiner Sicht oft vernachlässigt. Man startet direkt mit der IT und dann steht man auf einmal vor einer Frage: "Soll ich jetzt linksrum gehen oder rechtsrum?" Und da muss auf jeden Fall, bevor man eine Softwarelösung einführt, sollte man sich diese ganze prozessuale Landschaft anschauen.

34

[0:24:19.1] **I: RS** Was ja da auch oft mit reinspielt. Du hast vorhin oft gesagt, du agierst in Matrixorganisation und hast dabei ja schon das Risiko mit angesprochen, dass sie eher ihrem Linienvorgesetzten berichten oder sich verantwortlich fühlen als dir. Jetzt ist es natürlich auch für mich spannend, was du auch für die geeignete Organisationsform hältst. Also ist es trotzdem eine Matrixorganisation, wegen, du müsstest dann mal Vorteile sagen? Oder würdest du eine andere wählen, wenn du aus deiner Erfahrung heraus sagen könntest, das könnte man anders machen?

35

[0:24:55.4] **B: CR** Also, die Frage ist nicht so leicht zu beantworten, weil so ein bisschen Wunschdenken gegen Realität steht. Und ich weiß, dass ich die Realität nicht ändern kann. Ich sage mal, wenn man eine projektorientierte Organisationsform hat, ist man in einer Art Systemhaus. Es gibt für einen Kunden ein Projekt und dann werden entsprechende Ressourcen zusammengezogen und sind auch voll und ganz dafür zuständig. Das ist natürlich eine Idealform, sage ich mal, dann ist man zu 100 % dabei und wird auch darüber den größten Output generieren und auch, ich sage mal, mit einer guten Qualität, also wird die eigentlichen Projektziele On Time, On Budget, On Scope, die wird man damit am ehesten erreichen. Aber in der Praxis, würde ich schon fast sagen, ist das utopisch, man wird vermutlich nie über eine Matrixorganisation hinauskommen, weil einfach die Unternehmen, in der Regel, sich dem Produktionsbetrieb, also den Operations, zugewandt haben und ein Projekt ist immer on Top. Also man muss verstehen, ein Projekt ist für die meisten Mitarbeiter eine Belastung. Die müssen auf einmal aus ihren gewohnten Operations raus, müssen sich mit neuen Inhalten oder neuen Vorgehensmodellen konfrontieren, haben oft auch eine Abwehrhaltung dagegen, weil es ja eine Änderung mit sich bringt. Also, auch das ist ein Thema, was man von der emotionalen Seite vorab beleuchten muss. Wie groß ist der Änderungswille in einem solchen Unternehmen, weil oft auch da Leute im Prinzip das blockieren können und auch die Stimmung

herunterreißen können. Vielleicht kannst du mir deine Frage, ich schweife wieder ab, merke ich gerade ein bisschen. Vielleicht kannst du mich noch mal konzentriert auf den Kern zurückbringen?

36

[0:27:07.9] **I: RS** Ich möchte gerne wissen, was du aus einer Projektmanagementsicht sagst, ist die beste Organisationsform bei der Einführung von einer No Code/ Low Code-Plattform? Also auch, weil du die Matrixorganisation selber angesprochen hast, hatte ich die als Beispiel angebracht. Ist es sinnvoll, die Mitarbeiter aus ihren Abteilungen herauszulösen oder macht es Sinn, die in den Abteilungen zu belassen, damit die an der Basis bleiben oder so was? Also, wie du es schon gesagt hast, die Projektmanagement-Schiene ist, sagst du, es ist ideal, das kannst du auch ruhig formulieren. Ich versuche ja hier, das bestmögliche Modell zu erfahren.

37

[0:27:46.2] **B: CR** Gut also, also danke erst mal, dass du mich auf die Spur zurückgebracht hast. Also ich denke, die meisten, zumindest in größeren Unternehmen, sind in Abteilungen organisiert und die werden auch in ihrem Daily Business dann in dieser Abteilung bleiben. Es macht am meisten Sinn, die auch dort während der Projektphase zu belassen, weil sie da das Fach-Know-how haben und sie einfach durch eine IT-Person vor Ort in dieser Abteilung zu unterstützen oder einen Coach, der sie da anlernen kann. Das ist für diese Art der Entwicklung das beste Vorgehen.

38

[0:28:24.4] **I: RS** Es gibt ja innerhalb von Projektmanagement gewisse Vorgehensweisen zum jetzigen Zeitpunkt. Das ist zum Beispiel Wasserfall oder ein agiles Vorgehen. Die Mischung wäre dann ein hybrider Ansatz. Welches Vorgehen würdest du bei der Einführung von einer No Code/ Low Code-Plattform, also einer neuen Softwarelösung, wenn man das so sagen will, bevorzugen? Eine von diesen dreien oder hast du da einen komplett neuen Ansatz, wo du sagst: "Ich würde das eher so machen"?

39

[0:28:57.5] **B: CR** Das kann man nicht, oder ich würde eher jetzt spontan zu einer hybriden Lösung tendieren. Aus meiner Sicht ist es so, dass eine rein agile Lösung im klassischen Sinne, ein reines Modell, unvermischt, laut Lehrbuch, findet man in den meisten Projekten nicht. Es wäre maximal in so einer projektorientierten Umgebung möglich, aus meiner Sicht, wo wir wirklich Experten haben, die auch in diesem Modus schon zusammengearbeitet haben. Die also genau wissen, was sie zu tun haben. Wo ihr Fokus darauf ist und noch in der Lage sind, selbsttempowert, sage ich mal, Entscheidungen zu treffen und, zum Beispiel, jetzt die Sprintziele festzulegen und das Ganze dann auch halt selbst motiviert voranzutreiben. Wir haben es jetzt hier aber, glaube ich, in einer No Code/ Low Code-Einführung eher mit einem Umfeld zu tun, wo eine gewisse Unerfahrenheit in IT-Projekten besteht. Also insgesamt in der Projektarbeit, weil man davon ausgehen muss, dass die Kollegen, die das machen, dann tatsächlich diesem Daily Business verhaftet sind, ihren täglichen Routinen. Also da fehlt die Projekterfahrung, deswegen wäre es eher ein Waterfall-Ansatz. Dass es jemanden gibt, der so ein bisschen den Rahmen festlegt und die nächsten Schritte, aber und das kommt dazu vom hybrid her, durchaus hybride Teile einstreut. Das ist, glaube ich, das was ich sehe, was sich bewährt. Gerade, wenn da keine Projektkultur sich schon entfaltet hat. Agile Elemente, die wirklich sich sehr bewährt haben, sind zum Beispiel, ein Daily zu machen, ein Daily Stand-up - 15 Minuten am Anfang des Tages und auch Retrospektiven sowie auch eine graphische Übersicht der Todos an einem Kanban-Board. Das sind Elemente aus dem agilen Leben, die man wunderbar in eine Waterfall-Logik einstreuen könnte. Das wäre aus meiner Erfahrung heraus, hierfür glaube ich ein machbarer, praktikabler Ansatz.

40	<p>[0:31:34.5] <b>I: RS</b> Also würdest du sagen, dass, wenn du dem jetzt ein Gewicht geben müsstest, es schon einen Schwerpunkt auf dem Waterfall gibt innerhalb des hybriden Modells, aber agile Tools in der Realisierungsphase, vermute ich mal, meinst du, dann zur Anwendung kommen? Also dort ein iterativer Gedanke mit eingestreut wird.</p>
41	<p>[0:31:51.8] <b>B: CR</b> Ja genau, das würde ich so sagen und Hintergrund ist einfach, dass sich da noch keine Projektkultur entfaltet hat bzw. nicht gelebt wird und dass die Kollegen da erst abgeholt werden müssen.</p>
42	<p>[0:32:03.7] <b>I: RS</b> Aber sagen wir mal, wir sind in einem sehr, sehr jungen, agilen Unternehmen, das schon voll dabei ist, wo die eingespielt sind, wo die auch schon erfolgreich ein agiles Management angewendet haben. Würdest du da sagen dort ist eher der agile Anteil im Übergewicht bzw. nur agiler Ansatz sinnvoll? Oder würdest du da auch sagen, trotzdem Waterfall?</p>
43	<p>[0:32:29.3] <b>B: CR</b> Ja, also ich stehe dem Agilen gar nicht Kontra gegenüber, im Gegenteil, ich finde das sehr gut. Ich würde in so einem Fall, es ist halt immer schwierig, wie lange man in dem Unternehmen arbeitet und wie sehr man das kennt. Es hängt aber wirklich sehr viel von den Kollegen ab, gerade der Kultur und dem Know-how. Wenn ich, angenommen, neu reinkommen würde, würde ich mich vorarbeiten. Ich würde sehr agil, glaube ich, anfangen und dann schauen, wie es läuft. Und wenn man merkt, es funktioniert nicht, dann könnte man da unterstützend, sage ich mal, eingreifen, indem man peu à peu anfängt, dann da eventuell einen Weg vorzugeben, bis zu einem Punkt, wo man sieht, ab da geht es wieder selbst dirigiert.</p>
44	<p>[0:33:19.9] <b>I: RS</b> Ich verstehe, was du meinst. Also mich interessiert jetzt natürlich, aber auch wenn du schon sagst: "Ja ich favorisiere halt diesen hybriden Ansatz." Was siehst denn du da direkt als Vorteil? Also, wenn du es jetzt mal auf Bulletpoints benennen müsstest. Was sind denn die Vorteile, die du dabei siehst, die ein Unternehmen daraus gewinnt, wenn es dieses Vorgehen anwendet?</p>
45	<p>[0:33:45.6] <b>B: CR</b> Also ich sage mal, der agile Anteil ist sehr wichtig für die, wie soll ich sagen, für die mentale Zufriedenheit der Kollegen, die es umsetzen. Diese Selbstbestimmung, diese Möglichkeit, Feedback zu geben. Das sind ganz wichtige Punkte, um einfach jetzt ein erfülltes Arbeitsleben zu haben. So, dem gegenüber steht natürlich auch der gewisse Projektdruck, den man hat, ich sage mal, Zeit und Budget hängen einem im Nacken. Das ist immer ein Problem. Mit der Zeit, das ist wie Raum und Zeit bei Einstein, die korrelieren miteinander. Überschreite ich die Zeit, überschreite ich quasi auch das Budget, das ist ein Punkt letztendlich. Und man hat natürlich auch die Umsetzungsqualität im Nacken, weil man natürlich als Projektmanager weiß, dass die schweren Probleme, die schweren Fehler, die werden ganz am Anfang gemacht. Das sind in der Regel Designfehler, konzeptionelle Fehler und hier, genau hier in diesem Punkt, halte ich das total wichtig, eine Waterfall-Schiene mit einzubauen. Einfach um Dinge, diese konzeptionellen Probleme relativ frühzeitig erkennen zu können. Als aber auch hier eine Direktive vorzugeben. Wo wollen wir denn hin und was sind unsere Meilensteine? Also diese Definition, damit wir auch wissen, was sind unsere Deliverables in der Zeit X. Das muss ganz klar auf dem Schirm sein und meine Erfahrung ist einfach, auch wenn man agil im Prinzip den Scope immer wieder entscheidet, das große Ganze, dieser Endtermin, das darf nicht aus dem Blick geraten und hier bedarf es schon einer klassischen Unterstützung von einem Projektmanager an der Stelle.</p>
46	<p>[0:35:55.0] <b>I: RS</b> Entschuldige, wolltest du noch etwas sagen?</p>

47	[0:35:58.9] <b>B: CR</b> Ich war durch. Ich wollte dich nur fragen, ob ich die Frage damit beantwortet habe.
48	[0:36:05.2] <b>I: RS</b> Ja, aber jetzt interessiert mich aber auch, siehst du dabei Nachteile, denn die Vorteile klingen ja schon sehr gut, aber siehst du auch Nachteile, auf die ein Unternehmen vielleicht achten sollte, wenn es diesen Weg beschreitet? Also, was schiefgehen könnte?
49	[0:36:17.6] <b>B: CR</b> Weg im Sinne von hybrid, oder?
50	[0:36:20.9] <b>I: RS</b> Im Sinne von hybrid, da es das favorisierte Modell ist, über das wir die ganze Zeit sprechen.
51	[0:36:25.1] <b>B: CR</b> Also, definitiv ist das auch das nicht die optimale Lösung. Ich kann dir jetzt einfach nur sagen, die Nachteile sind halt die, die jetzt auch die beiden Einzel- oder reinen Modelle mitbringen würde jeweils auf ihrer Seite. Also es gibt jetzt keinen zusätzlichen hybriden Nachteil. Wenn das die Frage so beantwortet, würde ich das so stehen lassen.
52	[0:36:56.8] <b>I: RS</b> Das beantwortet die Frage. Also ich würde das mal kurz für das Transkript zusammenfassen. Du siehst zwar die ganzen positiven Aspekte, also theoretisch ist ein hybrides Modell auch ja deswegen vorteilhaft, weil es ja viele von den Nachteilen ausgleicht, die die jeweiligen Modelle haben. Und du sagst, die Nachteile wären die gleichen und die würden sich in diesem hybriden Modell teilweise aufheben, aber sie wären trotzdem noch merkbar da.
53	[0:37:30.6] <b>B: CR</b> Wären da, ja.
54	[0:37:33.3] <b>I: RS</b> Dann habe ich das richtig zusammengefasst. Ich würde mich auch noch dafür interessieren, tatsächlich. Es ist ja in Projekten häufig auch so, dass während eines Projektes, also während der Einführung jetzt hier an dem Beispiel gesagt, das plötzliche Änderungswünsche kommen. Entweder von außerhalb des Projektes, dadurch dass Leute darauf aufmerksam geworden sind und sagen: "Das hätte ich noch gerne." Oder auch innerhalb des Projektes, wenn die Projektmitglieder merken: "Ok, hier ist gerade was nicht ganz so gut gelaufen. Das möchte ich gerne ändern." Wie würdest du diese Änderungen bearbeiten, behandeln, wie würdest du mit ihnen umgehen?
55	[0:38:18.5] <b>B: CR</b> Ich überlege gerade, ob es einen Unterschied macht, von wem eine solche Änderung herangetragen wird. Also ich sage mal, ob es einen Unterschied macht, ob es von intern kommt oder von extern. Ich glaube, es kommt gar nicht so sehr auf die Quelle drauf an, sondern auf die Art der Veränderung. Es gibt natürlich schwerwiegende Veränderungen, wo man im Prinzip alles umstoßen kann, als auch kleinere. Ich glaube, wichtig bei jeder Veränderung ist einmal, die erst mal zu analysieren. Was ist der Impact? Wenn man, ich sage mal, an das Projektteam herantritt, da ist eine Änderung, da fragt sich natürlich jeder: „Was machen wir jetzt damit?“ Das muss klar sein, dass man sich einen alternativen Weg klarmacht. Das kann jetzt je nach Methode einfach mal im Leitungsrahmen sein oder im Team, dass man sagt: "Hör zu, hier ist eine Änderung. Lasst uns darüber diskutieren über die Auswirkungen." Dass man diese Veränderung, wie auch immer, zusammen erarbeitet und die Konsequenzen klar auf dem Schirm hat. Zeitlich und so weiter, personell zum Teil auch, wenn man neue Leute braucht für neue Sachen. Und sobald das klar ist, sollte man das zurückspiegeln an denjenigen, der die Veränderung, den Change, initiiert hat und ihm sagen: "Hör zu, das sind dann die

Konsequenzen." Eventuell, je nach Projektform, dann auch mit einem Angebot, Änderungsangebot. In der Regel führt das zu einer Verzögerung. Das auch klar kommunizieren. So die klassischen Dinge. Aber gibt es, neben diesem klassischen Changemanagement, noch andere Themen, die da ganz speziell von Interesse sind?

56

[0:40:23.6] **I: RS** Du kennst ja agiles Projektmanagement. Dort wird jetzt zum Beispiel mit Änderungen umgegangen, in dem das in das Backlog, bei einem Scrum System, eingearbeitet wird und dann wird das in einem der Sprints mit abgearbeitet. Beim klassischen Waterfallmodell würde es ja, zum Beispiel, würde es ein Change Request geben. Das würde eingearbeitet in die Projektplanung, also dieser ganze lange Lifecycle würde ja damit wieder anfangen. Meine Frage zielt dahin gehend, wie in diesem hybriden Modell, was du ja für dich favorisierst, wie du da mit der Änderung umgehen würdest. Würdest du das in ein Product Backlog packen. Dass quasi ein Daily Scrum, das hast du ja auch gesagt, dass du das als sinnvoll empfindest, einarbeiten und dann abarbeiten. Oder sagst du: "Nee, das gibt ein Change Request und das wird über den langen Weg neu eingeplant." Darauf zielt meine Frage quasi.

57

[0:41:13.5] **B: CR** Danke für die Klarstellung. Es kommt tatsächlich beides vor und das hängt an der Gewichtigkeit der Änderung. Natürlich würde ich Änderungen, die im normalen Rahmen sind, einfach ins Backlog packen. Wenn bestimmte Sachen, das machen wir auch hier, wenn wir sagen, bestimmte Sachen können wir in dieser Etappe nicht durchführen, dann wird das ins Backlog zurückgeschrieben und dann ist das Thema durch. In meinem aktuellen Projekt habe ich auch mit größeren Sachen zu kämpfen. Größere Sachen sind für mich Fluktuation. Hier ist eine Kollegin, die bekommt bald ein Kind, da freue ich mich sehr für sie. Sie ist aber trotz allem spontan ausgefallen und wir hatten keinen Ersatz, weil unser Team so klein ist. Und das war jetzt eine Sache, da mussten wir wirklich eine große Welle schieben, weil diese ganze Veränderung bedeutet auch, dass, weil diese Kollegin auf dem kritischen Pfad unterwegs war, dass wir jetzt direkt eine Verzögerung haben, bis der neue Kollege eingearbeitet ist. Und das bedeutet in dem Fall, jetzt mal 2 Monate, von jetzt auf gleich. Und da muss man im Prinzip die entsprechenden Stellen informieren und eine Art, na ja, nicht Change Request in dem Fall, aber so eine Art Verzögerungsmeldung, mit dem Impact der zeitlichen Verzögerung, herausgeben. Also auch hier hat man, zumindest wenn man diesen hybriden Ansatz fährt, alle Optionen des klassischen als auch des agilen Managements dann zur Verfügung. Und das macht es wirklich sehr angenehm, weil man dann sehr, sehr angemessen auf den jeweiligen Change reagieren kann.

58

[0:42:57.6] **I: RS** Dadurch, dass du ja auf diesem hybriden Weg unterwegs bist. In Projekten gibt es ja immer verschiedene Rollen. Also klassisch Projektleiter, agil wäre ein Scrum Master, Product Owner, Project Owner, Sponsor, Development Team. Also diese klassischen Rollen gibt es ja. Welche wichtigen Rollen und Kompetenzen siehst du im Projekt, die du essenziell als nötig ansiehst, damit das Projekt ein Erfolg wird?

59

[0:43:32.1] **B: CR** Also definitiv die Rolle des Sponsors, ist natürlich ganz klar. Ohne Finanzierung geht es in der Regel nicht. Was ich als essenziell ansehe, neben dem Team natürlich, als in dieser Teamrolle ist ein Product Owner. Jemanden, der sich für das, was man entwickelt, dann auch verantwortlich fühlt und vor allen Dingen, der auch in der Lage ist zu sagen, wie er sein Produkt gestalten möchte. Also eine relativ klare Ansage. Weil, man kommt in der Entwicklung immer wieder an diese Entscheidungszweige oder Abzweige, wo man sagt: "In welche Richtung soll es jetzt weitergehen?" Und hier eine kompetente Person zu haben, die sagt: "Ich möchte das genau so haben" ist dann von absoluter Entscheidung oder entscheidender Bedeutung. Der Product Owner spielt für mich eine zentrale Rolle. Derjenige, den ich tatsächlich nicht so sehr im Fokus sehe, wäre die Rolle des Scrum Master. Vielleicht ist das auch so,



weil ich diese Rolle auch ausfülle, persönlich, kann sein, aber wir hatten in verschiedenen Projekten, die ich im Moment begleite, auch einen reinen Scrum Master und es fällt dieser Person, aus meiner Sicht, in der Realität immer wieder auf die Füße, dass sie zu wenig Fach-Know-how hat. Also rein den Prozess zu begleiten, ist schon fast, naja, ich will es nicht zu sehr abwerten, aber so einen Touch von Sekretärinnen-Arbeit, von Hilfsarbeit. Das ist jetzt zu hart ausgedrückt. Ich finde jetzt aber kein mittleres Wort dafür, aber es entwertet auch so ein bisschen diese Person. Die Rolle des Scrum Masters ist für mich so ein bisschen skurril manchmal. Die des Product Owners und des Sponsors hingegen vollkommen klar.

60

[0:45:37.1] **I: RS** Projektleiter?

61

[0:45:39.6] **B: CR** Ja, ist durchaus sinnvoll, je nach Größe. Aber eher so in der coachenden Rolle, so dass man im Prinzip eine Instanz hat, die bei kritischen Themen dann noch mal entscheidend darüber geht und noch mal klares Gewicht reingibt. So wie der Bundespräsident, so was in der Art. Der hat ganz kraftvolle Worte, aber am Ende des Tages ist er jetzt nicht der Entscheidungsträger. So eine Rolle ist wirklich gut, wenn man die im Projekt noch besetzen kann. Das auch jeder weiß, zur Not kann ich mich an den wenden. Also der hätte dann neben der Projektleiterrolle noch andere Rollen impliziert. Die des Schlichters zum Beispiel, des Beraters.

62

[0:46:33.1] **I: RS** Siehst du außerhalb des Projekts wichtige Stakeholder, die auf jeden Fall an Bord sein sollten? Jetzt nicht im Projektteam direkt mit drin, aber die in der Kette mit dabei sein sollten.

63

[0:46:44.7] **B: CR** Definitiv. Also zum einen sind es natürlich definitiv die ganzen Fachexperten, die man aus der technischen Sicht benötigt. Ich sage jetzt mal ein paar Beispiele: Netzwerker, Datenbankadmin, Serverspezialist, so diese harten technischen Rollen, die sind in der Regel nicht im Projekt selber benötigt, aber die können einen entscheidenden Einfluss haben, wenn, zum Beispiel, ich habe das jetzt erlebt wieder, da haben wir in der Cloud ein System aufgesetzt, browserbasiert und dann sollte das zu einem Single-Sign-On umgebaut werden und nachher stellte sich heraus, dass die URL eine ganz spezielle Konvention haben muss. Also da gab es im Prinzip eine ganz harte technische Vorgabe, damit man einen Single-Sign-On einführen kann und deswegen diese wichtigen Dinge Netzwerkdaten, Servertechnik, da müssen die Leute Stand-by sein. Das zum einen, zum anderen ist es aber auch wichtig, organisatorisch, also solche Nicht-Projektrollen, da zu haben. Jemanden, der, zum Beispiel, sagt, was jetzt betriebsratentechnisch möglich ist oder wo da die Grenzen sind. Es ist wichtig. Ich überlege gerade, welche externen Einflüsse wir in der Regel noch haben.

64

[0:48:19.2] **I: RS** Du kannst auch an dein eigenes No Code/ Low Code-Projekt zurückdenken. Wer war da von Vorteil? Was war gut, dass der zumindest in der Chain of Command irgendwo mit dabei war und informiert war.

65

[0:48:38.7] **B: CR** Ich kann es jetzt eher so allgemein formulieren, dass es so total sinnvoll ist, über seinen eigenen Projekttellerrand zu schauen, weil ich halt merke, dass andere, sagen wir mal, Teilsystemverantwortliche noch einen ganzen anderen Blick auf das eigene Projekt haben. Weil das für deren Systeme, die stehen in Wechselwirkung oft miteinander, wenn dann so ein Projekt fertig wird, eine Software an den Start geht, dann beeinflusst das auch andere Systeme. Und diese Personen, die diese anderen Systeme unter ihrer Verantwortung haben, mit denen sich vorab auszutauschen und deren Sichtweise zu kennen, weil, das spiegelt auch das eigene Handeln wider oder die eigene Vorgehensweise, was da wichtig ist und was nicht. Also

einfach, ich sage mal so, dass man sich vorab Gedanken macht, wenn diese Änderung eingeführt wird, wie die Auswirkungen sein können auf die Wechselwirkung im Unternehmen, weil, jedes Teilsystems beeinflusst das andere. Und oft reden die Leute oder Abteilungen nicht gut genug miteinander, wissen nicht voneinander Bescheid und sind ganz überrascht, was da geschieht alles. Das vorab zu klären, das halte ich für eine ganz wichtige Aufgabe. Das hilft auch, das man sich nicht verrennt und in den falschen Zweig reinrennt und am Ende alles wegwerfen muss.

66

[0:50:15.5] **I: RS** Bei dieser Vernetzung mit den anderen Abteilungen, meinst du dann damit die Abteilungsleiter oder eine spezifische Person in den Abteilungen, die dann informiert wird oder ist das egal?

67

[0:50:26.6] **B: CR** Also sinnvoll schon, auf Leitungsebene, weil meistens die Abteilungsleiter auch nur den Überblick haben, was bei Ihnen in der Abteilung alles passiert. Auf jeden Fall sinnvoll halt mit den Linienverantwortlichen, sage ich mal, die würde ich immer mit ins Boot holen. Weil die ja in der Regel auch später dann diesen operationalen Betrieb verantworten müssen von der Software, die innerhalb eines Projektes entwickelt wird. Und das größte Problem, was die halt haben, ist, dass das Projektteam dann weg ist, wenn das Projekt dann abgeschlossen wird, aber die das restliche Leben lang mit den Ergebnissen zu tun haben werden. Also hier sehe ich die größten Hindernisse und die größten Chancen mit für so ein Projekt, dass man frühzeitig an die herantritt, die damit später involviert sind. Was mir jetzt noch so als Nebenstakeholder oder Stakeholder mit einfällt, sind, ganz wichtig auch vorab, die Themen Datenschutz, IT-Security und es gibt je nach Unternehmensgröße die verschiedensten Abteilungen. Es gibt zum Beispiel Unternehmen, die haben so eine Art operational resilience dabei. Das bedeutet, was passiert, wenn dieses Systems dann auf einmal ausfällt. Die sind da die SLA's. In welcher Zeit muss wer welche Reparatur durchgeführt haben. Also diese Gedanken, so ein Target Operation Modell aufzusetzen. Wie das später im Betrieb aussieht. Das sollte man auf jeden Fall in der Projektphase schon anfangen und alle die, die dann später involviert sind, sei es ein User Help Desk, diese Software geht kaputt und irgendjemand möchte jemand qualifiziert anrufen, der ihm dann hilft. Dann muss dieses User Help Desk informiert werden. Also es ist von Unternehmen zu Unternehmen und der Größe unterschiedlich und dem, was man macht. Ganz wichtig, aus meiner Sicht, in der Projektphase schonmal sich Gedanken machen, wie es später in der Business as Usual- Phase aussieht. Wer dann für Störungen verantwortlich ist? Wie lange die Reparaturfristen sind? Ist das vertraglich abgesichert? Dafür die richtigen Stakeholder herauszufinden, das ist schon ganz wichtig.

68

[0:53:09.3] **I: RS** Jetzt hast du ja gesagt, du hast auch schon eine Plattform eingeführt. Jetzt ist es natürlich meisten nicht so, dass man das direkt im ganzen Unternehmen auf einen Schlag einführt, sondern man fängt ja meistens in einer Prototypen-Abteilung an und skaliert das dann in das Unternehmen hinein. Wie ist das damals dann bei dir passiert? Wie hat diese Skalierung funktioniert? Kannst du das mal ganz grob umreißen?

69

[0:53:35.4] **B: CR** Ja, in der Regel ist es so, man ist erstmal isoliert mit seiner Entwicklung. So ein Projekt ist ja abseits des Daily Business und ab einem gewissen Punkt hat das System, das Testentwicklungssystem, eine Reife erhalten, in dem man einen acceptance test durchführen kann. Spätestens an diesem Punkt macht es Sinn, da so einen Prototyp vorzustellen und auch bestimmte Leute einzuladen, daran teilzuhaben, daran teilhaben zu lassen, in dem man denen zum Beispiel einen Testaccount verschafft, damit die das dann weitergeben, diese Information über die Systemeinführung. Wichtig ist hier erst mal, natürlich ein Produkt zu schaffen, was den Vorgaben entspricht und dass die Leute, die in dieser Entwicklung dabei sind, natürlich immer abgeholt werden just in time. Das ist klar, aber wenn es jetzt um die

unternehmensweite Einführung geht, man in dem Moment, wo man ein vorzeigbares, sage ich mal, Deliverable hat. Also muss nicht das Gesamtprodukt sein. Diejenigen, die es später betrifft, schon frühzeitig abholt und denen anhand eines Beispiels oder eines Testaccount oder so zeigt, was das bewirken wird. Meine Erfahrung ist hier, dass reine Worte oft links rein, rechts rausgehen. Also ohne visuelle Unterstützung, im Idealfall durch einen Account oder Testaccount oder halt durch eine PowerPoint etwas aufbereitet, die abholt und dann mit teilhaben lässt an der Entwicklung, soweit das möglich ist, durch eigene Gedanken so, und dann, wenn die Leute, die Kollegen, mental abgeholt sind, dann kann man das auch unternehmensweit einführen.

70

[0:55:55.5] **I: RS** Das spielt ja auch ein bisschen in die Frage von Befähigung von Mitarbeitern ein, denn im Endeffekt sollen ja in den Fachabteilungen später Leute das anwenden. Wie würdest du diese Befähigung gestalten?

71

[0:56:09.2] **B: CR** Ja, das ist ein schwieriges Thema, weil also die Erfahrung ist, dass oft der Horizont von Fachkollegen eingeschränkt ist. Ich möchte niemanden zu nahe treten, nicht falsch interpretieren meine Worte, aber ich merke halt, man ist oft das links und rechts und diejenigen, die ein neues System oder später mit einem neuen System arbeiten werden, sollten wirklich frühzeitig da abgeholt werden. Und man wird auch mit relativer Sicherheit da Widerstände spüren. Das bedeutet, hier ist aus meiner Sicht, steter Tropfen höhlt den Stein, die beste Strategie. Man würde sagen, dass man in einem gewissen Zyklus die Änderung vorstellt und auch die Kollegen vom Fachbereich mit einbindet, soweit das geht. So dass die auch aktiv da mitarbeiten können. Sobald das der Fall ist, ist auch die Akzeptanz groß und man lernt dann auch Learning by doing, in dem Fall, und natürlich klassischerweise würde man noch eine Schulung geben, bevor es dann losgeht mit dem neuen System. Aus meiner Erfahrung ist es so, wenn man jetzt sagt, ich plane jetzt für die gesamte Abteilung einen halben Tag Schulung ein, dann sind die fit. Das funktioniert nicht, da sind Ängste dabei bei den Leuten. Können die das leisten? Wie sieht der Prozess aus? Was passiert, wenn? Das sind alles Fragestellung, die dann so hochkommen. Die kann man in der Regel nicht beantworten mit so einer Frontalbeschallung. Es macht mehr Sinn, wenn möglich, die vor Ort mit einzubeziehen und diese Dinge im täglichen Rahmen, auf kleiner Basis zu klären und dadurch eine Akzeptanz zu schaffen. Auf jeden Fall: Stichwort hier ist, mit einbeziehen und nicht nebenher entwickeln und dann sagen: "Tam tam, die sind fertig. Jetzt wird alles gut. Seht her." Das wird in der Regel nicht funktionieren, da ist zu viel Widerstand da.

72

[0:58:22.1] **I: RS** Nach der Einführung von so einem Projekt und der Skalierung kommt natürlich irgendwann auch die Frage der Qualitätssicherung. Wie man die Qualität des Projekts generell sichern kann und wie man den Erfolg am Ende misst? Also einerseits Qualitätssicherung, wie sicherst du die Qualität des Projekts während des Projekts? Und wie misst du am Ende den Erfolg, ob das alles geklappt hat?

73

[0:58:47.5] **B: CR** Ja, das geht stark in die Qualität dessen, was man, also ich muss mich kurz sortieren. Also natürlich, messbare Größen sind natürlich immer Zeit und Budget. Das ist am einfachsten. Das ist auch das, was in der Regel die Stakeholder abfragen, weil das wirklich einfache Größen sind. Jetzt geht es um den Scope und die Qualität dessen, was man erstellt hat. Und das ist gerade bei einer Einführung eine ganz spannende Sache. Also man sollte realistisch davon ausgehen, dass die Qualität auf gar keinen Fall 100 % hat, wenn man ein Softwaresystem in die Produktion einführt. Sondern es wird immer irgendwas geben, was man nicht bedacht hat, weil es einfach von der Komplexität oder von der Zeit her oder den fehlenden Ressourcen nicht anders machbar war. Es wird immer Dinge geben, die als Workaround starten. Wo man sagt: "Wir haben das jetzt gerade so hinbekommen. Hoffentlich klappt alles."

Diese Dinge erfordern bei der tatsächlichen Produktionseinführung viel Aufmerksamkeit und das muss man vorab einplanen. Also, ich kann jetzt nur so ideale Sachen geben. In der Regel ist so ein Projektteam ja schon durch, wenn es gerade in der Projektendphase viel gearbeitet hat. Die Leute sind erschöpft, ausgelaugt und wenn dann direkt die Einführung kommt, was oft der Fall ist, dann wirds schwierig. Dann hat man ein Team, was schon so ein bisschen durch ist. Man sollte ganz offen damit umgehen, sagen: "Wir haben jetzt ein neues System. Wir erwarten hier Kinderkrankheiten. Das bedeutet für alle, ihr müsst jetzt Überstunden schieben. Ihr müsst diese Abläufe, diese neuen Automatismen, nachkontrollieren. Funktioniert das so? Ist alles dran gedacht und ihr steht zumindest bereit, für den Fall, dass was sein kann." Und dann kann man jetzt dieses Level, sage ich mal, an Aufmerksamkeit, das würde ich bei der Einführung hochhängen und dann peu à peu runterfahren. Je nachdem, was dann tatsächlich passiert oder nicht, wenn man sieht, ok, es funktioniert alles so weit. Nichts ist abgebrochen, dann kann man anfangen und sagen: "So jetzt könnt ihr euch langsam entspannen." Aber die Spannung sollte schon sehr hoch sein, mit der Erwartungshaltung, dass es halt nicht funktioniert alles, zumindest alles bei der Einführung. Beantwortet das deine Frage, muss ich jetzt fragen?

74

[1:01:35.1] **I: RS** Also, wie du Qualität von einem Projekt sicherst, hat es beantwortet. Möchtest du noch irgendwas zum Messen von Erfolg sagen? Oder sagst du, diese harten Fakten Kosten, Zeit, Budget, das ist eigentlich das, was für mich das ist, woran ich essenziell den Erfolg des Projekts messe?

75

[1:01:52.7] **B: CR** Ja, gute Frage. Was haben wir denn noch als Erfolgsfaktoren? Also diese harten Fakten sind schon in der Regel auch ausschlaggebend. Es gibt dann viele weiche Faktoren. Wie zufrieden sind die Leute? Das merkt man dann danach, wenn man sich mal das Team und die Stakeholder hinterher anschaut. Ist es gut gelaufen, sind alle glücklich? Es ist halt keine wirklich messbare Größe. That's it, mehr fällt mir gerade tatsächlich nicht ein.

76

[1:02:32.4] **I: RS** Alles gut, wie gesagt, es geht ja um deine Erfahrungswerte und wenn du sagst: "Mehr gibts da jetzt hart so für mich nicht, dann gibt es hart für dich mehr nicht." Dann sind wir jetzt quasi auch schon fast am Ende von unserem Interview. Meine letzte Frage wäre jetzt einfach nur noch. Hast du Anmerkungen zum Interview? Best Practices to share? Oder Sachen, wo du sagst: "Ist im Interview jetzt nicht vorgekommen, aber das würde ich gerne noch sagen, weil mir das besonders wichtig ist."

77

[1:02:54.4] **B: CR** Also, erst mal das Interview, so wie du das geführt hast, fand ich sehr gut. Vielen Dank dafür, dass du das auch so gut vorbereitet hast mit deinem Entscheidungsbaum. Das war auch teilweise eine kleine Hilfe für mich. Was ich noch unbedingt sharen oder teilen möchte? Ich kann jetzt nur rein aus der Projektmanagementsicht oder aus meinem täglichen Business reden, da spielt natürlich viel auch eine Rolle von diesen ganzen Hard Facts. Jetzt dieses Kennen von Systemen, das Kennen von Prozessen, Abläufen und der Technik, aber entscheidend ist tatsächlich und das merke ich immer wieder im täglichen Leben, ist eher schon so die emotionale Seite. Also so diese Softskills, weil die in der Regel das Thema haben, dass die Kollegen für ihre Themen Experten sind. Die wissen genau Bescheid. Es gibt nichts, was ich da irgendjemand sagen kann in der Regel. Das ist auch gar nicht mein Ziel, aber woran das halt scheitert, sind diese rein menschlichen Themen. Da redet A nicht B oder, ein absoluter Klassiker ist: A schreibt B eine E-Mail. B antwortet nicht, weil er denkt: "Ist ja gar nicht mein Thema." Also er gibt einfach kein Feedback oder vergisst es. A wartet, die Zeit verstreicht und so gehen immer wieder ganz viele gute Sachen verschütt und die dann wieder rauszuholen, dass voranzubringen, das ist wirklich schwierig. Weil viele auch ein bisschen auf Zeit spielen, sage ich mal und das ganze Thema kann man, glaube ich, nur dann dem begegnen, wenn man da ein gutes Klima schafft. Ein freundschaftliches Klima, wo die Leute auch

aktiv Lust haben, miteinander zu reden, sich auch auszutauschen. Das ist, glaube eine der wichtigsten Aufgaben mit. Dass diese Seite, diese nicht messbare Seite, gut passt und dann könnte es auch ein Selbstläufer werden. Was in der Regel nicht der Fall ist, das ist auf jeden Fall eine wichtige Voraussetzung. Das war ein kleiner Nachtrag von mir persönlich.

78

[1:05:27.1] **I: RS** Alles gut. Dann danke ich dir ganz sehr für das Interview und ich würde jetzt die Aufzeichnung beenden.

79

[1:05:32.9] ENDE

## Einwilligungserklärung Interview

Ich erkläre hiermit mein Einverständnis zur Nutzung der personenbezogenen Daten, die im Rahmen des folgenden Gesprächs erhoben wurden:

Datum des Interviews	08.07.2022
Namen der zu interviewenden Person	Christian Rohrbeck
Name der interviewenden Person	Rocco Schenk
Kurzbeschreibung des Forschungsprojekts	Einführung von No Code/ Low Code aus Sicht des Projektmanagements
Durchführende Hochschule	Hochschule Mittweida – University of Applied Science

Die Daten werden im Rahmen eines mündlichen Gesprächs erhoben, das mit einem Aufnahmegerät und Videoaufnahmegerät aufgezeichnet wurde. Zum Zwecke der Datenanalyse werden die mündlich erhobenen Daten verschriftlicht (Transkription), wobei die Daten nicht anonymisiert werden.

Vorhandene Kontaktdaten, der interviewten Person, werden aus Dokumentationsgründen in einem separaten Schriftstück lediglich den Gutachter\*innen der wissenschaftlichen Ausarbeitung zur Verfügung gestellt. Nach dem Abschluss des Projekts werden diese Daten gelöscht.

Der Speicherung der personenbezogenen Daten zu Dokumentationszwecken kann durch die interviewte Person jederzeit widersprochen werden. Die Teilnahme an dem Gespräch erfolgt freiwillig. Das Gespräch kann zu jedem Zeitpunkt abgebrochen werden. Das Einverständnis zur Aufzeichnung und Weiterverwendung der Daten kann jederzeit widerrufen werden.

Christian Rohrbeck

Vorname und Name in Druckbuchstaben



Unterschrift

13.07.2022, Köln

Datum, Ort

## Anlagen 7, Interview EB06

RS: Rocco Schenk

MH: Moritz Hesse

1	[0:00:00.0] <b>Start (MH - Moritz Hesse/ RS - Rocco Schenk)</b>
2	[0:00:05.6] <b>I: RS</b> Dann, zum Anfang, ich begrüße Sie erst mal. Ich würde das Interview jetzt aufzeichnen. Sind Sie damit einverstanden?
3	[0:00:13.5] <b>B: MH</b> Ja, damit bin ich einverstanden.
4	[0:00:15.8] <b>I: RS</b> Ihre Daten werde ich zum Zwecke der Transkription aufzeichnen und dann aber wieder löschen. Dann würde ich direkt einsteigen wollen in das Interview und Sie mal bitten sich ganz kurz vorzustellen und vielleicht den Firmenhintergrund dazu, einfach, um Sie einzuordnen im wissenschaftlichen Kontext.
5	[0:00:33.5] <b>B: MH</b> Ich bin Moritz Hesse und arbeite zurzeit als Technical Consultant bei der Firma smapOne. SmapOne ist ein Anbieter von einer Low Code / No Code-Plattform, mit der User eigene kleine Applikationen erstellen können per Drag-and-drop und ich berate Kunden in, ja um unser Produkt herum, insbesondere Integrationsszenarien und betreue den Kunden auch im Rahmen der Integration, der Integrationsvorhaben und setze die Integration entsprechend um.
6	[0:01:07.8] <b>I: RS</b> Können Sie noch kurz etwas zu Ihrer Position sagen. Sie haben auch Projekte geleitet? Sie haben Projektmanagement-Erfahrung?
7	[0:01:17.1] <b>B: MH</b> Ja, allerdings nicht im Low Code Bereich. Ich hatte im Vorfeld zahlreiche Kontakte mit Projekten, sowohl auf fachlicher Seite, dass ich dort als quasi Projektmitglied mitgearbeitet habe, als auch in projektleitenden Funktionen, aber wie gesagt, da ging es um andere Themen als um Low Code, also Low Code-Projekte habe ich noch nicht betreut, also jedenfalls nicht in einer großen, quasi in einer Firma, die ein Low Code-Projekt umsetzen möchte, was ich natürlich mache hier, Low Code-Projekte für Kunden umzusetzen.
8	[0:01:47.2] <b>I: RS</b> Das geht nur darum, dass Sie die Expertise haben, ja auch Antworten zum Projektmanagement zu stellen. Meine erste Frage wäre, wie würden Sie No Code/ Low Code definieren?
9	[0:01:59.8] <b>B: MH</b> No Code / Low Code würde ich definieren als eine Reifestufe in der Softwareentwicklung, wo der User nicht mehr vollständig darauf angewiesen ist, einen eigenen Quellcode schreiben zu müssen, sondern dass man in der Lage ist, mithilfe von Oberflächen, interaktiven Oberflächen, sich zum Beispiel Applikationen per Drag-and-drop zusammen zu klicken und nicht drauf angewiesen ist hier mit ja entsprechenden Quellcode, Programmiersprachen, Frameworks etc. zu arbeiten, sondern das halt mit entsprechender, ja grafischer Oberfläche, zu gestalten.
10	[0:02:40.2] <b>I: RS</b> In dem Zusammenhang, haben Sie schon mal von dem Begriff Citizen Development gehört?
11	[0:02:45.5] <b>B: MH</b> Ja, auf jeden Fall.

- 12 [0:02:48.1] **I: RS** Und wie würden Sie das in dem Konzept Low Code / No Code jetzt einordnen, Citizen Development?
- 13 [0:02:53.6] **B: MH** Also vielleicht noch mal ganz kurz mein Verständnis von Citizen Development, das beschreibt einfach, dass die Entwicklung der erforderlichen Anwendungen dort durchgeführt wird, durchgeführt werden kann, mit Hilfe von z. B. No Code/ Low Code-Lösungen, wo die Probleme tatsächlich anfallen, also der Begriff, also quasi auf Basis der Bürger und nicht auf Basis irgendeiner höher geschalteten Autorität oder höher geschalteten Instanz, die diese Applikationen umsetzt, sondern das ist tatsächlich Bottom up von den Bürgern oder, im unternehmerischen Kontext, von den fachlichen Mitarbeitern selber entwickeln kann.
- 14 [0:03:33.7] **I: RS** Meinen Sie mit fachliche Mitarbeiter IT-Mitarbeiter?
- 15 [0:03:36.7] **B: MH** Nein, sondern ganz im Gegenteil die fachlichen Mitarbeiter aus den Fachabteilungen, aus den IT-fremden Abteilungen, also aus dem Einkauf, Warenwirtschaft, Materialhaltung. Also die IT-fachfremden Abteilungen. So dass nicht die IT dafür verantwortlich sein muss, die Applikation zu entwickeln, sondern die Anwender selber.
- 16 [0:03:54.6] **I: RS** Okay und welche Vorteile sehen Sie dann, wenn das so geschieht?
- 17 [0:03:59.7] **B: MH** Nun ja, ich komme selber aus der Softwareentwicklung, also bin selber auch langjährig in der Softwareentwicklung tätig gewesen und auch zu Zeiten, wo es noch nicht agile Software-Entwicklungsverfahren gab. Das heißt, ich kenne noch die sehr klassischen Verfahren, wie zum Beispiel das Wasserfallmodell. Und derartige Modelle sind ja geprägt, das zeigen die Vorteile von so agilen Methoden auf. Dass sie sehr zäh sind, sehr langwierig sind. Und insofern sind Vorteile von solchen Low Code-Lösungen oder auch von solchen Initiativen, wie Citizen Development, dass man die Notwendigkeit, einfach von langfristigen IT-Projekten einerseits, aber auch andererseits überhaupt die erforderliche Abstimmung mit der IT, dass die außen vor bleibt und dass letztendlich jeder seinen eigenen unmittelbaren Bedarf mit einer No Code/ Low Code-Lösung decken kann. Das sehe ich als unmittelbaren Vorteil. Ich selber bin Wirtschaftsinformatiker von der Historie und ich weiß, das ist quasi so ein wesentliches Charakteristikum von Wirtschaftsinformatikern, dass sie gerne als Dolmetscher zwischen Fachabteilung, also IT fremden Abteilungen und der IT wiederum fungieren, weil sie in der Lage sind, einerseits betriebliche-fachliche Prozesse zu verstehen, andererseits aber auch IT-Zusammenhänge, technische Zusammenhänge und insofern sind Wirtschaftsinformatiker immer gut geeignet, um ebendiese fachlichen Vorgänge technisch zu beschreiben. Und das ist aber immer ein schwieriger Prozess und diesen Vorgang kann man sich dann ja sparen, weil der Fachmitarbeiter muss dem IT'ler nicht erklären, was seine Anforderungen sind, sondern er geht einfach auf die Lösung zu und die Low Code/ No Code-Lösung zu und sieht, was er benötigt und richtet sich das halt alles selber ein und das ist ein absoluter Vorteil, dadurch entsteht unglaublich hohe Geschwindigkeit. Und daraus natürlich weitergeleitete Effekte, weitergeleitete betriebliche Effekte.
- 18 [0:06:04.2] **I: RS** Zum Beispiel?
- 19 [0:06:05.6] **B: MH** Nun ja, daraus lässt sich eine Erhöhung, viel höhere Produktivität zum Beispiel, ableiten. Also wenn ich jetzt sage, eine Applikation ist in der Lage, meinen Prozess zu beschleunigen. Ja, dann habe ich ja die höhere Effektivität durch die Lösung, wenn ich jetzt aber sage, die Lösung braucht einen unglaublich hohen Aufwand, um hergestellt zu werden, dann ist quasi mein Produktivitätseffekt, den ich daraus gewinne, mit einem unglaublichen hohen komplexen Aufwand verbunden, den ich für die Lösung quasi bereitstellen muss. In dem ich aber einen sehr geringen Aufwand nur habe, um die Lösung mir zu entwickeln, habe ich diesen Produktivitätseffekt einfach viel schneller erzeugt, das heißt, die Effektivität kann



	ich in beiden Fällen gewährleisten, die Effizienz aber, um zu dieser Lösung zu kommen, ist bei Low Code / No Code viel, viel höher.
20	[0:06:47.4] <b>I: RS</b> Sie haben jetzt viele Vorteile genannt. Also würden Sie auch Nachteile sehen bei der Einführung von No Code/ Low Code-Plattformen?
21	[0:06:56.9] <b>B: MH</b> Absolut, sehe ich auch im täglichen Geschäft in Anwendungen der Kunden unserer Lösung und zwar ist es so, wenn man quasi von der grünen Wiese, nein, das stimmt auch nicht unbedingt, aber man muss ein gewisses Maß an Struktur mitbringen, wenn man eine Software erstellt. Auch diese Aussage ist pauschal sehr schwierig zu bewerten man kann auch sehr dreckige Scripte schreiben, aber nichtsdestotrotz man muss sich anders mit Struktur und Modellierung und fachlichen Zusammenhänge zwischen den einzelnen Entitäten auseinandersetzen. Was bei Low Code und No Code-Lösungen möglicherweise auch gefordert wäre. Die Anwender bringen das aber häufig gar nicht mit und das liegt auch in der Charakteristik, quasi in der DNA von den Produkten selber, dass man ja eben genau diese Abstraktion haben möchte. Also man möchte den Nutzer ja gar nicht mit all diesen Dingen belasten, wie keine Ahnung, Entity Relationship-Modell. Was sind denn da für Zusammenhänge zwischen irgendwelchen Entitäten? Genau das möchte man dem Nutzer ja erleichtern. Aber das führt dann auch häufig zu, oder ich sehe hier die Gefahr einfach, dass sich diese Strukturlosigkeit, auch in den entwickelten Lösungen letztendlich wiederfindet und dass natürlich hier ein gewisser Wildwuchs entsteht, den es durchaus, ja ich sage mal, betrieblich würde man sagen, den es zu managen gilt. Also zu beobachten, zu planen, gegebenenfalls Richtlinien zu erstellen, Konventionen vorzugeben. Das ist durchaus eine Gefahr oder Nachteil, die ich hier sehe. Und insbesondere, wenn die Lösungen dann doch irgendwie wieder an die IT gereicht werden, also der User kann sich zwar selber seine App irgendwie zusammenbauen, aber jetzt soll die nicht als Insellösung irgendwo stehen, sondern wenn er da irgendwie die Daten eingegeben hat, sollen die dann bitte doch an umliegende Systeme weitergeleitet werden, die Daten. Und spätestens dann trifft man ja doch wieder auf die IT und spätestens dann gilt es doch wieder die Strukturen zu finden, die entsprechenden Datentypen zu wrappen, alles so was und die quasi Schnelligkeit und Einfachheit und Flexibilität, die der Anwender in der Lösung hat, schlägt sich dann später bei der Integration, kann die dann wieder auf die Füße fallen, weil dann eben dann genau das, ja, dass was letztendlich der Vorteil der Lösung für den Anwender ist, kann sich dann als Nachteil wiederum ausgestalten bei der technischen Implementierung und insbesondere Integration.
22	[0:09:41.1] <b>I: RS</b> Also den Nachteil, den Sie gerade beschrieben haben, würden Sie den unter dem großen Feld Governance quasi zusammenfassen?
23	[0:09:50.9] <b>B: MH</b> Durchaus ja, ja das ist durchaus eine Governance-Aufgabe, oder eine gute IT Governance kann helfen, diese Probleme auf jeden Fall, ja, ich möchte nicht sagen zu verhindern, aber es abzuschwächen. Ja, dass man die Mitarbeiter entsprechend schult oder vielleicht auch die Technik, dass die Lösung, die Low Code-Lösung vielleicht sogar auch die Möglichkeit bietet, irgendwelche Best Practice zu implementieren oder Constraints oder irgendwelche Restriktionen zu hinterlegen. Ja, weiß ich nicht, dass eine bestimmte Komplexität nicht überschritten wird oder so. Aber ja, würde ich sagen, Governance ist hier ein Stichwort, was passen würde.
24	[0:10:29.4] <b>I: RS</b> Und im Kontext der Vor- und Nachteile, die sie gerade gesagt haben, was würden Sie sagen, sind realistische Erwartungen, die man heutzutage an solche Low Code / No Code- Plattformen setzen kann? Was können die heute schon leisten?
25	[0:10:43.2] <b>B: MH</b> Es gibt eine sehr breite Bandbreite aus No Code-Lösungen. Wir von smaOne sind ein Ende, haben unsere Stärken und unsere Schwächen im Vergleich zu Wettbewerbern. Und diese Unterschiede, deshalb fällt es mir jetzt schwer, für die Gesamtheit aller Lösungen zu sprechen im Bereich No Code/ Low Code.

- 26 [0:11:12.9] **I: RS** Aus Ihrer Erfahrung heraus gesprochen.
- 27 [0:11:14.8] **B: MH** Ich würde mal grundsätzlich behaupten, mit No Code/ Low Code-Lösung kann man mit Sicherheit heutzutage schon einen enorm großen Anteil von betrieblichen Prozessen abbilden. Ich würde aber behaupten, wenngleich ich das als eine unbestätigte oder unüberprüfte Hypothese kennzeichnen wollte, dass No Code und Low Code-Lösungen durchaus dort besser geeignet sind, einsetzbar sind und zu einer erfolgreichen Einführung führen, wo es nicht um hochoperative Prozesse geht. Also um die wesentlichen wertschöpfenden Prozesse, sondern eher um die Auxiliar-Prozesse, also bestimmte Hilfsprozesse, kleine Prozesse, die vielleicht auch nicht ständig und in dem standardisierten Wertschöpfungsmodell stattfinden, sondern die vielleicht eher mal eben schnell stattfinden müssen, also wo man mal schnell eine Datenerhebung benötigt. Wo man mal schnell irgendwie eine kleine Zwischenlösung benötigt, meine Erfahrung würde eher dagegensprechen, das wirklich für hochstandardisierte Industrieprozesse anzuwenden, da würde ich sagen, sind die Skaleneffekte für ein hochprofessionell entwickelten und auch überwachten Softwareartefakt, was man da einsetzt, durchaus geeigneter als so Low Code-Lösungen. Ich sehe den Vorteil eher bei diesen Hilfsprozessen, bei diesen Sekundärprozessen, bei diesen Auxiliar-Vorgängen.
- 28 [0:12:57.1] **I: RS** Alle nachfolgenden Fragen sind immer im Zusammenhang mit der Einführung einer Low code-Plattform in einem Unternehmen. Wenn Sie jetzt Ihre Low Code-Plattform zum Beispiel in einem Unternehmen einführen, welche Voraussetzung muss ein Unternehmen mitbringen, bevor sie sagen, „okay, das führen wir jetzt ein?“ Wenn Sie möchten, kann ich die Frage auch noch weiter eingrenzen, aber ich würde die erst mal so offen lassen.
- 29 [0:13:25.7] **B: MH** Welche Voraussetzungen wir als Firma voraussetzen, dass Kunden unser Produkt einsetzen können?
- 30 [0:13:31.0] **I: RS** Ihre Low Code-Plattform. Oder aus Ihrer Erfahrung heraus die Plattform, je nachdem.
- 31 [0:13:33.6] **B: MH** Ja, also ich sage jetzt mal, tatsächlich harte Voraussetzungen sind eigentlich nur technische Systemvoraussetzung um die Plattform, also den Designer zu bedienen, diese App zu bauen und dann die entsprechenden Betriebssysteme und Endgeräte, um die Plattform zu betreiben. Ich würde sagen, ansonsten gibt es A priori keine Voraussetzungen. Es gibt bestimmte Dinge, die die Anwendung sehr stark begünstigen können, aber ich würde sie nicht als K.-o.-Kriterien voraussetzen. Ich überlege gerade, also, ich würde behaupten, technische Expertise ist, ja, man muss eine Maus bedienen können, man muss einen Computer anmachen können, den Monitor natürlich anmachen können und das Internet verbinden. Aber darüberhinausgehend, würde ich genau sagen, das ist ja ein ganz großer Vorteil von solchen Lösungen, wie snapOne, dass man halt nicht darauf angewiesen ist, umfangreiche Voraussetzungen erfüllen zu müssen.
- 32 [0:14:37.8] **I: RS** Also es würde jetzt nicht bedeuten, dass eine bestimmte Unternehmensgröße vorhanden sein muss, bestimmtes Kapital, bestimmte Mitarbeiteranzahl bzw. muss ein bestimmtes Mindset vorherrschen? Das, sagen sie jetzt, ist keine besondere Voraussetzung?
- 33 [0:14:50.8] **B: MH** Also nein, Firmengröße spielt für unser Produkt keine Rolle. Im Gegenteil, das hilft den großen Konzernen hier, ja, die große Rollen spielen, wie auch die ganz Kleinen, mit wenig Mitarbeitern und kleinen Kundenkontakten. Einfach weil die Lösung so frei in ihrer Gestaltung ist, ist man in der Lage, sowohl dort als auch dort zu helfen. Sie haben das Mindset angesprochen und deshalb gehe ich da gerne noch mal drauf ein. Ich würde es als sehr begünstige Komponente beschreiben, die, wenn man die mitbringt, das Richtige in Anführungsstrichen, das Richtige, aber ein Mindset, was die Nutzung von No Code/ Low Code begünstigt, wenn man das mitbringt, ist das durchaus vorteilhaft zur Nutzung. Ich würde es aber nicht als Voraussetzungskriterium definieren.

- 34 [0:15:39.7] **I: RS** Und unter dem Aspekt der Einführung von Low Code-Plattformen, welche Problemstellungen sehen Sie da in Unternehmen oder welche Risiken?
- 35 [0:15:48.9] **B: MH** Die hatte ich schon bei den Nachteilen angesprochen, und die würde ich an der Stelle auch tatsächlich noch mal mit reinbringen. Das eine ist dieser Wildwuchs. Einfach, weil man ja eben nicht an diese klassischen Verfahren gebunden ist. Also egal, ob das jetzt ein Wasserfallmodell ist oder ob das, sag ich jetzt, ein agiler Software-Entwicklungsprozess nach Scrum ist, wo man ja doch eine gute Dokumentation und eine Übersicht trotzdem behält über das, was entwickelt wird und man kann natürlich auch beide Verfahren auf Low Code-Lösungen anwenden. Also man kann sagen, das, was wir jetzt bauen, wollen wir nach Wasserfall bauen oder das, was wir jetzt bauen, wollen wir trotzdem nach Scrum oder nach Kanban oder nach agiler Software-Entwicklungsmethodik bauen. Ich würde aber mal behaupten, das passiert meistens nicht, weil das ja diese Leichtigkeit und Schnelligkeit der Lösung mitbringt, dass man auf diese ganzen Verfahren vielleicht geneigt ist, verzichten zu wollen. Das ist natürlich ein Risiko, weil, dann hat man irgendwann einen Wildwuchs. Man kann vielleicht nicht mehr gut nachvollziehen, wann ist also dieser Change, Change-Management-Prozess, ist vielleicht nicht so gut nachvollziehbar, das muss nicht einhergehen mit einer Low Code-Lösung. Wie gesagt, man kann sich ja trotzdem Governance mäßig quasi vorschreiben, dass bestimmte Prozesse auch entsprechend dokumentiert werden müssen oder so was, aber das sind natürlich, ich sage mal, Schmerzen, die bei solchen Lösungen wahrscheinlich eher auftreten, als wenn das ein strukturierter Software-Entwicklungsprozess ist. Und welche Probleme und Risiken treten auf? Insbesondere würde ich dann halt sagen, das, was ich auch schon vorher berücksichtigt hatte. Die Lösung wird geschafft, die funktioniert schön, aber jetzt geht es darum, diese Insellösung irgendwie wieder in die Hauptarchitektur zu integrieren. Und spätestens dann wird jede Low Code-Lösung, ich kriege das mit als technischer Integrator, als technischer Berater hier. Es ist ja technisch gesehen keine Low Code-Lösung, die darunter liegt, weil, das müssen ja Leute programmieren und wenn es angebunden wird, da fallen ganz normale JSON-Objekte oder XML-Objekte oder so was raus. Oder irgendwelche Bytestreams, und die müssen ja entsprechend verarbeitet werden und deshalb spätestens dann verlässt das ja den gesamten Low Code-Bereich und insbesondere Firmen, die dann Low Code-Lösungen tief integrieren wollen in ihre Unternehmensarchitektur, die sind dann doch angewiesen auf entsprechende IT-Kompetenzen und Kapazitäten, weil die Integration dann doch häufig halt noch eine entsprechende Expertise erfordert.
- 36 [0:18:11.7] **I: RS** Und wenn Sie jetzt in so einem Einführungsprojekt stecken, welche Organisationsform bevorzugen Sie bei sowas? Als Beispiel, würden Sie zum Beispiel die Projektmitarbeiter und den Projektleiter komplett herauslösen bzw. sagen Sie: „Nein, es ist besser, wenn die in ihren Abteilungen verbleiben.“ Wie organisieren Sie Ihr Projekt?
- 37 [0:18:34.6] **B: MH** Das fällt mir schwer, dazu eine pauschale Aussage zu treffen, weil ich finde, dass es weniger eine Fragestellung danach ist, ob es eine Low Code-Lösung oder keine Low Code-Lösung ist, sondern da würde ich mich eher nach anderen Kriterien orientieren. Ja, also ich sage jetzt mal einfach nur, um ein Beispiel zu geben. Der eine Kunde möchte beispielsweise einen Urlaubsantrag, zum Beispiel über eine Low Code-Lösung erhoben wird und die Urlaubsanträge sollen danach in einen Kalender hineinfließen. Und das ist ein klitzekleiner Aufwand, ich sage jetzt mal, das ist innerhalb von einem halben bis einem Tag entwickelt und es ergeben sich auch aus dem Prozess keine weiteren Fragestellungen. Dann ist mir persönlich am liebsten, wenn ich einfach nur den Ansprechpartner vor Ort habe, der das gelöst haben möchte, mit dem kann ich mich kurz irgendwie austauschen. Ganz anders sieht das aber aus. Ein Kunde, der möchte das aus einem SAP-System täglich Daten herausfallen. Die sollen synchronisiert werden in den Datenbestand unserer Lösung. Der Mitarbeiter soll mit unserer Lösung arbeiten, soll dort bestimmte Werte eintragen und am Ende des Tages sollen die Daten dann wieder in das SAP zurückgemeldet werden. Und dass eine war, keine Ahnung, eine kleine Kfz-Werkstatt und das andere ist ein großer DAX Konzern.

Und deshalb, bei dem einen bin ich sehr froh, wenn es schnelle, kurze Dienstwege, wenn was nicht klappt, ruft der mich an, ich kann noch mal reingucken, kann das beheben. Bei dem anderen Kunden, wenn ich da was im Automatismus vielleicht falsch hinterlege, führt das dazu, dass 20 LKWs losrollen, obwohl nur 1 losrollen sollte und mir stellt plötzlich jemand eine Rechnung ins Haus, weil das nicht gut umgesetzt wurde. Dann bin ich natürlich schon gewillt, ein anderes Maß an Dokumentation, Verbindlichkeit, Entscheidungsnachvollziehung, ja, dass jemand eine bestimmte Entscheidung getroffen hat, der entscheidungsbefugt war. Deshalb, das sind Kriterien, die würde ich eher an ganz allgemein, Fragestellung aufmachen, wann ist im Projektmanagement ein PMO erforderlich, wann brauche ich eine Meilensteinplanung. Diese Fragestellung würde ich weniger davon abhängig machen, ob es sich um eine Low Code-Lösung oder eine nicht Low Code-Lösung ist, sondern nach allgemeineren Sachen. Aber ich würde hier auf jeden Fall wieder ins Spiel bringen, Mindset. Das heißt, ich würde behaupten, in einem Umfeld, wo ich Low Code-Lösungen einsetze, könnte es durchaus den Prozess befruchten, das Projekt befruchten, wenn das Mindset insgesamt agil ist und wenn man nicht quasi an alten Strukturen festhält. Aber das ist auch eine schwierige Aussage, die individuell sicherlich nicht immer zutrifft.

- 38 [0:21:24.0] **I: RS** Ok, ich entnehme der Aussage, dass sie sagen: Es ist sehr situationsabhängig in welchem Unternehmen man ist und wie da die Voraussetzungen sind.
- 39 [0:21:34.1] **B: MH** Ja.
- 40 [0:21:36.3] **I: RS** Dann kommen wir zum Kern des Fragebogens, wenn Sie eine Low Code-Plattform einführen, welches Modell bevorzugen Sie? Sie haben ja bereits Wasserfallmodell oder agiles Vorgehen gesagt. Welche Vorgehensweise würde Sie dabei bevorzugen und können Sie gleich noch dazu sagen, welche Vorteile Sie in dieser Vorgehensweise sehen und in Abgrenzung zu der anderen oder den anderen, wenn es ein hybrides Modell zum Beispiel ist?
- 41 [0:22:05.6] **B: MH** Ich bin grundsätzlich ein Freund von agilen Vorgehensweisen. Das ist aber weniger meiner Erfahrung aus der Low Code-Ecke geschuldet, sondern das ist einfach eine sehr persönliche Ansicht, aus meiner Erfahrung als Softwareentwickler, mit den Vorteilen oder auch Nachteilen, die die agile Softwareentwicklung oder das agile Projektmanagement mit sich bringen. Ich glaube, bei der Einführung und der nachhaltigen Nutzung von Low Code-Lösungen tun sich Unternehmen keinen Gefallen, an einem starren Wasserfallmodell festzuhalten. Lassen Sie mich ausführen, warum. Eine wesentliche Stärke von Low Code-Lösungen zeichnet sich durch diese Flexibilität und Schnelligkeit aus. Diese Schnelligkeit ermöglicht den Kunden, zum Beispiel, sehr, sehr schnell auf regulatorische Anforderungen reagieren zu können. Das klassische Wasserfallmodell, das klassische Projektmanagementmodell, hat genau diese Schwäche, dass es nicht flexibel und schnell ist. Sondern im Gegenteil, es baut sich genau an diesen Projektphasen auf, wo die nachgelagerten Phasen immer erst den Abschluss vorgelagerten Phasen erfordern und genau das sind ja die Vorteile der agilen Softwareentwicklung. Dass man quasi durchaus auch diese ganz klassischen Phasen hat. Planung, Durchführung, Testing, Release und die Problembehebung, aber die Zyklen sind einfach viel kleiner. Man hat ja quasi diese klassischen Sprints, im Scrum-Verfahren, im Scrum-Modell, und die ermöglichen einfach ein viel schnelleres reagieren auf einen Wechsel der Anforderungen und ich habe selten eine so hohe Volatilität der Anforderungen im Verlauf der Implementierung erfahren, wie ich das mit Kunden, dieser Lösung hier, erfahre. Einfach, das bringt das Produkt halt mit sich. Das sind Kunden, die wollen ja Flexibilität, die wollen Schnelligkeit und es ist selten so, dass ich am Ende des Projekts die Anforderung noch so habe, wie sie ursprünglich genannt wurden. Und insofern tut sich ein Unternehmen bei der Einführung von No Code/ Low Code- Lösungen glaube ich keinen großen Gefallen, wenn es zu starr an den alten Strukturen, an alten Modellen, festhält und würde immer eher die agilen Modelle, quasi empfehlen, wenn gleich man hier auch noch einmal unterscheiden kann

zwischen der Einführung der Lösung als solche. Das könnte man ja durchaus als ein größeres Projekt im Wasserfallrahmen irgendwie realisieren, wenngleich ich aber insbesondere die Anwendung und dann später die Realisierung von Projekten, von echten Projekten mittels Low Code-Lösungen, dann doch eher agil einführen würde. Also wenn ich dafür die Entscheidungsbefugnis hätte, würde ich mich eher auf ein agiles Modell orientieren.

- 42 [0:25:11.8] **I: RS** Würden Sie aus einem Wasserfallmodell, wenn Sie sich etwas herauspielen könnten, irgendwelche Sachen mit in dem Modell einbauen? Weil Sie gerade gesagt haben, für die Einführungsphase könnte man auch ein Wasserfallmodell nehmen. Gibt es da etwas aus einem Wasserfallmodell, wo Sie sagen, also das hat schon einen Vorteil, das könnte man deswegen berücksichtigen?
- 43 [0:25:30.6] **B: MH** Ich glaube, was beim Wasserfallmodell sicherlich noch mal ein Vorteil sein dürfte gegenüber agilen Methoden ist, bei agilen Methoden verliert man sich dann schon vielleicht zu schnell in den Detailfragestellungen, weil man sich ja mit der Umsetzung der kleinsten, von einem MVP beschäftigt, also so einem Minimal Viable Product, das ist so eine kleine Einheit, die nach jedem Sprint dann irgendwie fertig ist und womit man dann arbeiten kann. Und ich glaube bei so Wasserfallprozessen einfach, natürlich man hat ein bisschen länger Zeit, sich in der Planung Gedanken darüberzumachen, ggf. Seiteneffekte, die durch bestimmte Vorhaben auftreten. Auch so diese übersichtliche Gesamtblick-Brille, diese Berühmte. Die ist sicherlich auch in so einem Wasserfallprojekt eher gegeben, als dass es, ja, man verliert in der Umsetzung bei so kleinen Sprints dann manchmal eher den Überblick, als das bei so Wasserfallverfahren der Fall ist. Und was, ich glaube, auch wieder so ein bisschen, die kleine Kfz-Werkstatt oder der kleine Dienstleister würde sich, glaube ich, immer eher das agile Verfahren wählen, wohingegen der Konzern eher zu einem Prozess tendieren würde, der vielleicht ein bisschen langfristiger, langwieriger ist, aber dafür wiederum besser auditierbar, besser die Compliance-Richtlinien erfüllt. Von daher fällt mir das auch schwer die Entscheidung, ob ich agil oder Wasserfall mache oder klassisch. Weiß ich nicht, ob ich das auch wieder von dem Faktor Low Code oder kein Low Code abhängig machen würde. Ich glaube, die Variablen wären da wieder andere, Unternehmensgröße, Unternehmensorganisationsform etc. pp.
- 44 [0:27:21.5] **I: RS** Aber in jedem Fall wären bei Ihnen, also selbst beim großen Unternehmen, würden Sie selbst, wenn Sie Wasserfall-Anteile mit reinnehmen, primär immer den agilen Anteil bevorzugen?
- 45 [0:27:31.4] **B: MH** Ja, aufgrund der Agilität der Lösung und jeder Low Code- Lösung, also ich kenne durchaus nicht alle, aber ich würde behaupten, die bringen alle eine gewisse Leichtigkeit, Agilität mit sich. Das ist ja der Grund, warum man die wählt und deshalb würde ich, allein aus so einer Perspektive des Mindsets einfach argumentieren wollen. Dass es sich beißen würde, einfach so, der eine ist eher so dieser Langsame, Sichere, ich will erst die Unterschrift haben und der andere ist eher so, ja aber guck mal hier, ich kann doch in 3 Minuten die Lösung zusammenklicken, hier haben wir es doch schon. Ich könnte mir vorstellen, dass sich das auf so einer kulturellen Ebene einfach beißt, auf einer organisatorisch-kulturellen Ebene.
- 46 [0:28:12.8] **I: RS** Bei allen Fragen geht es immer um Ihre persönliche Erfahrung. Ich verlange nicht, dass Sie alles wissen. Bei dem agilen Vorgehen, was Sie ja bevorzugen, sehen Sie da auch Nachteile beim Vorgehen, worauf man achten sollte, wenn man dieses Vorgehen wählt?
- 47 [0:28:33.1] **B: MH** Ja, wir hatten gerade schon diese durchaus auftretende mangelnde Übersicht angesprochen. Und es gibt ein Sprichwort: Ein guter Plan ermöglicht überhaupt erst eine Improvisation. Weil, ich habe immer wieder quasi meinen Fahrplan, auf den ich, wenn die Improvisation fehlschlägt, quasi mich verlassen kann. Und das ist jetzt sicher wenig

wissenschaftlich, was ich sage, aber wenn man es mal auf ein Kontinuum legt und sagt, das ist eine Achse, wo die eine Seite ganz wenig Agilität ist und die andere ganz viel Agilität, dann ist es ja so... Jetzt habe ich ein bisschen den Faden verloren. Können Sie die Frage noch mal stellen?

- 48 [0:29:31.8] **I: RS** Sie bevorzugen das agile Vorgehen, welche Nachteile sehen Sie, worauf man achten sollte, wenn man dieses Vorgehen wählt?
- 49 [0:29:41.6] **B: MH** Übersicht behalten und sich trotzdem gewisse Konventionen schaffen, einen gewissen Fahrplan schaffen, einen gewissen Rahmen schaffen, ein gewisses Framework schaffen, was es ermöglicht, das klingt widersprüchlich, aber die Agilität ein bisschen einzugrenzen. Das klingt widersprüchlich, weil, ich will ja eigentlich genau durch diese Agilität das Team sich selbst befähigen und sprudeln lassen und auch Kreativität enablen. Nichtsdestotrotz glaube ich, bei Low Code Lösungen und auch bei agilen Projekten ist es hilfreich, wenn man sich initial zu Beginn des Projekts oder eben auf unternehmerischer Ebene, als Konvention für alle Projekte der Zukunft, sich gemeinsam bestimmte Konventionen, Restriktionen entwickelt oder Best Practices sich vor Augen führt, wie man da diesen Wildwuchs, den Verlust der Übersicht einfach kompensieren kann.
- 50 [0:30:53.2] **I: RS** Ich muss jetzt in dem Zusammenhang noch mal zurückkommen. Das, was Sie beschrieben haben, wäre nicht dann eine Kombination mit einem übergeordneten Wasserfallmodell eine sinnvolle Einschränkung?
- 51 [0:31:07.2] **B: MH** Wie würden Sie das aufbauen. Können Sie das mal skizzieren?
- 52 [0:31:11.5] **I: RS** Bloß als Beispiel, ein Wasserfallmodell zeichnet sich ja dadurch aus, dass es, Planungssicherheit, dass es ein Framework vorgibt. Es wird viel im Voraus schon gestaltet und in den Phasen wird dann entwickelt, aber viel Planung findet ja im Voraus statt, viel der Dokumentation. Das wäre dann quasi das umliegende Framework und in diesen Phasen wäre es ja durchaus möglich, dann eine agile Entwicklungsweise anzunehmen. Also, wenn man das einführt, immer wieder in Sprints, also in agilen Methodiken, immer wieder zu reflektieren und es darüber einzuführen. Aber innerhalb der Realisierungsphase, nachdem man vorher bestimmte Planungen schon als Vorstufe hatte. Es ist bloß eine Frage.
- 53 [0:31:52.7] **B: MH** Es fällt mir ein bisschen schwer das zu beurteilen. Ich kann mir durchaus vorstellen, also das entspricht ja auch so ein bisschen dem, was ich gesagt hatte. Also quasi so ein bisschen dieser Versuch, so ein bisschen die Agilität einzufangen. Irgendwie durch ein Rahmenwerk irgendwie dann doch, zwar ein bisschen weniger agil zu machen, aber da auch wieder ein bisschen besser steuerbar oder manageable quasi zu machen. Mir fällt es aufgrund mangelnder Erfahrung so ein bisschen schwer, das irgendwie bewerten zu können. Kann mir durchaus vorstellen, dass das Vorteile mit sich bringt, kann mir aber auch durchaus vorstellen, dass man dann mit so einem Chaos verwirrt. Zwischen: Was sind wir denn jetzt eigentlich für eine Form? Ist das jetzt eher Starr und wollen wir erst mal abschließen zu planen? Oh, jetzt hat aber jemand einen coolen agilen Gedanken, darf er den jetzt in diesem kleinen Cycle zu Ende bringen? Oder muss er jetzt. Das heißt, das fällt mir jetzt schwer zu beurteilen.
- 54 [0:32:54.8] **I: RS** Ok, es hatte sich bloß ergeben, weil ihr beschriebener Nachteil, quasi genau die Vorteile des Wasserfallmodells beschrieben hat. Bei der agilen Methodik, Sie haben ja schon Scrum erwähnt, wäre das auch Ihre bevorzugte Methode oder würden Sie auch weitere einbinden? Es gibt ja auch noch Kanban und ähnliche Methodiken.
- 55 [0:33:12.1] **B: MH** Ich habe eigentlich gute Erfahrung mit Scrum gemacht. Insbesondere auch weil ein Leitsatz von Scrum selber ja ist „Halte dich nicht zu sehr an die Regeln, sondern lege es dir im Zweifelsfall auch so ein bisschen selber aus.“ Insofern sind ja alle agilen Frameworks nicht zu verstehen, also nach meinem Kenntnisstand, nicht zu verstehen als ein

	festes hartes Korsett, in dem gearbeitet wird, sondern eine Empfehlung, wie man bestehende nicht agile Methoden verbessern kann oder wie man die Schwächen quasi ausgleichen kann. Insofern, ich habe in verschiedenen Unternehmen mit Scrum gearbeitet, die eine unterschiedliche Ausprägung hatten, also die unterschiedlichen Parameter mitgebracht haben, was Sprintlänge anging, was Artefakte in dem Prozess anging. Also die einen haben eine Sprint-Retrospektive gemacht, die anderen haben es nicht gemacht. Und insofern sehe ich da jetzt keine Schwächen, wo ich sagen würde: OH, da ist Scrum überhaupt nicht geeignet, da müsste man eher Kanban nehmen. Dafür kenne ich die Modelle in der Tiefe auch zu wenig. Fällt mir jetzt schwer, da jetzt irgendwie entsprechende Vergleiche zwischen Kanban und Scrum zu ziehen. Ich glaube, Kanban ist noch ein bisschen lockerer und basiert eher auf so einem Pull-Prinzip, also des Aufgabenpullings, als dass es zugewiesen wird. Ich glaube, es gibt auch kein Planing-Programm bei Kanban. Weiß ich jetzt nicht genau.
56	[0:34:47.8] <b>I: RS</b> Es gibt Methodiken, wo dann die Aufgaben quasi im Team aus einem Backlog gezogen und dann zieht sich quasi jede Arbeitsstation, wenn das von der anderen freigegeben worden ist, ziehen die sich, also dieses Pull-Prinzip, ziehen die sich ihre Arbeit ran und unterscheiden halt in Bearbeitung und fertig. Also so funktioniert Kanban.
57	[0:35:08.3] <b>B: MH</b> Also ich kenne die Methoden nicht so gut, aber wie gesagt, weil ich jetzt behaupten würde: Scrum ist ja kein hartes Regelwerk, was unbedingt nur so angewendet werden darf, wie es im Lehrbuch steht, sondern das Modell sagt von sich selber: Sieh mich als ein gedankliches Mindset, eher als ein Framework und versuch damit deine Projekte zu meistern. Insofern würde ich sagen: Passt schon.
58	[0:35:34.4] <b>I: RS</b> Scrum zeichnet sich ja, wie Sie selber gesagt haben, dadurch aus, dass es sehr gut mit Änderungen umgeht. Wie würden Sie mit Änderungswünschen an das Projekt oder aus dem Projekt selber umgehen?
59	[0:35:47.1] <b>B: MH</b> In einem Low Code-Projekt?
60	[0:35:50.0] <b>I: RS</b> Ja, also immer unter dem Aspekt: Einführung einer Low Code-Plattform. Wie würden Sie da mit Änderungswünschen aus dem Projekt selbst oder mit Änderungswünschen von außen an das Projekt umgehen?
61	[0:36:01.7] <b>B: MH</b> Wir arbeiten auch agil. Die Änderungswünsche werden in den entsprechenden Backlog überführt bzw. mit den bestehenden Backlog Items abgeglichen. Also entweder es ist eine neue Anforderung oder es ist eine geänderte Anforderung oder auch der Wegfall der Anforderung. Das wird entsprechend im Backlog vermerkt und dann findet das wieder Einsortierung in den regulären agilen Fluss, in das agile Projektmanagement. Also Bewertung der Items ziehen in den aktuellen Sprint und Umsetzen.
62	[0:36:34.9] <b>I: RS</b> Und welche wichtigen Rollen würden Sie jetzt dafür benennen? Also es gibt ja, Sie sagen ja, Scrum ist die Methode, die Sie bevorzugen. Welche wichtigen Rollen sehen Sie da? Es gibt ja bestimmte klassische Rollen wie Product Owner und Scrum Master und Development Team. Sind alle Rollen dann bei Ihnen wichtig oder sagen Sie, bestimmte Rollen sind wichtiger als andere?
63	[0:37:02.9] <b>B: MH</b> Ja, also eine Anmerkung und eine Frage. Die erste Frage ist: Gibt es denn in einem Low Code Projekt, wenn es nicht um Integration geht, gibt es denn den Developer dann überhaupt?
64	[0:37:13.0] <b>I: RS</b> Es würde mir aber um Integration gehen.
65	[0:37:15.1] <b>B: MH</b> Also der Citizen Developer, dann ja. Wir haben ja gar nicht mehr den klassischen Developer. Also ich würde es auch wieder, das ist der Hinweis, die Anmerkung, ich würde es auch wieder abhängig davon machen, wie ist das Projekt setting denn drum herum? Ist das jetzt wieder der kleine Dienstleister, der keine Ahnung, einmal am Tag Essen

ausfährt oder so was. Bei dem würde es mir für das Projekt setting grundsätzlich reichen, sofern er mich als technischen Berater benötigt, dass er und ich oder sie und ich in einem Projektteam sitzen und dann sehe ich nicht die Erfordernis für weitere Rollen, weil es geht dann darum, dass eine PDF irgendwo in einem lokalen Archiv abgelegt wird, wo habe ich heute Essen ausgeliefert oder so was. Jetzt gucken wir uns aber die andere Sache wieder an. Keine Ahnung, ein großer DAX-Logistikdienstleister oder so was, möchte uns einsetzen, dann bin ich schon darauf angewiesen, dass dort mir entsprechenden IT-Leute an der Seite sitzen, die den Counterpart, also alle Gegenseiten beraten können, also ich kann ja immer nur meine Seite beraten, ich kann ja immer nur meine Schnittstelle beraten. In die Kunden-seite kann ich mich quasi nur reinarbeiten. Also am besten ist da jemand vor Ort, der mich beraten kann. Also quasi der technische Counterpart. Ich brauche durchaus bei größeren Projekten auch jemand, der einen Blick auf das Projekt hat. Also ob das jetzt ein fachlicher Projektleiter ist oder ein PMO, im Sinne eines eher administrativen Officers. Der fachliche Projektleiter muss sowieso irgendwie mit drinne sein. Entweder das implizit der Kunde und ich, irgendwie so, je nachdem ob das der Kunde und ich ist oder ob das eine dedizierte Instanz ist, die irgendwie benannt wird und so eine administrative PMO-Stelle braucht man sicherlich auch, wenn es ein umfangreiches Projekt ist. Wenn das so ein 2-Jahres- Läufer ist, mit 35 beteiligten Abteilungen und paar 100, da brauche ich schon eher einen PMO als wenn es das kleine 2-Personen-Projekt ist. So ein Project Owner, Product Owner im Sinne eines, nicht der fachliche Projektleiter, sondern im Sinne eines Auftraggebers, eines Sponsors. Den habe ich natürlich auch. Also im kleinen Projekt Setting habe ich das auch alles, aber es ist meist vereint in Personal-Union, alle Rollen sind irgendwie eins. Da macht einer PMO und Projektleitung und ist gleichzeitig der Sponsor, also Auftraggeber. Und in großem Unternehmen ist das dann natürlich, kommt das automatisch wieder mit. Ich würde behaupten, wirklich essenziell ist der Citizen und, je nachdem ob ich technische Umsysteme integrieren muss, auch die IT Counterpartseite. Ansonsten würde ich aus so einer technischen Perspektive, ich habe auch viele Projekte, quasi fachlich betreut, aber ich spreche jetzt mal aus meiner Brille als Informatiker, als Techniker. Da ist jede weitere Person im Projekt oft nur hinderlich, weil, wenn ich so diese technische Brille aufhabe, dann geht es mir nur um technische Wahrheiten. Und die lassen sich in der Regel immer sehr einfach bewerten, weil, in der Informatik ist es einfach. Es ist True oder False, es sei denn wir gehen in die Quanten-Computer-Theorie rein, aber bleiben wir mal bei den normalen Bits und Bytes, da ist es True oder False. Ich komme nicht in solche politischen Grabenkämpfe und als Techniker ärgert mich das dann natürlich, wenn da so ein Projektleiter zwischendurch kommt und sagt: Wir hatten hier einen Shareholder oder Stakeholder oder wir hatten noch mal den Manager oder da irgendjemand, der hat gesagt, hier will er auch noch mal von links seinen Senf dazu geben. Aus der technischen Perspektive ist das natürlich hinderlich und möchte ich das eher nicht. Aber das sind dann Compliance. Die Herausforderung oder die Erfordernis weiterer Personen oder Rollen ergeben sich dann weniger aus technischer Sicht, also Umsetzung des Kundenwunsches, den er mit Low Code ja oft selber erreichen kann. Insofern würde ich sagen, die einzig wichtige Projekttrolle ist der Citizen selber, also der Anwender selber, der sich seine Lösung selber bastelt.

- 66 [0:41:33.7] **I: RS** Also, der sollte dann auch von Anfang an mit im Einführungsprojekt vorhanden sein?
- 67 [0:41:37.8] **B: MH** Also, um die Lösung einzuführen?
- 68 [0:41:41.0] **I: RS** Ja.
- 69 [0:41:42.3] **B: MH** Ja, muss nicht zwangsläufig, da müssen wir das jetzt auch wieder trennen. Es gibt das eine Projekt Firma XY, die Wasserwunder GmbH, möchte jetzt eine Low Code-Lösung einführen, bei sich im Unternehmen. Das ist ja ein Projekt und dann haben wir das Projekt mittels der Low Code-Lösung XY soll der Prozess ABC umgesetzt werden. Das sind



ja zwei komplett unterschiedliche Prozesse, die andersartiger nicht sein könnten. Insofern, da muss man auch noch mal unterscheiden. Ich würde aber sagen, für die Umsetzung, also nicht Einführung, sondern für die Umsetzung der Anforderungen braucht es nur den Citizen Developer. Für die Einführung der Lösung im Unternehmen braucht es ganz dringend nicht nur den Citizen Developer, sondern hier ist es ganz wichtig, dass wir diese Restriktion, Konventionen im Blick behalten, das heißt, dass diese Governancebrille aufgenommen wird, damit eben nicht diese Nachteile und Schwächen, die wir vorhin schon mal diskutiert haben, eintreten. Insofern halte ich es für essenziell, ich würde aber auch gleichzeitig behaupten, dass der Citizen Developer diese Notwendigkeit nicht sieht und es im Gegenteil als einen Knüppel zwischen den Beinen empfindet.

70 [0:43:00.1] **I: RS** Und wir sind jetzt beim Einführungsprojekt. Sie haben ja gesagt, der Citizen Developer kann gut sein, es ist kein Muss. Wen sehen Sie denn dann bei dem Einführungsprojekt? Weil, das ist für die Arbeit wirklich der Fokus, auf das Einführungsprojekt, das Umsetzungsprojekt ist ein bisschen hintenangestellt.

71 [0:43:25.4] **B: MH** Also ich würde das mal von oben nach unten durchgehen. Geschäftsführung: Je nachdem, ob es ein Vorhaben ist, dass ich No Code/ Low Code ganz neu einführe, um ein Digitalisierungsvorhaben umzusetzen, dann ist natürlich die Geschäftsführung, einfach als wesentlicher Projekttreiber, durchaus gefragt. Wenn das jetzt aber heißt, ich möchte von Low Code-Lösung A auf Low Code-Lösung B und insgesamt sind wir schon sehr agil und Low Codig unterwegs, dann ist dieses Erfordernis natürlich nicht so sehr gegeben. Zweite Stelle: IT, je nachdem wie tief ich die Low Code-Lösung in meine IT integrieren möchte. Also möchte ich, wie ich vorhin gesagt habe, ein SAP-System, Daten sollen rausfallen, am Ende wieder zurückfallen. Ist selbstverständlich, die IT als wichtigen Projektpartner mit ins Boot zu holen, sonst hängen wir die ab. Nicht technisch, sondern kulturell/ organisatorisch, die haben dann einfach keinen Bock mehr. Ich würde mal behaupten die verdrehen sowieso häufig die Augen bei Low Code Lösungen, weil, am Ende ist es viel Wildwuchs, der dann irgendwie in der IT zusammengefeigt werden muss. Ansonsten, auch wieder abhängig von der Projektgröße und der Unternehmenskultur, ist es sicherlich hilfreich bei so einem Einführungsprojekt, diesen kulturellen Aspekt auch irgendwie zu begleiten. Ja, also wir gehen ja im Grunde ganz krass, findet ein Wandel statt von: Ich habe als Fachabteilung eine Bittstellung. Die muss ich erst mal irgendwie mühselig an die IT herantragen. Vielleicht muss ich die fachlich erst mal von der Geschäftsführung absegnen lassen, bevor ich damit auf die IT zugehen kann. Versus, ich habe eine Idee, ich log mich eben ins Web ein. Bau mir so ein Ding. 3 Minuten später habe ich das Ding bei mir auf dem Handy und kann damit operativ loslegen. Das ist ja ein unglaublicher kultureller Shift, der hier stattfindet. Ich würde mal behaupten, das ist kein Erfordernis, dass man das begleitet, kulturell. Ich halte es aber für durchaus sinnvoll, dass man das irgendwie so durch einen Citizen Development- Beauftragten, einen Mindset-Beauftragten, wie auch immer, das ist so ein bisschen Bullshitbingo, Buzzwordbingo. Aber ich halte es durchaus für geeignet, dass man die Leute mitnimmt und aber auch wieder, das ist sehr abhängig von der individuellen Situation im Unternehmen. Wenn ich jetzt ein sehr junges dynamisches Unternehmen, eine studentische Beratung oder so was. Der muss ich nicht viel erklären von Low Code. Wohingegen das alte Maschinenbauunternehmen, wo quasi viele, viele Mitarbeiter kurz vor der Verrentung stehen. Die werden sicherlich ein anderen Mindset haben oder diesen immensen Change, der dort stattfindet, dieses Paradigma was sich so ganz krass ändert in der Herangehensweise. Das heißt, so einem Unternehmen würde ich dann doch eher noch mal, wo ich sage, dann geht doch noch mal in kulturelle Workshops oder so was rein, um die mitzunehmen, als wie das in einem anderen Unternehmen der Fall ist. Geschäftsführung, IT, Kultur beobachten und wenn ich sage, die Kultur muss beobachtet werden, dann werden sich natürlich auch die Organisationsformen ändern. Wie gesagt, der Prozess von der Anforderung zum fertigen Produkt ist ja plötzlich ein ganz anderer. Das heißt, es ist durchaus auch sinnvoll, dass irgendwie eine

organisatorische Instanz mit drin ist, die diese Prozesse betrachtet und begleitet. Also einerseits, wie ändert sich denn der Prozess von der Anforderung zum Produkt, aber auch, wie finden denn die Schnittstellen statt von eurem Citizen Development gestalteten Produkt bis hin zur Integration in die Unternehmenssoftware. In die bestehende Unternehmenssoftware. Das heißt, dass man hier so eine organisatorische Brille mit aufhat. Entweder in Form vom Geschäftsführer, den ich schon genannt hatte, oder wenn es ein großes Unternehmen ist, was solche Instanzen dediziert bereithält, dann halt auch, dass man die ins Boot holt. Und sicherlich einen Digitalisierungsbeauftragten, wenn das nicht Chefaufgabe ist, klar den nenne ich jetzt mal nicht mit. Den nenne ich jetzt mal nicht mit oder ich nenne ihn an der Stelle mal ganz kurz mit. Und ansonsten vor allem die Leute, die es umsetzen sollen. Die Citizen, die Developer.

- 72 [0:47:39.9] **I: RS** Da spielt die Befähigung von Mitarbeitern gleich rein. Das ist eine gute Überleitung zu meiner nächsten Frage auch. Sie haben das jetzt eingeführt. Wie gestalten Sie die Befähigung der Mitarbeiter?
- 73 [0:47:50.3] **B: MH** Können Sie mal 1-2 Beispiele einfach aufmachen, damit ich weiß, in welche Richtung die Frage, die Sie mir stellen, geht?
- 74 [0:47:55.0] **I: RS** Sie haben jetzt die Plattform eingeführt. Jetzt haben Sie Ja gesagt, am wichtigsten sind die im Unternehmen, die es dann anwenden müssen. Die kommen ja nun nicht in das Büro rein und wissen sofort: Ach so funktioniert die Plattform, die müssen ja erst mal das Wissen erhalten und die müssen ja auch die Möglichkeit haben, sich Wissen anzueignen und auch vielleicht, also gegenseitig, sich zu helfen, weil irgendwann sind Sie ja nicht mehr Unternehmen und da müssen ja trotzdem die noch irgendwie Fragen stellen können. Wie lösen Sie das? Also wie machen Sie den Wissenstransfer der Citizen Developer von Ihnen zu den Developern und wie organisieren Sie das, wenn Sie weg sind, dass die trotzdem noch bei Problemen Lösungen finden könne, wo sie selber erst mal nicht weiterwissen.
- 75 [0:48:44.5] **B: MH** Weiß nicht, ob ich da jetzt quasi als Anbieter der richtige Ansprechpartner für bin.
- 76 [0:48:54.7] **I: RS** Sie müssen ja auch Ihr Wissen an das Unternehmen erst mal weitergeben. Sie müssen ja auch initial erst mal den Mitarbeitern Wissen geben.
- 77 [0:49:01.5] **B: MH** Ach so ok, also Sie meinen jetzt quasi auch aus Anbieterperspektive?
- 78 [0:49:03.8] **I: RS** Ja, aus Ihrem Wissen heraus, aus Ihrer Erfahrung heraus. Das ist mir wichtig bei dem Experteninterview, das sind Ihre Erfahrungswerte.
- 79 [0:49:12.6] **B: MH** Also, wir als Anbieter haben zwei wesentliche Initiativen, mit denen wir genau das erreichen wollen. Wir haben einerseits ein Format, das nennt sich smapCademy. Das ist die Möglichkeit, sehr gezielt mit Hilfe von geschulten Trainern, das Wissen zur Anwendung gezielt erlernen. Mit Kursen, mit Prüfungen etc. pp. Das heißt, das ist dieser erste initiale Wissenstransfer von uns als Anbieter, wie kann ich dieses Produkt überhaupt benutzen auf den Anwender. Und das Zweite, wie können sich die Anwender untereinander helfen. Da gibt's ja auch diese verschiedenen Community of Interest, Community of Experts oder so was. Ich habe sie gerade nicht mehr alle parat, aber genau in diesen Bereich gehen wir auch rein, mit einer zweiten Initiative, und zwar bauen wir gerade auf eine Community für sma-pOne, wo sich quasi die Anwender, natürlich insbesondere intensive Anwender, also vor allem Leute, die das sehr, sehr intensiv nutzen und die das sehr leidenschaftlich nutzen, versuchen wir die untereinander zu vernetzen. Und durchaus auch für einen Wissenstransfer untereinander. Also gar nicht mal nur für die Anwender in einem Unternehmen, in einer Organisationseinheit, sondern tatsächlich über Grenzen hinaus. Das sich zum Beispiel Unternehmen der gleichen Branche untereinander vernetzen können. Hey, ihr habt eine tolle Lösung gebaut, könnt ihr uns mal einen Tipp geben, wie ihr das gelöst habt? Und das sind genau

diese 2 Stränge, die ich gerade aufgezeigt habe, wo wir genau dafür jeweils Formate anbieten, um genau das zu machen. Wissenstransfer über die smapCademy, gezielt Wissen vermitteln und das Zweite ist, eine Plattform, ein Marktplatz zum Austausch und zur Selbsthilfe, zur gegenseitigen Hilfe zu schaffen.

- 80 [0:51:23.4] **I: RS** Nur kurze Frage. Wie sehen diese Kurse der smapCademy aus? Ist das ganz klassisch, wie man sich das auch an einer Universität zum Beispiel vorstellt, da steht einer vorn und trainiert die oder sind das Videokurse oder wie ist das gestaltet?
- 81 [0:51:34.5] **B: MH** Wir haben aktuell noch ein eher klassisches Modell, das befindet sich gerade auch selber in der Transformation. Wir haben momentan noch den Fall, dass wir sehr klassische Präsenzvorlesungen virtuell halten, an denen die Leute teilnehmen, die eher kleine Übungsaspekte live beinhalten. Wo dann die Übungen jeder selber zu Hause vornehmen kann und wir wollen das gerade, das fußt so ein bisschen auf meinen Erfahrungen als wissenschaftlicher Mitarbeiter, wo ich mich mit E-Learning und so was beschäftigt habe. Das bauen wir gerade um in ein Flip-Classroommodell, wo genau das umgedreht wird. Wo die Vorlesungen Asynchron als Video-on-Demand-Vorlesungen bereitgestellt werden. Das heißt das theoretische Wissen soll sich jeder selber aneignen in dem eigens gewählten Tempo, zu der eigens gewählten Uhrzeit, an dem selbst gewählten Tag. Und dann gemeinsame Treffpunkte, also Übungseinheiten, wo dann vertiefend im Gemeinsamen darauf eingegangen wird, wie die Lösungen entsprechend gestaltet werden können.
- 82 [0:52:41.6] **I: RS** Und in dem Zusammenhang: Befähigung bedeutet ja im Endeffekt auch Skalierung in einem Unternehmen. Es ist ja nie so, dass man es sofort in einem ganzen Unternehmen einführt. Wie gestalten Sie die Skalierung dann in das Unternehmen? Wenn Sie das jetzt in einer Abteilung, angenommen, eingeführt haben. Wie funktioniert die Skalierung?
- 83 [0:52:58.9] **B: MH** Also, Sie haben ja erst mal grundsätzlich gesagt, wir versuchen gerne, mit den Kunden zusammen die Eignung des Produkts zu testen oder zu validieren, indem wir es in einem fokussierten Bereich, in einem bestimmten Department, in einer bestimmten Abteilung zunächst erst mal einführen und dort vereinzelte Prozesse testen und wenn der Kunde damit zufrieden ist. Das ist ja so eine Kundenentscheidung. Wir möchten natürlich gerne, dass der Kunde das hoch skaliert und in allen Bereichen anwendet als Anbieter und der Kunde überträgt dann im Grund sein Wissen, also das ist in der Vergangenheit immer so gewesen, dass man sich dann weitere Abteilungen sucht, wo es gut passen würde und dann entsprechend den Transfer auf diese Abteilungen vornimmt. Wenn das nicht sowieso als eine große Digitalisierungschefaufgabe kam, dass es ganz weit eingeführt werden soll, ist das so der Fall, dass man zunächst einmal beginnt, Prozesse und die Eignung zu testen im kleinen Rahmen und das dann halt ausrollt auf weitere Prozesse, auch mit unterschiedlichen Verantwortungsrollen. Manchmal ist das der Fall, dass man sagt, einer betreut alle Prozesse für verschiedene Abteilungen oder man sagt, man gibt die Verantwortung in die einzelnen Abteilungen. Da gibt's verschiedene Varianten und Lösungen, was durchaus auch immer ein Ansatz bei uns ist, um diesen Wissenstransfer, also gar nicht mal die Skalierung der Lösung oder die Diffusion in die verschiedenen Bereiche, sondern mehr so der Wissenstransfer, dass wir so ein Train-The-Trainer-Konzept versuchen herbeizuführen. Das heißt, dass wir sagen, wir versuchen nicht nur die Leute zu befähigen, es selber umzusetzen, sondern darüber hinaus die auch noch zu befähigen, andere Leute zu befähigen. Einfach um da quasi die Abhängigkeit zu verringern, in dem wir quasi sagen, Train-The-Trainer und der soll dann bitte auch die Leute befähigen. Dass wir so einen Multiplikations- und Skalierungseffekt dadurch erzielen.
- 84 [0:55:02.0] **I: RS** Wir nähern uns jetzt schon langsam dem Ende unseres Interviews. Mich würde noch interessieren, wie Sie die Qualität von einem Projekt sichern bzw. wie Sie den Erfolg des Projekts messen an sich?

- 85 [0:55:13.6] **B: MH** Das ist technisch, also ich sage das jetzt mal nur aus einer rein technischen Perspektive. Technisch messe ich den Erfolg, also ich spreche jetzt mal nur von Integrationsprojekten, ich rede jetzt nicht von reinen, obwohl man kann das jetzt auch mal ohne Integration, solche Projekte betreue ich hier im Unternehmen halt weniger. Ich komme immer dann ins Spiel, wenn es darum geht, irgendwie ein Umsystem anzuschließen. Und ich bewerte das als einen Erfolg, wenn der Kunde sagt: Ja, die Daten aus dem Fremdsystem kommen genauso, wie sie sollen, bei mir in meiner Low Code-Lösung an. Ich kann die Lösung effektiv und effizient bedienen. Also insbesondere, dass sie mir hilft, meinen Arbeitsprozess einfach zu bewerkstelligen und dass die Daten dann am Ende wieder sauber ins Systems kommen, und den Erfolg messe ich dann, wenn der Datenfluss sauber gewährleistet ist. Und das ist quasi mein technischer Erfolg dann und der Erfolg für den Kunden, daran bemisst sich dann auch, ob er unser Kunde bleibt oder nicht, ist natürlich die Ersparnis beim Kunden. Also da muss irgendein Effekt irgendwie sichtbar sein. Das muss schneller gehen. Das muss leichter gehen. Medienbrüche sind plötzlich weg. Wir haben ja ganz viele Kunden, die nach wie vor mit Papier und Klemmbrett arbeiten und Stift. Und die sagen: Herr Hesse, der Nutzer tippt das hier ein, dann fotografiert er das hier ab, dann landet das per Mail bei der Frau Schulze. Die Frau Schulze nimmt sich das, tippt das dann in die Excel-Tabelle, die Excel-Tabelle wird irgendwo abgelegt. Sie glauben gar nicht, wie oft das passiert, dass die Frau Schulze einen Zahlendreher drin hat und plötzlich gehen unsere Prozesse nicht mehr. Ja, das heißt, der Erfolg bemisst sich ja durchaus daran, wie die Effekte beim Kunden wahrnehmbar sind. Wow! Ich habe früher was an die IT gegeben. Nach 4 Wochen hatte ich die Antwort: Können wir umsetzen. Ist Ende Herbst da. Jetzt habe ich eine Idee. Ich klick mich rein in die Plattform und 15 Minuten später habe ich eine Applikation, mit der kann ich meine Daten hier draußen im Feld erfassen und wenn ich es abgeklickt habe, landet die Excel-Tabelle bei der Frau Schulze und die muss nichts mehr umändern. Also das ist dann die wesentliche Erfolgsmessung.
- 86 [0:57:23.9] **I: RS** Dann würde mich noch interessieren. Haben Sie Anmerkungen zu dem Interview oder Best Practices, die Sie gerne teilen möchten? Also Sachen, wo Sie sagen: Die sind jetzt nicht im Interview vorgekommen. Finde ich aber wichtig, die möchte aber gerne noch unterbringen.
- 87 [0:57:37.8] **B: MH** Nö, fällt mir spontan nichts ein. Also eigentlich ich habe da alle wesentlichen Punkte, war ich drauf eingegangen.
- 88 [0:57:43.2] **I: RS** Dann sind wir jetzt am Ende. Ich bedanke mich bei Ihnen und ich würde jetzt die Aufnahmen beenden.
- 89 [0:57:52.9] **ENDE**

## Einwilligungserklärung Interview

Ich erkläre hiermit mein Einverständnis zur Nutzung der personenbezogenen Daten, die im Rahmen des folgenden Gesprächs erhoben wurden:

Datum des Interviews	30.06.2022
Namen der zu interviewenden Person	Moritz Hesse
Name der interviewenden Person	Rocco Schenk
Kurzbeschreibung des Forschungsprojekts	Einführung von No Code/ Low Code aus Sicht des Projektmanagements
Durchführende Hochschule	Hochschule Mittweida – University of Applied Science

Die Daten werden im Rahmen eines mündlichen Gesprächs erhoben, das mit einem Aufnahmegerät und Videoaufnahmegerät aufgezeichnet wurde. Zum Zwecke der Datenanalyse werden die mündlich erhobenen Daten verschriftlicht (Transkription), wobei die Daten nicht anonymisiert werden.

Vorhandene Kontaktdaten, der interviewten Person, werden aus Dokumentationsgründen in einem separaten Schriftstück lediglich den Gutachter\*innen der wissenschaftlichen Ausarbeitung zur Verfügung gestellt. Nach dem Abschluss des Projekts werden diese Daten gelöscht.

Der Speicherung der personenbezogenen Daten zu Dokumentationszwecken kann durch die interviewte Person jederzeit widersprochen werden. Die Teilnahme an dem Gespräch erfolgt freiwillig. Das Gespräch kann zu jedem Zeitpunkt abgebrochen werden. Das Einverständnis zur Aufzeichnung und Weiterverwendung der Daten kann jederzeit widerrufen werden.

\_\_\_\_\_  
Vorname und Name in Druckbuchstaben

  
\_\_\_\_\_  
Unterschrift

\_\_\_\_\_  
Datum, Ort

## Anlagen 8, Interview EB07

RS: Rocco Schenk

TJ: Tino Janke

1	[0:00:00.0] <b>Start (TJ - TinoJanke/ RS - Rocco Schenk)</b>
2	[0:00:05.9] <b>I: RS</b> Hallo Tino. Erst mal, bist du mit der Aufzeichnung einverstanden?
3	[0:00:11.7] <b>B: TJ</b> Hallo Rocco. Ja, ich bin einverstanden damit.
4	[0:00:14.4] <b>I: RS</b> Deine Daten werden vertraulich behandelt und zum Zwecke der Masterarbeit verwendet und danach wieder gelöscht.
5	[0:00:22.9] <b>B: TJ</b> Okay.
6	[0:00:24.8] <b>I: RS</b> Wir haben uns ja für das Interview heute verabredet und ich wollte mal kurz, dass du dich schnell vorstellst, was du so gemacht hast bis jetzt. Wie deine aktuelle Position ist, um das einzuordnen, wie du da im wissenschaftlichen Kontext, also wie da deine Expertenmeinung zu werten ist.
7	[0:00:48.0] <b>B: TJ</b> Gerne, mein Name ist Tino Janke, ich habe ganz ursprünglich mal Jura studiert. Im Bachelor-Master-Programm und bin aber inzwischen als IT-Berater bei einer Telekomtochter tätig und da speziell für Microsofts Low Code-Plattformen, die Power-Plattform, zuständig. Genau. In meiner Rolle erarbeite ich zum einen für unsere Kunden selber Lösungen, also stelle Workflows, baue Apps, zum anderen ist aber auch ein Teil der Arbeit so, dass ich Anforderungen der Kunden steuere, Projekte steuere und je nach Schwierigkeitsgrad der Anforderungen, wie gesagt, ich hab ja keinen Entwickler-Hintergrund, gegebenenfalls auch die Erarbeitung von Lösungen zusammen im Team mit begleite, dass also einfach bei schwierigen Anforderungen, komplexen Anforderungen einfach noch ein, zwei Entwickler bei uns aus dem Team mit dazu kommen und wir dann zusammen im Team eine Lösung für den Kunden bauen. Genau. Das vielleicht so zu meiner aktuellen Rolle.
8	[0:02:15.2] <b>I: RS</b> In meiner Arbeit geht es ja um die No Code- und Low Code-Plattformen. Könntest du mal kurz sagen, wie du No Code und Low Code für dich definieren würdest?
9	[0:02:27.2] <b>B: TJ</b> Low Code bzw. No Code ist aus meiner Sicht ein Ansatz zur Entwicklung von Software. Wobei die Besonderheit bei Low Code und No Code-Daten liegt, dass es eine vergleichsweise selbsterklärende Nutzeroberfläche gibt und man zum Beispiel, vergleichbar mit PowerPoint, mit einer Drag-and-drop Oberfläche arbeiten kann, so dass auch Nutzer diese Plattformen diese Low Code/ No Code-Plattformen nutzen können, obwohl sie eben keinen Entwickler-Hintergrund haben und keine Programmiersprache beherrschen.
10	[0:03:21.3] <b>I: RS</b> Okay, hast du schon mal was von Citizen Development gehört?
11	[0:03:27.4] <b>B: TJ</b> Ja, das geht ja stark mit dem Thema Low Code/ No Code einher. Citizen Development ist dabei aus meiner Sicht die oder der Ansatz, dass eben Nicht-Entwickler in die Lage versetzt werden, selber Applikationen und Workflows zu bauen, also in diese Digitalisierung reinzugehen und da geht es dann natürlich um Themen, wie Befähigung der Nutzer, die bisher keine Entwicklererfahrungen haben. Und typischerweise diese Citizen Developer, das sind typischerweise die Kollegen aus IT-fernen Abteilungen, zum Beispiel Fachabteilungen, irgendwie eine Einkaufsabteilung oder so was. Genau.

- 12 [0:04:22.9] **I: RS** Also welches Problem löst das? Also welchen Vorteil generiert man aus diesem Konzept?
- 13 [0:04:33.7] **B: TJ** Man muss sich ja vorstellen, bisher obliegt die Umsetzung aller digitalen Produkte, aller Prozesse, obliegt momentan der IT-Abteilung, wenn man sich da mal so umschaut, ist es eigentlich so, dass die IT-Abteilungen einfach ein riesiges Backlog mit sich her schleppen. Das heißt, große Anzahl an unbearbeiteten Aufgaben, an unbearbeiteten Anforderungen. Das ist so eine Art Flaschenhals geworden, weil es auch einfach an Fachkräften mangelt. Genau. Durch den Einsatz von Low Code/ No Code möchte man erreichen, dass man diesen Flaschenhals so ein bisschen auflöst, indem man Kollegen, die eigentlich nicht programmieren können, in die Lage versetzt, ihre, ihren Digitalisierungsbedarf selber vornehmen zu können. Diese Themen selber begleiten zu können, also sie würden dann in der Low Code/ No Code-Welt keine Anforderungen mehr an ihre IT-Abteilung oder an externe IT-Berater stellen, sondern würden sich anhand ihrer Anforderungen selber mit der Entwicklung beschäftigen, was zu schnelleren Entwicklungsergebnissen führt, weil die Anforderungen eben nicht erst umgesetzt und übersetzt werden müssen und dann auf freie Kapazitäten warten, sondern man als Fachabteilung im Prinzip seine Anforderungen selber steuern kann, direkt selber in die Umsetzung gehen kann. Zum anderen würde es auch die IT-Abteilung natürlich entlasten, wenn Anforderungen bei denen nicht mehr aufschlagen, weil sich die Fachabteilung selber damit beschäftigen kann.
- 14 [0:06:45.4] **I: RS** Siehst du da auch Nachteile jetzt, wenn du dir das jetzt so anguckst? Das IT-ferne Abteilungen so was machen?
- 15 [0:06:55.6] **B: TJ** Ja, also auf jeden Fall gibt es da Nachteile, man muss halt, wenn man sich vorstellt, dass jetzt die Fachabteilung so eine Einkaufsabteilung, jetzt selber sich Anwendungen bauen kann und Prozesse digitalisieren kann, muss man halt schon sagen, dass es zu dem umgangssprachlichen Wildwuchs führen kann, wenn jetzt jede Abteilung für sich losmacht und für sich die besten Anwendungen baut. Das heißt, man braucht im Hintergrund ganz klar das Thema Governance, damit muss man sich auseinandersetzen, wie man sozusagen das so ein bisschen orchestrieren kann, dass nicht jeder drauflos entwickelt, sondern dass das Ganze eben den gewissen Leitplanken folgt, weil es auch einfach schlussendlich IT-Security- Thema ist, wenn jeder was entwickelt. Niemand hält sich aber an gängige Vorgaben. Genau. Und ein weiterer Nachteil ist aus meiner Sicht, dass man das, glaub ich, organisationsweit nutzen muss, also man muss sich darauf verständigen, in der Organisation, dass es genutzt wird, weil, wenn es nur eine Abteilung gibt, die damit arbeitet, alle anderen arbeiten nicht damit, kann das Thema auch nicht richtig wachsen und skalieren, also dann hab ich einfach diese Vorteile, zum Beispiel einer Community, nicht, die kann sich dann nicht voll entfalten. Das heißt, ich müsste mehr in die Wissensvermittlung investieren, und dann, glaube ich, ist der Nutzen auch nicht so groß, wie wenn man das über die gesamte Organisation macht dieses Thema.
- 16 [0:09:03.5] **I: RS** Und wenn du jetzt mal wirklich das zusammenschrumpfen müsstest, was Low Code/ No Code- Plattform jetzt realistisch schon leisten könnten? Anhand der Vor- und Nachteile, die du gerade genannt hast. Was würdest du sagen, was sind also realistische Erwartungen, die man an diese Plattform stellen kann, zum jetzigen Zeitpunkt?
- 17 [0:09:24.9] **B: TJ** Also aus meiner Sicht, muss man natürlich immer schauen, dass es richtig in der Organisation verankert ist, aber wenn das einmal der Fall ist, dann kann man schon sagen, dass es zum einen die Innovation innerhalb der Organisation fördert, weil ich natürlich viel mehr Leute in die Lage versetze, dass sie selber einfach mal Ideen ausprobieren können. Und dass sie auch wesentlich schneller sind, als wenn sie sich jetzt mit Programmiersprachen oder so beschäftigen müssten, dass sie dann auch Ergebnisse zeigen können. Das heißt, ich glaube, ich rege die Mitarbeiter dazu an, auch selber mal Produkte zu verbessern, neue Produkte zu kreieren, das ist, glaube ich, für mich ein wesentlicher Faktor, den können

wir auch jetzt schon mit Low Code/ No Code-Plattformen erreichen. Auf der anderen Seite habe ich natürlich schnellere Entwicklungszeiten für meine Anwendungen und Prozesse, damit kann ich mich schneller auf Marktsituationen einstellen. Das heißt, ich habe nicht die Situation, wie es beispielsweise bei Corona war. Das ist jetzt ein Beispiel, ich erhalte jetzt die gesetzliche Auflage, dass ich den Impfstatus meiner Mitarbeiter kontrollieren muss, beziehungsweise die Testergebnisse meiner Mitarbeiter nachhalten muss. Ich würde das dann ohne Low Code/ No Code als Anforderungen an die IT geben. Ich bau dafür eine Anwendung. Die IT ist aber ausgelastet und das Ganze liegt erst mal, bis man in die Umsetzung starten kann. Mit Low Code/ No Code war es eben so, dass man aus ganz vielen Organisationen gehört hat, dass diese gesetzlichen Anforderungen, die ja neu waren, überraschend waren, nicht vorhersehbar waren, dass die innerhalb von Tagen bis wirklich wenigen Wochen so umgesetzt werden konnten, dass man eben auch ein digitales Produkt dafür hat, wo man diese Anforderungen abbilden kann und das ist natürlich ein Thema, das gäbe es mit einem traditionellen entwickeln nicht und ich kann natürlich auch, das vielleicht auch, als dritten Punkt, so die Flaschenhalse auflösen, die ich in meiner Organisation habe, wenn es darum geht, neue Produkte zu entwickeln, durch das Verteilen von Aufgaben auf mehrere Schultern.

18 [0:12:31.1] **I: RS** Und welche Voraussetzungen von Organisationen oder Unternehmen siehst du dann, wenn sie jetzt eine Power-Plattform oder ähnliche No Code/ Low Code-Plattformen einführen möchten? Also, was brauchen sie vielleicht an Kapital? Was müssen die Mitarbeiter an Befähigung mitbringen? Was muss das Mindset eventuell sein oder wie groß sollte so ein Unternehmen sein?

19 [0:12:57.3] **B: TJ** Also, ich glaube, 2 wichtige Faktoren sind da, dass einfach ein gewisses Mindset da ist in der Organisation, dass man also sagt, das ist ja ein vergleichsweise neues Thema. Low Code/ No Code, man muss ein bisschen neue Wege gehen bei dem Thema, weil man Zuständigkeiten umverteilt. Man muss mit der IT-Abteilung ins Rennen gehen und dieses Umdenken, was es eben benötigt. Dafür braucht man ein gewisses Mindset. Wenn man zu traditionell in seiner Denkweise ist und da auch nicht alle Leute motivieren kann, ist das, glaube ich, ein Thema, was einen schwierigen Stand haben wird. Demgegenüber braucht man einfach so eine gewisse Offenheit für Neues und man muss dem nicht nur offen gegenüberstehen, sondern man muss dem Thema halt auch mal Raum geben, sich zu beweisen also, dann braucht es ein gewisses Maß an Aufwand, der da initial investiert werden muss, damit man Ergebnisse sehen kann und über die Ergebnisse kann man dann auch schneller einschätzen, das jetzt gutes Thema ist für mich, ob es mir helfen kann. Oh, also dieses Mindset-Thema sehe ich als sehr wichtig an. Auf der anderen Seite, glaube ich, auch so eine gewisse Herangehensweise, wie ich mit meinen Mitarbeitern umgehe. Also da fallen mir so die Themen kontinuierliches Lernen ein, das man eben sagt: Ich muss meinen Mitarbeitern zutrauen, dass sie dieses mehr an Möglichkeiten auch sinnvoll umsetzen und ich muss eben auch da den Aufwand in Kauf nehmen, dass sich die Mitarbeiter erst mal auf die neue Plattform, also sie müssen lernen, sich darin zu bewegen in dieser Plattform und einfach Wissen aufzubauen in dieser Plattform. Diese beiden Punkte, wohingegen ich sagen muss, so Themen, die du auch angesprochen hast, Unternehmensgröße oder Kapital, sehe ich da als nicht so wichtig an. Es gibt Start-ups, die mit Low Code/ No Code arbeiten. Es gibt Konzerne, die mit Low Code/ No Code arbeiten. Und auch das Thema Kapital ist, glaube ich, eher zweitrangig, weil da einfach für viele Preisklassen sich eine Low Code/ No Code-Plattform anbieten kann. Es ist jetzt kein K.-o.-Kriterium, wenn man jetzt gerade zum Beispiel als Start-up oder am Anfang noch, ja, das Budget sehr überschaubar ist, das soll einen nicht davon abhalten, auf diese Plattform zurückzugreifen.

20 [0:16:18.3] **I: RS** Und welche Problemstellungen siehst du dann, wenn Unternehmen das einführen? Du sagst ja, dass es jetzt unabhängig von der Unternehmensgröße ist. Was könnten Probleme sein, auf die Unternehmen stoßen?



- 21 [0:16:29.7] **B: TJ** Probleme, die ich sehe, sind zum einen, dass das Thema unterschätzt wird, dass es als IT-Thema betrachtet wird, was es aber eigentlich nicht ist, sondern dieses Thema muss halt höher aufgehängt werden als, in Führungsstrichen, nur in der IT. Möchte damit nicht die IT-Abteilung abwerten. Aber es ist halt ein Thema, was die ganze Organisation betrifft, weil man ja gerade Verantwortlichkeiten aus der IT herausholen möchte und in andere Abteilungen hineinbringen möchte. Und damit kann ich beispielsweise nicht nur den Head of IT damit betrauen, sich dieses Projekt mal anzuschauen, sondern es muss einfach der Sponsor, der muss weiter oben in der Organisation sitzen. Zum anderen muss es in der Organisation einfach ganzheitlich betrachtet werden, das heißt: Es wäre schade für das Projekt, wenn man sagt, ich schaue mir Abteilungen A und B an und die Abteilungen C bis J lass ich bei dem Thema raus, weil ich das bei denen nicht sehe. Sondern ich muss im Prinzip das Thema so weit wie möglich versuchen, in der Organisation zu verankern. Governance ist noch ein ganz großer Punkt, das hatte ich vorhin schon mal kurz angesprochen, dass man versucht, durch interne Leitplanken diesem Wildwuchs vorzubeugen, dass einfach klar ist, wie ist die Aufgabenteilung. Wer darf was machen, welche Fachabteilung darf was bauen, wo sind aber gegebenenfalls Aufgaben, die bei der IT-Abteilung liegen bleiben und es muss natürlich auch klar sein, dass es ein Thema ist, dem man sich immer wieder widmen muss. Also man hat nicht einmal Low Code/ No Code-Plattform eingeführt und ist dann damit fertig, sondern es ist natürlich ein Thema, was kontinuierlich verbessert werden muss. Man muss kontinuierlich zum Beispiel neue Mitarbeiter, welche die Organisation neu betreten, die muss man in dieses Thema einführen, man muss gucken, wo knackt es, wo gibt es Sachen, die noch nicht so gut laufen, und da muss man nachsteuern, also diese Bereitschaft, das Ganze kontinuierlich zu begleiten. Genau.
- 22 [0:19:17.3] **I: RS** Siehst du da hauptsächlich, also ein strukturelles Vorgehen, wo die Probleme auftauchen können, oder habe ich dich missverstanden?
- 23 [0:19:31.4] **B: TJ** Genau. Also zum einen ist es natürlich so, dass das Projekt in seiner Tragweite unterschätzt werden kann. Das kommt ja sehr IT-nahe daher, man muss aber schon sehen, dass man es schafft, das Ganze für die komplette Organisation zu denken, und man muss auch gucken, dass ja, diese Bereitschaft, die man in dem Moment hat, wenn man sich mit der Low Code/ No Code-Plattform beschäftigt, dass die eben auch dableibt, also nicht im Sinne von wir machen das jetzt ein halbes Jahr. Wir führen das ein und dann können wir alle Projektmitglieder da rausziehen, weil wir das jetzt leben, sondern das muss natürlich weiterbearbeitet werden. Es muss verbessert werden, es muss weiterhin begleitet werden.
- 24 [0:20:31.9] **I: RS** Und welche Organisationsform aus Projektmanagement-Sicht würdest du als am sinnvollsten für sowas dann erachten?
- 25 [0:20:40.2] **B: TJ** Es gibt ja die unterschiedlichsten Organisationsformen. Wobei ich das, glaube ich, losgelöst von den Organisationsformen sehe. Also ich würde sagen, die Organisationsform ist nicht so wichtig. Wichtiger sehe ich, dass das Commitment bei allen relevanten Teilnehmern für dieses Thema da ist, also es muss halt klar sein, dass alle Projektmitglieder und der Sponsor am selben Strang ziehen und auch das gleiche Ziel vor Augen haben. Das sehe ich als wichtiger an, und dann ist mir das egal, ob das jetzt eine sehr hierarchisch geprägte Organisation ist oder ob die Organisation in Projekten arbeitet. Oder wie auch immer. Das sehe ich als entscheidenden Faktor an, dass eben alle relevanten Teilnehmer am selben Strang ziehen.
- 26 [0:21:46.0] **I: RS** Kurze Zwischenfrage. Wie definierst du Sponsor? Also was ist ein Sponsor für dich?
- 27 [0:21:52.9] **B: TJ** Kurz gesagt, es ist, glaube ich, derjenige, der das Geld gibt. Und derjenige, der im ersten Moment, wenn er das Geld gibt, auch die Verantwortung für das Thema trägt, also diese beiden Punkte: er stellt das Budget zur Verfügung und im ersten Moment, man

stelle sich vor, nach einem halben Jahr, stellt man das ganze Projekt ein. Das bereitgestellte Budget ist weg und vertan, dass er dann auch die Verantwortung im ersten Moment trägt, also diese beiden Punkte, Budget und Verantwortung, bringt er initial ein, wobei sich das dann natürlich über Projektlaufzeit und auch einer gewissen Etablierung im Unternehmen dann noch mal ändern kann, aber in der ersten Stufe sehe ich da den Sponsor bei den beiden Themen.

- 28 [0:22:54.0] **I: RS** Und würdest du eher empfehlen, dass die Teammitglieder aus ihren Fachabteilungen herausgelöst werden oder in den Fachabteilungen trotzdem noch verbleiben?
- 29 [0:23:03.7] **B: TJ** Ich glaube, am Anfang ist es wichtig, dass sie in ihren Fachabteilungen bleiben. Dass sie einfach den Bezug zu den Fachthemen haben. Es sollte ja so sein, dass man mit dem Thema schnell Erfolge produziert, dafür ist es halt wichtig, dass man auch Leute an Bord hat, die wissen, was in der Unternehmensrealität wesentlich ist, die auch wesentliche Probleme in der Unternehmensrealität kennen und wie man diese gegebenenfalls mit einer Low Code/ No Code-Plattform lösen kann. Und wenn man das einmal schafft, so eine Lösung zu kreieren, dann kann man natürlich auch durchs Unternehmen gehen und sagen, dass man im Prinzip diese Ergebnisse mithilfe der Low Code/ No Code-Plattform geschaffen hat und ja, für weitere Unterstützer sozusagen werben. Also am Anfang würde ich die Teammitglieder/ Projektmitglieder in ihren jeweiligen Abteilungen belassen und dann eher zu einem gewissen Prozentsatz für dieses Thema freistellen. Beziehungsweise, was heißt freistellen, denen die Aufgaben übertragen, zu einem gewissen Prozentsatz ihrer Kapazität. Wenn das Thema einmal eingeführt ist, gibt's dann, glaub ich, schon Kollegen in der Organisation, die nicht mehr in der Fachabteilung drin sind, sondern die sich dann, ja, um dieses Thema Low Code/ No Code kümmern, um dieses Thema Community kümmern, die sich dann um dieses Thema Enabling der Mitarbeiter kümmern werden. Also das wird sich dann über die Zeit, über die Projektlaufzeit noch mal bisschen ändern. Auch am Anfang sollten die Kollegen in der Fachabteilung bleiben.
- 30 [0:25:11.6] **I: RS** Und wenn du jetzt ein Vorgehen quasi bestimmen müsstest, welche Vorgehensweise würdest du dabei am besten wählen? Also eher ein typisches sequenzielles Vorgehen in Phasen oder eher ein agileres Vorgehen oder eine Mischung aus beiden oder sagst du, es gibt ein komplett anderes Modell, was du jetzt hier bevorzugen würdest?
- 31 [0:25:35.1] **B: TJ** Es gibt kein anderes Modell, also ich glaube, die Entscheidung liegt in der Regel zwischen diesen 3 Formen, die du genannt hast, wobei man da immer schauen muss. Wo die Organisation, die das Thema einführen möchte, beziehungsweise beleuchten möchte, wo die herkommt. Wenn sie zum Beispiel bisher sehr klassischen Projektansatz hat, dass dort die Projekte immer mit Wasserfall zum Beispiel bearbeitet werden, mit Wasserfallmethoden, dann macht das, glaub ich, keinen Sinn, in diesen Unternehmen das neue Thema Low Code/ No Code und Citizen Development in einem agilen Vorgehen einzuführen, weil es einfach zu einer doppelten Unsicherheit führt. Also ich würde in der Regel das Vorgehen nehmen, was ich in der Organisation vorfinde. Wobei ich denke, dass wenn man die Wahl hat, glaube ich, ein hybrider Ansatz eine ganz gute Lösung ist.
- 32 [0:26:51.7] **I: RS** Und wie würde dieser hybride Ansatz aussehen, den du jetzt für dich favorisieren würdest?
- 33 [0:27:00.2] **B: TJ** Na, ich würde im Prinzip, wenn man jetzt in beide Methodenkoffer reinschaut, würde ich mir im Prinzip die Sachen raussehen, die ich als wichtig erachte für das Thema Low Code/ No Code. Das heißt, ich würde die Methoden durchaus mischen und ja, dann zu einem, ja nicht ein neues Modell, aber man hat ja dann immer auf der einen Seite klassisch, auf der anderen Seite agil und irgendwo in der Mitte, dann muss man für sich eben diese Mitte finden, wie man das hybride Modell definiert, aber das wäre so meine Herangehensweise, dass ich schaue, welche Methoden aus welchem Koffer passen für mich

- besonders gut. Und genau die würde ich dann eben, ja, zusammenfassen. Für das Vorgehen in dem konkreten Projekt.
- 34 [0:28:05.3] **I: RS** Und wenn du jetzt die freie Wahl hättest, du könntest am agilen Koffer und aus dem sequenziellen Koffer oder dem traditionellen Koffer Sachen rausnehmen, was würdest du gerne mit aus dem Jeweiligen nehmen? Also, zum Beispiel, die Phasen aus dem sequenziellen und das Daily Scrum vielleicht, als Beispiel, aus dem agilen oder wie würdest du das jetzt als ideal oder für dich ideal sehen, was man da Schönes reinnehmen könnte?
- 35 [0:28:30.3] **B: TJ** Für das konkrete Low Code/ No Code-Einführungsprojekt empfinde ich es als wesentlich, dass man das Augenmerk so ein bisschen auf den Projektaufsatz legt, das heißt, da bin ich so ein bisschen in der klassischen Welt unterwegs. Bei Project Management Institute zum Beispiel heißen die ersten beiden Projektphasen Initiierung und Planung, dass man sich einfach anschaut: Wer ist denn ein geeignetes Team, um dieses Projekt anzugehen, sind die wirklich alle motiviert dazu? Ich muss mir in der Phase auch anschauen, wer sind die Stakeholder? Mit wem muss ich wie umgehen während der Projektlaufzeit und ich muss auch mit allen Beteiligten den Scope klar abgrenzen, also es muss sozusagen klar sein, wo die Reise hingehen soll. Dass, ja, dass es eben dazu führt, dass eben alle an einem Strang ziehen. Auf der anderen Seite würde ich von der Seite der agilen Methoden, kann ich mir vorstellen, dass man durchaus auch in Sprints arbeitet, also wenn man zum Beispiel darüber spricht, dass man in einer iterativen Vorgehensweise einzelne Fachbereiche in dieses Projekt hinzuholt und die darin schult. Das wäre eine Sache, wo man iterativ vorgehen könnte, einfach damit man die Gelegenheit hat, aus den Erfahrungen des einen Onboarding des Fachbereiches für den nächsten zu lernen. Ich würde bei den agilen Methoden auch Wert darauf legen, schnell Erfolge sichtbar zu machen, dass man also nicht mit einem beispielsweise Big Bang arbeitet und alles passiert im Kämmerchen bis das ganze Thema leicht geht, sondern man muss eben zwischendrin auch die Erfolge sichtbar machen. Also immer ein Stückchen mehr, ja, Inkrement sozusagen, ein Stückchen mehr zeigen, was man geschaffen hat, um die Projektmitglieder einfach am Ball zu behalten und auch den Stakeholdern zu zeigen, dass man hier konsequent Mehrwert schafft für die Organisation. Und ich bin selbst auch immer ein großer Freund von Retrospektiven, also, das wäre auch ein Thema, was ich aus der agilen Welt mit einbeziehen würde, dass man sich einfach regelmäßig hinterfragt, was haben wir geschafft? Wie lief das? Also, wie war der Weg bis hierher? Was können wir einfach für die Zukunft noch besser machen? Also, das wären so ein paar Anteile, die ich aus beiden Methoden-Koffern nehmen würde.
- 36 [0:31:50.2] **I: RS** Und wenn du dir das jetzt so anguckst, wie würdest du dann mit Änderungswünschen umgehen, die aus dem Projekt herauskommen? Beziehungsweise von außerhalb an das Projekt gestellt werden.
- 37 [0:32:05.0] **B: TJ** Also, ich würde das, glaube ich, mit dem Team zum einen besprechen, das Team muss es ja vertreten. Die Änderungswünsche, es muss sie umsetzen, das heißt, das Team muss da einfach dabei sein, dass das eine sinnvolle Änderung ist. Und ich würde auch immer den Abgleich machen mit dem Ziel, was ich ursprünglich mal für das Projekt ausgegeben habe, sind wir denn noch auf dem richtigen Weg? Also so ein bisschen dieser Blick zurück, schaffen wir es, mit dieser konkreten Änderung jetzt, dem ursprünglichen Projektziel näherzukommen? Schaffen wir Mehrwert für die Organisation? Das wären so die Fragen oder die beiden Seiten, die ich da beleuchten würde, also einmal: Das Team muss da dabei sein und das mit vertreten und auf der anderen Seite muss ich dem ursprünglichen Ziel durch die Änderung näherkommen und wenn das beides passt, dann kann man das gerne mit in die Umsetzung nehmen, die Änderungswünsche.
- 38 [0:33:16.9] **I: RS** Welche wichtigen Rollen und Kompetenzen siehst du da im Projekt? Wenn du jetzt sagst, das ist so ein hybrider Ansatz, was sind da die wichtigen Rollen, die du siehst?

- 39 [0:33:30.5] **B: TJ** Also, sehr wichtig aus meiner Sicht ist halt jemand, der das ganze Thema vorantreibt. Ja, so ein bisschen Begeisterung für das Thema schafft, der das einmal auf die Agenda legt. Bei den ganzen Rollen in der Organisation, die darüber entscheiden. Also das ist, glaube ich, eine zentrale Rolle, weil das am Anfang erfahrungsgemäß so ein kleiner Kampf gegen Windmühlen ist. Man darf nicht davor zurückschrecken, sondern man muss das Thema eben kontinuierlich treiben und das klappt am besten, wenn man selber dahintersteht, also im besten Fall hat der, ich nenne ihn jetzt mal Projektleiter für dieses Thema, eben eine intrinsische Motivation für das Ganze, weil er selber zum Beispiel da ein Leidenthema hatte, das irgendwie eigene Anforderungen von ihm, in einem seiner Projekte bei der IT-Abteilung zu lange lag, oder wie auch immer, aber er muss selber den Mehrwert sehen und das Ganze auch dahin treiben und zum anderen auch die übrigen Projekt-/ Teammitglieder, die müssen eben auch im besten Fall selber darauf Lust haben, selber den Mehrwert erkennen. Und es darf nicht so dieses Gefühl haben, es wird ihnen jetzt übergestülpt, diese Verantwortlichkeit und jetzt müssen sie das das nächste Jahr mal machen und dass es so das ungeliebte Projekt wird in meinem Aufgabenkalender, sondern es muss eben, ich glaube, da muss man jemanden finden, der einfach Lust auf das Thema hat und darum mitzieht. Genau, das wären so die 2 wesentlichen Rollen. Augenmerk ist ganz klar auf dem Projektleiter, der das Thema einfach treibt und vorantreibt und sich Unterstützung in der Organisation dafür sucht.
- 40 [0:35:53.7] **I: RS** Würdest du auch den Sponsor, wie du ihn vorhin genannt hast, als eine wichtige Rolle sehen? Oder ist das eher jemand, der im Hintergrund trotzdem agieren sollte und jetzt nicht dabei treiben sollte oder wie ordnest du die Rolle ein?
- 41 [0:36:11.6] **B: TJ** Ja, also beim Sponsor bin ich mir unsicher, ob er zum Projektteam gehört oder nicht. Auf jeden Fall ist der Sponsor auch eine wichtige Rolle, weil er das Thema, glaube ich, auch noch mal auf einer anderen Ebene vertritt. Also wenn man sich vorstellt, dass der Sponsor regelmäßig aus einer anderen Hierarchieebene kommen wird. Das heißt, er bewegt sich auf einer anderen Hierarchieebene und in der Organisation und kann das eben auf dieser höheren Ebene auch besser vertreten als zum Beispiel der Projektleiter. Aber ansonsten muss der Sponsor, glaube ich, offen sein für das Neue, da ist wieder dieses Thema Mindset. Das muss er auf jeden Fall mitbringen und ob er dann dafür noch aktiv Werbung macht für dieses Thema oder nicht? Das würde ich jetzt dem Sponsor überlassen. Ich glaube, viel Wert ist einfach, wenn dieses Buy in vom Sponsor da ist, dass der das Projekt unterstützt und sich da auch so ein bisschen dahinter stellt, genau, wie aktiv der jetzt wird, das möchte ich gar nicht sagen. Ich glaube, da gibt es unterschiedliche Ansätze. Es genügt auch, wenn der Sponsor das Budget gibt und am Anfang in die Verantwortung geht und dann kann, glaube ich, auch die Projektleitung viel, viel ziehen und viel leisten.
- 42 [0:37:50.7] **I: RS** Wir haben angenommen, du hast jetzt ein Projekt durchgeführt so weit und bist jetzt quasi in der Prototyping Phase, wo du sagst, wir haben das jetzt ganz klein schon für uns entwickelt. Wie würdest du die Skalierung machen? Vom Prototyping bis zur Serienreife? Also bis es in der Organisation ausgerollt werden kann? Also, wie ist die Skalierbarkeit, um die Frage vielleicht so einzuschränken?
- 43 [0:38:19.2] **B: TJ** Ja genau, also typischerweise muss man ja schnell Erfolge zeigen, damit man einfach immer mehr Leute auf seine Seite holt und von dem Projekt begeistert. Das heißt, im besten Fall steht am Ende des Prototypings ein praxisrelevanter Use Case für das Unternehmen, wo man mal Applikationen gebaut hat, mit dieser Applikation, die ein echtes Problem im Unternehmen löst, kann man dann in andere Abteilungen gehen und sagen: „Guck mal, was wir hier gemacht haben, das Ganze, innerhalb von beispielsweise einer Woche umgesetzt, erspart uns irgendwie den Aufwand von sonst 2 Monaten.“ Also, da muss man dann so ein bisschen eben die Benefits darstellen und das ganze Thema, ja, mehr oder weniger unter die Leute bringen. Je nach Unternehmensgröße muss man dann halt gucken, ob das schon genügt, ob man damit sozusagen weitere Abteilungen dazu animieren kann,

bei dem Thema auch dabei sein zu wollen. Ansonsten muss man halt gucken, dass man vielleicht noch 1 – 2 weitere Use Cases umsetzt, damit man so eine Art Blumenstrauß an Lösungen hat, auch vielleicht, um die Vielseitigkeit zu zeigen. Genau, und wichtig ist aus meiner Sicht eben immer, dass man das Ergebnis teilt, dass man den Erfolg feiert, einfach um Lust in anderen Unternehmensteilen zu schaffen. Sobald man ausreichend Interesse und Bewusstsein für die Plattform gesammelt hat in den anderen Unternehmensteilen, kommt man, glaube ich dahin, dass man die Strukturen aufsetzen muss, also wir bewegen uns jetzt weg von 1 bis 3 Einzelfalllösungen, sondern wir wollen jetzt das Thema organisationsweit ausrollen und da muss ich mir dann Gedanken machen, wie stell ich denn die Governance zum Beispiel auf für diese Plattform? Also was sind die Leitplanken, in denen ich die Fachabteilungen berechtige, etwas umzusetzen? Was sind denn die Leitplanken, wo die IT-Abteilung immer noch unter die Arme greift, weil, es wird nicht ohne einander gehen, weil da beide Abteilungen müssen sozusagen zusammenarbeiten? Ich muss mir Gedanken machen über eine Community. Wie ich das Wissen in der Organisation verteile und gegebenenfalls ist das auch der Punkt, wo feste Rollen für dieses Thema Low Code/ No Code beziehungsweise Citizen Development im Unternehmen ins Spiel kommen. Also, beim Thema Community, wer managt die Community? Beim Thema Governance, wer arbeitet das aus, stimmt das mit den einzelnen Abteilungen ab, wenn die IT-Abteilung die Governance vorgibt, kann es sein, dass die Fachabteilung so ein bisschen hinten runterfällt? Wenn die Fachabteilung Governance vorgibt, wird ziemlich sicher sein, dass das mit der IT-Abteilung nicht zu hundert Prozent abgestimmt ist. Im besten Fall findet man jemanden, der weder in der einen noch in der anderen Welt zu Hause ist, sondern der sich wirklich in Zukunft um dieses Thema Citizen Development Low Code/ No code kümmern wird. Das wäre also so eine erste feste Rolle, die man im Unternehmen dann finden würde. Und wenn die Strukturen dann da sind, um das Thema ausweiten zu können, würde ich dann an die Skalierung gehen, also ich würde mir die nächste Fachabteilung nehmen oder je nachdem auch mehrere, aber nicht alle. Ich würde mir sozusagen die nächsten Fachabteilungen nehmen, die in dieses ganze Projekt oder über dieses Projekt begleiten, die mit den Themen vertraut machen und sie schlussendlich onboarden, also sie in die Lage versetzen, dass sie selber die Low Code/ No Code-Plattform nutzen können. Und da auch wieder der Rückschluss zu dem agilen Arbeiten, also aus dieser ersten Iteration lernen und die nächste Iteration mit der nächsten Fachabteilung dann besser machen und das wäre dann so eine Art Inkubator, wo man sagen kann, man lässt die Fachabteilung nacheinander dieses Programm durchlaufen. Danach sind die Fachabteilungen in der Lage, selber diese Plattformen zu nutzen. Sie wissen, welche Governance, also welche Regeln, sie einhalten müssen, welche Verantwortlichkeiten es für das Thema gibt, sie wissen, wohin Sie sich mit Fragen wenden müssen, nämlich an die Community. Genau. So also, das wäre dann dieses Thema Skalierung und das, glaube ich, über allem so drüber schwebt, wäre eine Phase, die könnte man dann Innovationen nennen, dass man also sagt „Ok, ich hab jetzt die Organisationseinheiten onboarded, sie nutzen die Plattform, es kommen Ergebnisse raus und jetzt schauen wir, wie wir das Ganze eben kontinuierlich verbessern können.“ Also wir wollen uns nicht auf dem eigenen Stand ausruhen, das wird auch nicht funktionieren, weil die Plattform wächst, also die Low Code/ No Code-Plattform wird wachsen. Die Funktionalitäten werden sich ändern und die Skills der Fachabteilungen werden wachsen und das ist einfach etwas was, für das man sich ständig mit diesem Thema beschäftigen muss und seine einmal getroffenen, zum Beispiel Leitplanken, auch immer wieder hinterfragen sollte. Das wäre dann so dieses Thema Innovation, wäre dann so, dass das letzte beziehungsweise eigentlich auch wieder das Thema, wo dann die Iteration die nächste Iteration startet.

44 [0:45:03.2] **I: RS** Und wie also, wenn man jetzt am Ende von dem Projekt ist, wie würdest du die Qualität von dem Projekt sichern beziehungsweise, wie misst du den Erfolg?

- 45 [0:45:15.9] **B: TJ** Also ich glaube, Erfolg kann man in verschiedenen Richtungen messen. Zum einen den gefühlten Erfolg, dass man also über zum Beispiel Retrospektiven herausfindet, was denken die Mitarbeiter dazu? Was ist der große Vorteil der in Anführungsstrichen Neuen Welt? Zum anderen kann man aber auch dahin übergehen, dass man richtige KPI's definiert, die man eben messen will und wo man dann beide Welten miteinander vergleichen kann, also zum Beispiel eine durchschnittliche App-Entwicklungsdauer könnte man jetzt festlegen und die vergleichen. Man könnte auch, das ist jetzt sehr organisationsbezogene, je nachdem, was man messen möchte und was Sinn macht, aber man könnte auch den gesamten Aufwand einer Applikation messen also, was hat es mich gekostet, diese Anwendungen insgesamt zu entwickeln. Genau. Da gibt es dann verschiedene Ansätze, die man dann einmal definieren muss, das kann man in dem Einführungsprojekt machen und die dann auch regelmäßig gemonitort werden sollten.
- 46 [0:46:49.0] **I: RS** Dann sind wir jetzt auch schon fast wieder am Ende von unserem Interview. Hast du noch Anmerkungen beziehungsweise Best Practices zu share, wo du sagst, das würde ich als wichtig empfinden, das wurde aber jetzt nicht gefragt zum Beispiel.
- 47 [0:47:11.0] **B: TJ** Ich glaube, es wurde alles gefragt, beziehungsweise würde meinen, es ist irgendwie alles untergebracht. Wichtig ist mir nur, in diesem ganzen Projekt, sind zum einen die Menschen, also die Mitarbeiter, die Kunden, das ganze Thema muss einen Mehrwert für die haben. Ansonsten fliegt es nicht. Und ich muss auch zwingend mir Gedanken machen über das Thema Governance. Ansonsten ist es toll, dass die Menschen damit arbeiten können, aber ich glaube, dann geht mit der Zeit auch der Nutzen verloren, weil es einfach zu unübersichtlich wird, das Ganze zu verwalten und es geht nur miteinander. Also mir bringt die beste Governance nichts, wenn ich die Menschen nicht mit an Bord hole, in dem Projekt, wenn es also niemand nutzt. Und wie gesagt, mir bringen die begeisterten Nutzer der Plattform nichts, wenn ich das nicht schaffe, in sinnvolle Leitplanken zu geben, diese Projekte. Genau.
- 48 [0:48:23.8] **I: RS** Dann ganz vielen Dank und ich würde jetzt die Aufnahmen beenden.
- 49 [0:48:29.1] **Ende**

## Einwilligungserklärung Interview

Ich erkläre hiermit mein Einverständnis zur Nutzung der personenbezogenen Daten, die im Rahmen des folgenden Gesprächs erhoben wurden:

Datum des Interviews	30.05.2022
Namen der zu interviewenden Person	Tino Janke
Name der interviewende Person	Rocco Schenk
Kurzbeschreibung des Forschungsprojekts	Einführung von No Code/ Low Code aus Sicht des Projektmanagements
Durchführende Hochschule	Hochschule Mittweida – University of Applied Science

Die Daten werden im Rahmen eines mündlichen Gesprächs erhoben, das mit einem Aufnahmegerät und Videoaufnahmegerät aufgezeichnet wurde. Zum Zwecke der Datenanalyse werden die mündlich erhobenen Daten verschriftlicht (Transkription), wobei die Daten nicht anonymisiert werden.

Vorhandene Kontaktdaten, der interviewten Person, werden aus Dokumentationsgründen in einem separaten Schriftstück lediglich den Gutachter\*innen der wissenschaftlichen Ausarbeitung zur Verfügung gestellt. Nach dem Abschluss des Projekts werden diese Daten gelöscht.

Der Speicherung der personenbezogenen Daten zu Dokumentationszwecken kann durch die interviewte Person jederzeit widersprochen werden. Die Teilnahme an dem Gespräch erfolgt freiwillig. Das Gespräch kann zu jedem Zeitpunkt abgebrochen werden. Das Einverständnis zur Aufzeichnung und Weiterverwendung der Daten kann jederzeit widerrufen werden.

Tino Janke  
Vorname und Name in Druckbuchstaben

T.  
Unterschrift

Leipzig, 14.07.22  
Datum/Ort

## Anlagen 9, Interview EB08

RS: Rocco Schenk

PF: Peter Fey

1	[0:00:00.0] <b>Start ( PF- Peter Fey/ RS - Rocco Schenk)</b>
2	[0:00:03.9] <b>I: RS</b> Dann heie ich dich erst mal herzlich willkommen. Zuerst mchte ich dich fragen, bist du mit der Aufnahme des Interviews einverstanden?
3	[0:00:13.7] <b>B: PF</b> Ja, ich bin mit der Aufnahme des Interviews einverstanden.
4	[0:00:17.0] <b>I: RS</b> Die Videoaufzeichnung wird zum Zwecke der Transkription jetzt angefertigt und am Ende aber wieder gelscht, wenn die Masterarbeit fertig ist. Das Transkript wird in den Anhang meiner Masterarbeit kommen. Du wirst nicht anonymisiert. Das einfach als vorab-Infos. Und dann wrde ich gerne direkt damit einsteigen, dass du dich erst mal vorstellst. Wer du bist? Wie dein Kontext innerhalb des Projektmanagements ist und zu Low Code?
5	[0:00:44.9] <b>B: PF</b> Also mein Name ist Dr. Peter Fey. Ich bin Angestellter bei der T-Systems Multi Media Solutions, arbeite fr dieses Unternehmen seit 2015. Habe davor fr die T-Systems, die Muttergesellschaft der MMS, seit 2010 als fest angestellter Mitarbeiter gearbeitet. Habe dann in den Jahren 1996 bis 2004 und von 2006 bis 2010 im Rahmen der Freelancerttigkeit fr die Telekom wie auch fr die T-Systems gearbeitet. Zuvor war ich bei den Unternehmen Treuarbeit Unternehmensberatung, die dann bergingen in Coopers und Lybrand Unternehmensberatung, die dann wiederum bergang in Pricewaterhouse/ Coopers und habe dann auch im Nachhinein fr die IBM Business Consulting gearbeitet. Das ist etwas zu meinem beruflichen Hintergrund. Die Firmen, fr die ich gearbeitet habe, die Firmen, bei denen ich gearbeitet waren unter anderem die Deutsche Lufthansa, die Deutsche Bahn. Dazu zhlten halt auch die Lotto-Gesellschaften mit ihren beginnenden Online-Spielen in den 1990er-Jahren und jetzt im Rahmen der letzten Projekte dann halt auch fr die DPDHL, fr Daimler und andere groe Unternehmen. Das ist jetzt zu meinem beruflichen Werdegang, in welchem Rahmen, bei welchen Unternehmen ich gearbeitet habe und ich selber bin noch darber hinaus in einer ehrenamtlichen Funktion bei dem Project Management Institute ttig. Dieses Project Management Institute ist eine weltweite Vereinigung von professionellen Projektmitarbeitern/ Projektleitern und im Rahmen dieser PMI-Mitgliedschaft bin ich zustndig fr die Community of Practice Citizen Development, da geht es um No Code/ Low Code-Entwicklungen und ich bin zudem noch head of Cologne Group. Also einer Gruppierung, die in dem Bereich Kln und Umgebung, also Bonn und andere Orte noch, ttig ist. Ja, das sind so die Eckpunkte, die man da ber mich sagen kann.
6	[0:03:28.1] <b>I: RS</b> Da hast du einen sehr erstaunlichen Lebenslauf auf jeden Fall. Knntest du mir mal sagen, wie du No Code/ Low Code definieren wrdest, jetzt aus deinem Kontext?
7	[0:03:40.9] <b>B: PF</b> Ja also Low Code oder No Code sind im Endeffekt die Mglichkeiten der grafischen Programmierung oder der Programmierung ber grafische Benutzeroberflchen und es sind keine einzelnen Programme, sondern es sind groe Plattformen, die von den Herstellern angeboten werden und sie sollen es ermglichen, ohne jetzt in das Schreiben von Codes einzusteigen, massive, groe Programme, groe Applikationen zu realisieren. Angefangen von kleineren Workflows im Rahmen einer Abteilung, bis hin zu groen unternehmensweiten Ablufen. Also von der Produktion ber den Verkauf bis zur Abrechnung bis



zum Lagerwesen, das kann man dann alles tun, ohne jetzt in eine Programmierung einzu-  
steigen und auf Basis dieser Plattform.

- 8 [0:04:42.3] **I: RS** Und mit diesen Plattformen im Kontext. Weißt du, was Citizen Development ist?
- 9 [0:04:51.4] **B: PF** Ja, das Citizen Development ist die Vorgehensweise. Also Citizen Development, halt im Endeffekt, wenn man auf das Deutsche überträgt, den Begriff Fachbereichsentwicklung und es geht einfach darum, wenn ich jetzt einen Fachbereich ermächtige, große Workflows für die Abläufe auf IT-Basis für Unternehmen zu erstellen, dann muss ich halt, unabhängig davon, ob die Leute Code schreiben wollen oder nicht, muss ich die ganzen Umgebungsbedingungen, die muss ich halt auch bedenken. Das heißt, es geht um Datensicherheit bei der Richtung, dass einerseits die Sicherheit der Daten, aber andererseits die Verweigerung von unberechtigten Zugriffen auf die Daten, also das ist die Datenkonsistenz zum Beispiel, das gehört auch noch zur Datensicherheit. Das ist ein Aspekt, dann die Governance. Wer darf mit welchen Tools eigentlich arbeiten. Das gehört halt auch in den Bereich der Vorgehensweise und hat mit den Plattformen, die Low Code und No Code ermöglichen, nur bedingt zu tun. Also diese Fragestellung wäre immer, auch wenn ich Code schreiben würde, wäre immer zu beantworten. Und das ist der Bereich von Citizen Development und in diesem Citizen Development sind dann viele, viele Vorgehensweisen, die man halt aus dem agilen Umfeld kennt, aber halt auch aus der Teammotivation heraus kennt, aus der Teamführung, transformationelle Führung zum Beispiel. Die spielen da eine große Rolle und die sollten im Citizen Development entsprechend auch berücksichtigt werden. Zum Beispiel bei Ideation Workshops, aber auch in der Kommunikation von Lösungsvorschlägen, einer Psychological safety, die dann gegeben sein muss. So dass jeder seinen Vorschlag machen kann, ohne dass Gefahr zu laufen, als Nichtfachmann dann diskreditiert zu werden, sondern genau diese Ideen bringen die Abteilungen weiter. Die sozusagen sich in neues Terrain vorwagen und altbekannte Möglichkeiten verlassen, ist halt auch nicht im Rahmen der Low Code/ No Code-Plattformen ein Bestandteil, aber im Citizen Development, als umfassendere Vorgehensmethode, soll es berücksichtigt werden.
- 10 [0:07:18.3] **I: RS** Und mit dem, was du jetzt gesagt hast, welche expliziten Vorteile siehst du dann von Low Code/ No Code-Plattformen, auch in Verbindung mit Citizen Development?
- 11 [0:07:30.3] **B: PF** Ja also die Low Code/ No Code-Plattformen, die habe ich ja eben schon beschrieben. Im Prinzip kann man also hingehen und gegen den Einwurf kleiner Münzen bei den herstellenden Unternehmen, die natürlich programmieren müssen, diese Plattformen sich aneignen und die, ich sage mal, die Gebrauchsbücher, Handbücher lesen und dann dort die großen Applikationen schreiben, die das Unternehmen tatsächlich aus Sicht der Fachabteilung braucht. Es ist keine Abstimmung notwendig zwischen der IT-Abteilung und den Fachabteilungen. Es ist, der Informationsverlust kann geschmälert werden dadurch, bevor ich dann so eine Applikation in die Tat umsetze. Und das sind alles große Vorteile, die die Low Code/ No Code-Applikationen mit sich bringen. Und der Unterschied zu Citizen Development, wenn ich die Frage jetzt richtig interpretiert habe, besteht halt darin, dass das Citizen Development, die Fachbereichsentwicklung, diese No Code/ Low Code-Plattformen als Basis der Realisierung voraussetzt, aber gleichzeitig den Rahmen auch schafft, dass in einem Unternehmen die Richtung dessen, wie gearbeitet werden muss, einheitlich ist. Dass vorgegeben ist, wie Abteilungen untereinander zum Beispiel ihre Applikationen aufbauen, damit ein leichter Datenaustausch möglich ist. Und das ist, sozusagen, das umfassendere Paket und das ist dann halt eben nicht möglich, indem ich mich vor einen Bildschirm setze und in die Tastatur hacke, sondern es geht da darum, dass man, bevor man etwas anfängt, wirklich sich auch schon mal eine Idee davon macht, wer ist eigentlich mein Stakeholder? Wer ist jemand, der partizipiert an dem Programm? Wo sind die Touchpoints, um es ganz neudeutsch wieder auszudrücken? Wie sieht die Customer Journey aus? Oder halt, wenn man es anders

ausgedrückt haben möchte, wie sieht die Journey der Fachbereiche aus? Welche Daten werden verarbeitet? Das kann man alles in den No Code/ Low Code-Applikationen abbilden, aber die gedankliche Arbeit, die ist dann vorher zu leisten. Also sich einfach hinzusetzen und zu sagen: "Ich schreibe jetzt mal schnell ein Programm. Ich weiß zwar noch nicht, was ich eigentlich schreiben will." Ist wahrscheinlich genauso erfolgreich wie zu sagen: "Ich lauf jetzt einfach mal durch die Stadt. Weiß jetzt nicht genau, wohin ich eigentliche komme, aber ich habe dann alles gesehen, was wichtig ist." Nein, man muss leider in der Welt oft eben den anderen Weg gehen. Das heißt, man muss eine Primary Research durchführen, Secondary Research und lauter Preperation Research, um dann wirklich sagen zu können: "Das sind die Informationen, die ich brauche. Jetzt habe ich die richtigen Fragen gestellt und jetzt kann ich zur Lösungsfindung kommen." Das ist aber eher Beritt von Citizen Development.

- 12 [0:10:30.7] **I: RS** Du hast es jetzt ja schon teilweise angedeutet, dass da auch Nachteile mit einher gehen. Kannst du die noch mal ausführen, die Nachteile?
- 13 [0:10:39.5] **B: PF** Ja, die Nachteile bestehen halt einfach darin, dass ich Freiheiten habe. Das hört sich jetzt vielleicht ein bisschen seltsam an, aber die Freiheit, die ich habe, die beschränkt natürlich auch die Möglichkeit einer zielgerichteten Kommunikation. Es gibt in der Literatur einige Beispiele dafür, dass gerade die Schnittstellen, die jetzt innerhalb dieser Applikationen aufgebaut werden, im Nachhinein die wirklich komplizierten Fragestellungen sind. Also schnell, sagen wir mal, eine Routine aus einer Perspektive zuschreiben, das ist, glaube ich nicht besonders herausfordernd. Herausfordernd wird es dann, wenn ich eine Information habe und sage, diese Information geht jetzt auch auf diesen Empfänger. Oder jener Empfänger muss mir etwas geben. Das heißt, ich muss ein Datenmodell haben, in dem die Informationen ausgetauscht werden können, und zwar so, Aspekt passive Datensicherheit, dass die Daten korrekt gespeichert werden. Also das da auch keine Datenkorruption stattfindet und man mit den gespeicherten Daten nichts mehr anfangen kann. Das heißt, es muss etwas aus einem Guss sein und das stellt sich nicht heraus, indem ich das praktisch umsetze, sondern indem ich einfach erst mal hingehe und wirklich mir genau überlege, in welchem Umfang muss ich meine Daten aufbereiten. In welchem Umfang möchte ich diese Daten austauschen im Nachhinein und was soll mit den Daten passieren? Was soll aber auch mit der Pragmatik, das heißt, mit der Handlung, passieren, auf die ich die dann aufbaue und sage, in diese und jene Richtung soll dann halt auch in dem Workflow gehandelt werden. Das sind alles Dinge, die muss ich mir vorher überlegen und diese Freiheiten, die ich dann in der Gestaltung habe, die kann ich im Endeffekt nur beherrschen, indem ich wirklich sehr, sehr genau weiß, was ich eigentlich tun will. Also der große Vorteil, ich kann alles tun, ist halt auch im dialektischen Verhältnis gesehen, auch der große Nachteil. Ich kann halt auch alles falsch tun, ich kann halt auch alles so tun, dass es eben nicht zu einem einzigen Guss wird und darauf kommt es ja an.
- 14 [0:13:06.9] **I: RS** Mit diesen Vor- und Nachteilen, die du jetzt quasi gesagt hast. Könntest du kurz mal nach deiner Experteneinschätzung sagen, was jetzt aus heutiger Sicht, zum heutigen Tag, realistische Erwartungen sind, die man an eine No Code/ Low Code-Plattform stellen kann? Also ein Unternehmen an die stellen kann? Oder was nur utopisch gedacht ist, zum Beispiel. Also realistische Erwartungen, was heute schon möglich ist.
- 15 [0:13:39.2] **B: PF** Genau. Also die realistischen Erwartungen, die habe ich ja eben ganz kurz schon mal aufgezeichnet, dazu fällt auf jeden Fall die Abbildung eines Workflows im IT-Bereich auf der Basis und auf der Handlungsmöglichkeit eines Fachbereichs. Also die Mitarbeiter, die Mitarbeiterinnen eines Fachbereichs, können sich hinsetzen, können darüber nachdenken, was wollen wir eigentlich für ein Programm haben? Was soll das leisten? Wie wollen wir uns mit anderen vernetzen? Was wollen wir dann mit diesen Informationen machen? Welche Handlungen sollen ausgelöst werden? Das ist aktuell möglich, und zwar von dem Beginn der Produktion, über die Auslieferung, über den Kundenkontakt, bis hin zum Einzug

der Zahlungen durch die Geldinstitute. Schnittstellen zu anderen Programmen, die werden im Nachhinein im Unternehmen wirksam, im Sinne der Finanzbuchhaltung. Die sind dann halt auch möglich. Das heißt, im Prinzip kann ich große, komplexe Anwendungen, die im Moment auch schon im IT-Bereich programmiert worden sind, die kann ich in diesen No Code/ Low Code-Bereichen abbilden und zum Teil, das muss man ganz deutlich sagen, nicht nur schneller, sondern auch wegen der Standardisierung, also ich habe zwar die Freiheiten, ganz viel zu machen, aber ich habe nur ganz wenige, wenige ist vielleicht übertrieben gesagt, aber nur einige Möglichkeiten das zu realisieren, indem ich mich ja auf die Programmierung der No Code/ Low Code-Plattform verlassen muss und nur diese mir bereitgestellten Funktionen nutzen kann. Ähnlich kann man das ja bei Excel auch beobachten. Also ich kann nicht beliebig viele Funktionen haben, sondern die Excel-Programmierer bieten mir Funktionen an und die kann ich nutzen. Das ist reichlich viel. Die meisten Menschen werden von diesen Funktionsmöglichkeiten höchstens 10 bis 25 % nutzen und so ist es wahrscheinlich auch bei den No Code/ Low Code- Plattformen, dass viel mehr angeboten wird, als tatsächlich nachher genutzt wird. Aber deshalb können diese Funktionen, die genutzt werden, noch besser sein, weil eben keine individuellen Programmierfehler mehr auftreten, sondern alles das, was da erarbeitet worden ist, zum Beispiel die Anbindung an Geld- oder Kreditinstitute, die sind dann geprüft und nur dann, wenn die Qualität tatsächlich sich eingestellt hat, nach vielen, vielen Prüfungen in dieser Plattform, werden sie dann freigegeben. Und wenn Fehler auftreten innerhalb der Plattform, werden die dann auch schnell durch die Kunden gemeldet und können dann bei den Plattformherstellern korrigiert werden. Also es kann zum Beispiel sein, gerade in der Anbindung an Kreditinstitute, dass da die No Code/ Low Code-Funktionen sicherer, besser und praktikabler sind als eine Einzelprogrammierung und neben der Zahl der Spalten sollen sie eben auch eine höhere Datensicherheit bieten.

- 16 [0:16:43.1] **I: RS** Es gibt ja nun auch die Diskussion bei No Code/ Low Code darüber, dass das helfen kann, den Fachkräftemangel in der IT-Branche und generell in den IT-Abteilungen zu lindern, zu beheben. Teilst du diese Auffassung oder sagst du: "Nein, das ist definitiv keine Lösung dafür"?
- 17 [0:17:06.6] **B: PF** Also zwei Richtungen muss man hier bedenken. Einerseits aus dem Citizen Development, wäre ja die Fachbereichsentwicklung. Das heißt, wenn ich jetzt jemanden habe im Fachbereich, der sagt: "Mensch ich finde das super. Ich würde mich gerne in so eine Low Code/ No Code-Plattform einfuchsen." Wobei dieses einfuchsen eben nicht zwischen Suppe und Kartoffeln passieren kann, sondern da muss man tatsächlich einige Wochen, sozusagen Schulungen, vielleicht neben der Arbeit, aber halt doch Schulungen einplanen. Das ist ähnlich, wie wenn man Excel sehr, sehr intensiv nutzen will und zum Power-User werden möchte, ist das dann keine Sache, die man innerhalb von einem Tag lernt. Man kann sich innerhalb eines Tages annähern. Man kann auch schon die ersten Schritte gehen. Das auf jeden Fall, aber wenn man sagt, ich strebe an, eigentlich einen größeren Workflow abzubilden, dann braucht man schon einige Wochen Schulung und Erfahrung auch mit diesem Tool, auch wenn es dann schon eher in die Programmierung geht. Insofern, wenn ich dann jemanden aus dem Fachbereich habe, dann ist es ganz klar, dann habe ich oder benötige ich niemanden mehr aus der IT-Abteilung, der das programmiert. Insofern geht ja die Rechnung relativ leicht auf. Auch wenn dann die Möglichkeiten, den Fachkräftemangel in der IT zu beseitigen, dann nicht so ganz umfangreich ist. So, das ist eine. Man kann es, wenn auch jetzt nicht so zu hunderten, dann durch einen Fachbereich, die Leute dann ersetzt werden können. Aber man kann es. Auf der anderen Seite und das ist, glaube ich, der charmantere Teil der Sache. Ich sagte es ja gerade, durch das Anbieten von Funktionen, das Anbieten von in Alt-IT-Sprache ausgedrückt, von vorgefertigten Bibliotheken, also man muss sich dann auch die No Code/ Low Code-Plattformen als eine riesige Anreihung von Bibliotheken vorstellen und man zieht die über grafische Module, zieht man die dann in den aktiven Code hinein. Dadurch erspare ich natürlich halt auch den Programmierern viel Einzelarbeit. Also wenn ich

in der Lage bin zu sagen: "Ich hätte gerne eine Anbindung an ein Kreditinstitut und ich wähle durch das Anklicken der entsprechenden Kreditinstitute oder der Karten, die ich nutzen möchte, wähle ich alles aus und diese Funktion kann ich dann per Drag-and-drop in meinen Code hineinziehen oder in meinen Ablauf hineinziehen", dann ist das natürlich auch eine enorme Erleichterung für die Programmierer und dadurch kann man natürlich auch sagen, der Arbeitseinsatz in der IT kann dann darauf fokussiert werden, was wirklich durch diese Plattform nicht abgedeckt werden kann. Der kann darauf fokussiert werden, was tatsächlich an neuen Dingen im Unternehmen zu regeln und zu lösen ist. Und durch die Entlastung auf der einen Seite habe ich dann mehr Zeit, mich auf die anderen Dinge zu konzentrieren. Das könnte halt auch zu einer großen Erleichterung in diesem IT-Bereich führen. Und ja, das sind so die, die beiden Aspekte. Also No Code/ Low Code, auch wenn Citizen Development Fachbereichsentwicklung heißt, wird halt eben nicht nur in den Fachbereichen eingesetzt, sondern halt auch in den IT-Abteilungen und das ist meines Erachtens auch eine gute Sache.

- 18 [0:20:46.9] **I: RS** Alle nachfolgenden Fragen erfolgen unter dem Gedanken: Ein Unternehmen möchte eine No Code/ Low Code-Plattform bei sich einführen. Nur damit du weißt, das ist quasi immer die Ausgangssituation der Fragen. Wenn du jetzt ein Unternehmen hast, das gerne No Code/ Low Code einführen möchte. Was würdest du sagen, sind Voraussetzungen, die das Unternehmen mitbringen muss, um überhaupt erfolgreich No Code/ Low Code einführen zu können? Das sind so Fragen wie: Unternehmensgröße? Ist eine bestimmte Menge Kapital oder bestimmte Manpower vorhanden? Muss ein bestimmtes Mindset sein? Also alles, was in diese Richtung geht, gibt es da irgendwelche Voraussetzungen, wo du sagst: das ist essenziell, damit es erfolgreich eingeführt werden kann.
- 19 [0:21:37.1] **B: PF** Genau. Also ich nehme den letzten Punkt, den du genannt hast, mal raus. Mindset, weil Mindset das Allgemeine ist. Unternehmensgröße, Kapital was man hat, das ist dann halt doch immer relativ individuell und von Unternehmen zu Unternehmen unterschiedlich, aber was eben nicht so unterschiedlich sein kann und sein darf, ist das Mindset. Das Mindset ist ein kooperatives. Um eine Fachbereichsentwicklung anzustoßen, brauche ich natürlich die Möglichkeit, kooperativ mit meinen Kolleginnen und Kollegen im Fachbereich zusammenzuarbeiten. Das heißt, der Mindset muss so gestaltet sein, dass ich in der Lage bin, Probleme, die ich habe, die vielleicht unabhängig von dem fachlichen liegen, die eher im Ablauf im IT-Bereich liegen, zu adaptieren. Dass ich die übertragen kann in mein Arbeitsgebiet. Dass ich in der Lage bin, mit anderen diese Fragestellung zu lösen. Wenn ich aber in der Lage sein soll, diese Fragestellung zu lösen, als Mitarbeiter innerhalb einer Firma, dann brauche ich auch gewisse Freiheiten, die ich umsetzen kann. Das heißt, die Chefetagen müssen so vorgehen, wie Jobs das einmal in diesem häufig kolportierten Satz gesagt hat: "Wir stellen nicht gute Leute ein, damit wir ihnen sagen können, was sie tun sollen. Sondern wir stellen gute Leute ein, damit die uns sagen, was sie tun sollen." Das heißt, der Chef nimmt sich dann etwas zurück und geht dann in eine ganz starke, aktive Position des Zuhörens hinein und sagt: "Welche Lösung bietest du mir an? Was könnte denn passieren?" Das heißt, ich ermutige und ermächtige die Menschen, da etwas zu tun. Mindset in diesem Sinne fasst dann für mich eigentlich auch eine sehr, sehr gute Teamführung zusammen. Oder beinhaltet die auch. Und gerade da dann eher diese Begrifflichkeiten, die dann manchmal etwas belächelt werden, aber diese Psychological Safety ist eine ganz wichtige Dimension dabei, wenn ich ein Fachbereich-Mensch bin und schaue immer nur aus diesen fachlichen Aspekten auf die IT, dann stoße ich natürlich an Grenzen mit dem Verständnis für die IT. Und komm dann vielleicht mit Lösungsvorschlägen, die auf den ersten Augenblick, aus IT-Sicht, deppert sich anfühlen. Und diese Möglichkeit dann aber auch zu schaffen, in dieser psychologischen Sicherheit, ja man kann auch einen Vorschlag machen, der vielleicht wirklich abstrus ist. Der vielleicht gar nicht so einem erst mal so in den Sinn kommt, aber daraus ergeben sich sehr, sehr häufig Möglichkeiten, zu sagen: "Ok das kann man nicht so und so umsetzen, aber die Grundidee ist gut, dann könnte ich auch das daraus nehmen." Das heißt, man ist in

der Lage, ganz vieles in dieser Lösung zu berücksichtigen, was aus IT-Sicht erst mal gar nicht sichtbar war, aus fachlicher Seite sichtbar wird, in Bezug auf die IT und das kann man umsetzen. Und das kann man aber nur dann umsetzen, wenn ich nach der ersten, sozusagen, überraschenden Lösung, die jemand bringt aus dem Fachbereich, das Lachen gar nicht mehr aufhören will und dann zieht der sich zurück und damit hat sich dann auch die Möglichkeit einer kreativen, ich will fast sagen design thinking-Lösung. Also das Vorstoßen ins Unbekannte gehört da ein bisschen mit rein und das muss in einem Unternehmen dann auch möglich sein. Das ist eine ganz wichtige Sache. Das heißt, die Chefetage muss, sollte in der Lage sein, Lösungskompetenz bei den Mitarbeitern aktiv zu fördern, auf diese Mitarbeiter und Mitarbeiterinnen aktiv auch zuzugehen und die um diese Unterstützung zu bitten, sich Lösungen unterbreiten zu lassen. Das Zweite in diesem Mindset ist, klar, wir haben Freiheiten, die können halt auch dann leicht in Gegenden vorstoßen, in denen wir eigentlich gar nicht aktiv sein wollen. Man muss eine gewisse Governance aufbauen. Man muss sagen, so und so wollen wir unsere Programme haben. So und so wollen wir nach außen auftreten. So und so wollen wir aber auch untereinander, miteinander kommunizieren, so sollen die Oberflächen zum Beispiel gestaltet sein, wenn man das so genau machen möchte. Also die Governance muss einhegen, muss diese Freiheit wieder einhegen und muss den Mitarbeitern auch die Sicherheit geben: Ja, wenn ich dieses in diesem eingehegten Bereich tue, bin ich noch auf dem richtigen Pfad und so können wir dann auch im Nachhinein diese so geschafften Applikationen weiterentwickeln.

- 20 [0:26:32.5] **I: RS** Und welche Problemstellungen und Risiken siehst du dann bei der Einführung?
- 21 [0:26:38.0] **B: PF** Die Risiken sind die, dass man zu wenig fehlertolerant ist. Das heißt, wenn ich jetzt hingehe und probiere etwas aus und das funktioniert jetzt nicht sofort, dass man dann sagt: "Mensch, das ist jetzt wieder alles schiefgelaufen und das ist alles Mist." Also diese Einstellungssache, Mindset gehört da auch rein und dass man dann sagt: "Ok, wir haben hier etwas entdeckt, was jetzt noch nicht so ganz gut funktioniert. Macht es noch Sinn, in dieser Richtung uns weiterzubeben? Oder müssen wir was ganz anderes als Lösung versuchen?" Also diese Start-up-Mentalität, die muss dann rein. Will ich dieses, wenn man es noch mal in so einem neudeutschen Ausdruck bezeichnen möchte, will ich dieses erste Produkt, was wir erstellt haben, dieses minimal viable product, will man das jetzt fortentwickeln, kommen da gute Feedbacks zu oder sagt man: "Naja, das war der falsche Weg." Aber selbst wenn es der falsche Weg war, dann muss man als Unternehmen sagen: "War eine klasse Idee. Haben wir probiert. War jetzt nicht mit dem Erfolg, den wir uns erwartet haben. Wir schauen jetzt nach was Neuem." Aber man sollte da nicht hingehen und sagen: "Der Meyer hat das wieder falsch gemacht." oder "Das war wieder eine Meyer-Idee." oder was auch immer. Sondern dass man wirklich sagt: "Mensch, von Meyer ist eine klasse Lösungsidee gekommen. Die hat noch nicht so funktioniert, aber vielleicht beim nächsten Mal, dass wir da besser uns drauf vorbereiten können." Also diese Fehlerkultur, die muss ausgeprägt sein und die muss den Mitarbeitern auch die Möglichkeit geben, wenn sie ihre Freiheit genutzt haben und lösungskompetent gehandelt haben, dass sie dann auch wirklich, dass diese Anstrengung auch belohnt worden ist. Sonst frustriert man gerne und maßgeblich und nachhaltig und das ist dann auf jeden Fall zum Schaden für ein Unternehmen. Und ein nicht ganz erfolgreiches Projekt oder Programm haben wir in den Unternehmen auch in der IT-Ägide immer gehabt. Das wird es auch unter der No Code/ Low Code-Zeit geben. Und da muss man einfach als Unternehmen wissen: Ja, von den so und so viel Programmen, die wir machen, kommen vielleicht so und so viel Prozent nur durch. Aber das ist eine Einstellung, die man haben sollte und diese Fehler werden auf jeden Fall passieren.
- 22 [0:29:15.4] **I: RS** In dem Zusammenhang, wo du darüber gesprochen hast, dass man Mitarbeitern gegenüber gewisse Freiräume lassen muss. Wenn du bei der Einführung bist, welche

Organisationsform würdest du bevorzugen, bei der Einführung, als Projektmanager jetzt gesprochen?

- 23 [0:29:31.1] **B: PF** Genau, weil, ich kann ja da als Citizen Development, als Vorsitzender der Community of Practice, eben auf die Anstrengungen von PMI verweisen. Also bei PMI haben wir uns genau um diese Vorgehensweise bei der Einführung von No Code/ Low Code-Applikationen in den Fachbereichen Gedanken gemacht, da gibt es dann einen besonderen Vorgehensweg. Der ist als Vorschlag unterbreitet. Der ist in Buchform gefasst worden. Der ist auf den Webseiten zu sehen und der umfasst vier Bereiche. Das eine ist die Foundation, also dass man überhaupt erst mal eine Schulung machen soll, welche Bestandteile gibt es in diesem Citizen Development.
- 24 [0:30:24.5] **I: RS** Ich muss ganz kurz einhaken. Ich komme noch mal auf die Vorgehensweise an sich, komme ich bei einer späteren Frage zurück. Jetzt geht es mir tatsächlich explizit um die Organisationsform für das Projektteam. Also wenn du das einführest, würdest du die Fachmitarbeiter und das Projektteam aus ihren Fachabteilungen zum Beispiel rauslösen? Oder würdest du sagen, die sollten in ihren Fachabteilungen verbleiben, auch während des Einführungsprojektes? Wie würdest du das erst mal organisieren?
- 25 [0:30:56.2] **B: PF** Ja, da können wir vielleicht doch wieder auf den PMI zu sprechen kommen. Also herauslösen aus diesen Zusammenhängen, in denen die im Unternehmen sind, ist eigentlich dort nicht vorgesehen, sondern da finden zu ganz gewissen Zeiten Workshops statt. Es werden Ideation Workshops durchgeführt. Das heißt, was wollen wir überhaupt erreichen? Dann geht das dann an einen oder zwei Menschen innerhalb dieser Abteilung, die die sogenannte Citizen Developer- Ausbildung gemacht haben. Die dann sagen, ich setze das mal an der Tastatur mit diesem Low Code/ No Code-Programm um. Dann käme die nächste Station innerhalb dieses Workflows und dann würden die Leute wieder zusammengezogen. Aber das wäre jetzt keine tagesfüllende Beschäftigung, sondern das wäre tatsächlich etwas, was man innerhalb der üblichen Arbeitszeit erbringen kann. Vielleicht eben mit der Möglichkeit, die Kollegen dann in den Workshops zusammenzuziehen und dann auch entsprechende Ergebnisse zu haben. Wie man dann mit den Citizen Developern umgeht, muss man schauen. Also das ergibt sich halt auch aus dem Arbeitsumfang. Aber die normale Arbeit der Fachbereiche, die sollte eigentlich ungestört sein und ich kenne im Moment kaum ein Unternehmen, was sagen kann: "Ich habe so viele Mitarbeiter in meinen Fachbereichen, die leistungsstark sind, die ziehe ich ab und die bringe ich jetzt alle nur noch in die Programmentwicklung unter." Dann wäre nichts gewonnen.
- 26 [0:32:29.4] **I: RS** Du hast ja gerade schon darüber gesprochen, dass das PMI da ein eigenes Vorgehensmodell auch entwickelt hat. Könntest du kurz einordnen, wie dieses Modell aufgebaut ist? Also ist es ein traditionelles Wasserfallmodell oder eher ein agiler Ansatz oder eine Mischung aus beidem oder ist es etwas völlig Neues, was gar nicht zu den bisher bekannten Vorgehensmodellen passt?
- 27 [0:32:54.4] **B: PF** Nein, es ist im Endeffekt eher im agilen Bereich anzusiedeln. Es geht jetzt nicht darum, dass man einen starren Projektplan hat, sondern es geht wirklich darum, dass man in kurzen Abschnitten etwas schafft. Es geht darum, dass eine Idee aufgegriffen wird. Dass eine Idee relativ schnell auch gezeigt werden kann, durch jemanden, der sich dann mit diesem Low Code/ No Code- Programm auseinandersetzt, der sagen kann, so könnte die Oberfläche aussehen, so könnten die Schnittstellen aussehen. Das ist alles in diesem Bereich drin. Es geht halt auch in vielerlei Hinsicht in den Bereich der User-Stories hinein. Also auch agile Elemente, die ja eine große Rolle spielen, sind da drin. Und es hängt eher damit zusammen, dass man sagt: "Ich habe eine Idee, ich habe die Information durch die Fachabteilung, ich habe mich ausgetauscht, ich habe das nach ganz gewissen Prinzipien festgestellt." Also Workshops wären zum Beispiel eine ganz wichtige Sache. Aber es gehören auch von dem Citizen Developer ohne weiteres dann Fachinterviews dazu. Natürlich dann auf

einem sehr hohen fachlichen Niveau, weil, die kommen ja aus dem Fachbereich und der Fragende wie der Befragte kennen sich sehr, sehr gut aus darin. Und dass man dann sagen kann: In dieser und jener Form wollen wir es realisieren, dann wird auch relativ schnell gezeigt, dann werden erst Prototypen davon gemacht. Also das ist sehr, sehr stark agil und da wollen wir ein sehr, sehr schnelles Feedback.

- 28 [0:34:29.2] **I: RS** Sagst du, es ist ein reiner agiler Ansatz? Oder nimmst du Elemente aus, zum Beispiel, den traditionellen Modellen, sind die auch vorhanden? Bloß als Ergänzungsfrage.
- 29 [0:34:43.6] **B: PF** Ja, die Trennung selber fällt mir jetzt ein bisschen schwer, weil viele, viele agile Ansätze sind halt eben auch im Wasserfallmodell drin. Also sind wir da im linearen Projektmanagementbereich drin. Also, wenn du jetzt zum Beispiel hingehst und sagst, Lessons Learned. Ein Begriff aus dem Linearen und wenn ich jetzt Lessons Learned übersetze, mit Retro, was tatsächlich halt auch von der Frau Derby so bezeichnet worden ist, also einer der klassischen Retromasterinnen, wenn man das so sagen darf. Dann sind viele Elemente, die im linearen Ablauf sind, halt auch im agilen drin. Das Gleiche gilt zum Beispiel für Workshops. Also wir haben auch im linearen Bereich ganz viele Workshops durchgeführt, um herauszubekommen: Was machen wir? Was wollen wir tun? Und so weiter und sofort. Das ist da eigentlich drin. Das heißt, die Trennung zwischen agilen und linearen Abläufen, die würde ich da ziehen. Es gibt in diesem Sinne keinen Projektplan. Es gibt ein Ziel, was aber immer wieder adaptiert werden kann auf die neuen Erkenntnisse. Es gibt also diese Möglichkeiten, das zu tun, die sind relativ groß. Es gibt aber auch nicht ein, ich sage es einfach mal in diesen Worten, starren Product Backlog. Es gibt keinen starren Sprint Backlog. Also diese Begrifflichkeiten, die man aus diesem Umfeld des Agilen kennt und die ja eben nicht mit Beliebigkeit zu tun haben, sondern die sehr, sehr stark für 14 Tage oder 3 Wochen, 4 Wochen, die Arbeit eines Teams bestimmen. Der Sprint Backlog, diese starren Vorgaben, die sind da weniger gefragt. Also hier geht es eher darum, in einem laufenden Prozess sich immer wieder auszutauschen. Also es gibt viele Elemente wie Workshops oder halt auch Interviews, die wir im normalen linearen Projektmanagement haben. Den gibt es da drin und die werden notwendigerweise durchgeführt. Viele andere agile Bereiche, wie Sprint Backlog und sonstige Dinge, wie: "Dann muss das fertig sein." oder "Wie weit bist du?" oder Product Owner gibt es in dieser Form nicht. Sondern das ist der Fachbereich selber und da hat vielleicht einer den Hut auf und sagt: "Mensch, wenn wir das machen, dann würde ich das gerne umsetzen" Aber das käme eher aus dem Fachbereich und da gibt es keine großartigen Verbindungen zur Geschäftsführung außer den üblichen, die ein Fachbereich hat und dann wird das abgesprochen. Also das wäre dann in dem Sinne ein dritter Weg, aber eher in der Tendenz zum agilen, zum schnellen Reflektieren darüber, zum schnellen Review: "Wie waren wir?" "Haben wir das richtig gemacht?" "Wollen wir das so haben?" Aber ein Review, der jetzt nicht von außen kommt, vom Kunden, sondern hier ist ja der Erstellende der Kunde selber und der würde dann sich selber auf die Arbeit schauen.
- 30 [0:38:03.1] **I: RS** Und mit diesem Bezug zur Einführung einer Plattform an sich im Unternehmen, gibt es dann noch Phasen oder wie ist da das Vorgehen, wenn du einen agilen Weg wählst, bei der Einführung jetzt, dieser Plattform?
- 31 [0:38:18.7] **B: PF** Ja, natürlich gibt es da Phasen. Also diese Phasen, wird von dem Vorgehen der PMI, sind im Endeffekt ähnlich wie bei allen anderen, auch bei agilen Projekten. Es geht darum, dass man tatsächlich die Frage analysiert. Also es gibt dann diese entsprechenden Meetings, die man durchführt. Dann gibt es diese Zusammenfassung, diese Meetings werden moderiert, es wird zusammengefasst, dann wird gesagt: "In die und die Richtung könnten wir gehen", dann gibt es die nächste Phase. Das ist sozusagen der erste Prototyp, wo man sich dann wieder trifft, aber nicht nach vier Wochen, sondern vielleicht schon mal nach zwei Tagen. Bei manchen Meetings ist es auch so, die einen setzen sich dabei in die

Ecke und beantworten ihre E-Mails und trinken eine Tasse Kaffee dabei. Und danach gehen sie zurück und dann ist es in dem Meeting schon, dass der Citizen Developer dann sagt: "Ich hab mal was vorbereitet. Geht das in die richtige Richtung?" Also diese Phasen der preparation, der Betrachtung, die sind da. Dann die nächste Phase der Umsetzung, die ist jetzt aber nicht viel später, sondern die passiert relativ zeitnah zu dem, was man da erarbeitet hat. Man führt diese preparation-Phasen durch, stellt die richtigen Fragen, überlegt sich die richtigen Lösungen und dann geht man mit dieser Lösungsidee auf denjenigen zu, der sich mit diesem Tool, ich sagte es ja gerade, das muss man dann schon ein bisschen üben. Das ist ähnlich wie bei Power User von Excel, ergibt sich das nicht von alleine. Das heißt, derjenige, der setzt das dann um und zeigt es sofort am Bildschirm und dann kann das erste Feedback schon stattfinden: "Ah, das können wir anders noch machen!" So, und dann geht das von diesem Moment an weiter in diesen iterativen Schleifen, bis man dann sagt: "Ok, jetzt haben wir eigentlich diese Lösung, die wir ganz zum Anfang mal angedacht haben. Die haben wir jetzt in der und der Form realisiert." Und dann geht es in das Testing rein, also, das muss ja, so gerade, wenn man nach außen geht, zum Kunden, muss das ja absolut sicher sein, in das Testing rein und dann muss man natürlich, bevor man das überhaupt auf die Welt loslässt, also "Hello World", wie wir alle früher mal gelernt haben, da muss man dann natürlich auch die ITler da und die IT-Governance befragen. Kommen wir mit diesen Programmierungen, die wir da getan haben, kommen wir damit um die Ecke oder stoßen wir da an Sicherheitsgrenzen. Was müssen wir zusätzliche bedenken? Also das fängt schon so eine Low Code/ No code-Applikation ganz gut ab, aber die, wie gesagt, das ist eine allgemeine Vorgabe, eine allgemeine Funktionalität und die Prüfungen, die müssen tatsächlich dann auch durch die IT-Security erfolgen, also Spezialgebiet, und, wie wir jetzt durch die vielen aktuellen Dinge gelernt haben, wird das immer wichtiger, dass man diese IT-Sicherheit betrachtet. Ich habe gestern noch einen Bericht gesehen über diese Fake-Interviews, die die russischen Komödianten da gemacht haben. Also, es ist enorm, was da an Möglichkeiten mittlerweile geschaffen worden ist und guter Programmierer kommt vielleicht drauf. Ein guter Fachbereichsmensch, der kommt nicht drauf, der ahnt nicht, wie viele böse Dinge es in dieser Welt gibt und die sich alle auf sein Programm konzentrieren. Und da muss dann halt, sagen wir mal, ein Fachmann draufgucken und dann sagt der: geht oder geht nicht.

- 32 [0:42:09.4] **I: RS** Du hast ja jetzt ganz viel positive Aspekte von dem quasi hybriden Vorgehen, aber in einem agilen Schwerpunkt gesagt. Könntest du jetzt auch direkt mal konkrete Nachteile benennen, wo jemand, der das nutzt, darauf achten sollte, wenn er das macht?
- 33 [0:42:34.8] **B: PF** Genau. Also das nutzt, gehe ich jetzt mal davon aus, nicht die Applikation, sondern die No Code/ Low Code-Plattform.
- 34 [0:42:40.1] **I: RS** Nein, der dieses Modell, was du gerade genannt hast, einsetzen möchte. Auf welche Nachteile er achten müsste, wo man ein bisschen aufpassen muss?
- 35 [0:42:51.7] **B: PF** Genau. Die Nachteile sind in der Regel ähnlich wie bei der IT. Niemand, der ein Programm schreibt, liebt es zu dokumentieren. Also es werden Dokumentationen gemacht, aber ich habe bisher noch keinen getroffen, der außer Lippenbekenntnissen zur Dokumentation wirklich dann auch dafür brennt, dass was er tut, da gut dokumentieren. Dass das getan wird, das ist ein Ausdruck der Professionalität und das ist ganz gut. Das heißt, je besser ein Programm, was ich beschrieben habe, dokumentiert ist, desto einfacher ist es und jetzt kommt das Stichwort, das ganz wichtig ist: zu pflegen. Wenn ich ein Programm schlecht dokumentiere und ich bin dann weg, wo auch immer hin. Also gerne auch auf eine Pilgerfahrt, wie man das von einem berühmten deutschen Komödianten gelernt hat. Auf jeden Fall der Programmierer, der es gemacht hat, ist nicht mehr da und ein anderer soll jetzt dieses Programm überarbeiten. Der wird seine Schwierigkeit haben, sich da einzuarbeiten. Das ist bei Low Code/ No Code-Applikationen nicht anders, das heißt, die Dokumentation ist etwas, das getan werden muss und wo ein Augenmerk draufgelegt werden muss. Da kann ein



Risiko draus entstehen. Pflege führt sozusagen aber halt auch direkt zu einem nächsten Punkt. Eine Weiterentwicklung des Programms. Also ich will ja nicht nur Fehlerpflege machen, sondern ich will ja vielleicht auch ganz neue Applikationen dann hineinbringen, nach einem halben Jahr oder einem Jahr, oder wenn es ein Superprogramm war, vielleicht auch nach drei bis vier Jahren erst, aber ich will es. Auch da muss in dieser Applikation, in dieser Low Code/ No Code-Applikation, muss schon angelegt sein, da und da sind die Stellschrauben, um dieses oder jenes zu verändern. Und auch das muss gut vorbereitet sein. Dann die Möglichkeit, schnell die Sache nicht nur zu entwickeln, sondern es ist ja tatsächlich dann auch ein IT-Programm zu implementieren. Das heißt also, auch diese Implementierung, die muss dann abgestimmt vollzogen werden. Auch da muss man sich drüber Gedanken machen. Also, Releasemanagement lässt hier auch grüßen. Also, die Entstehung dieses Programms ist eine Sache, auch IT-Abteilungen können damit arbeiten. Die Dinge Pflege, Weiterentwicklung, Implementierung, diese ganzen Dinge, die sind halt auch immer mit zu bedenken, mit zu planen. Also die Probleme, die ich in der IT habe, wenn ich eine überlastete IT-Abteilung habe, die kaum dazu Zeit hat, zu dokumentieren, die habe ich dann halt auch unter No Code/ Low Code-Aspekten. Vielleicht ist es einfacher zu durchschauen, aber ich zumindest, aus meiner Erfahrung, weiß, wenn ich eine Exceldatei habe, das ist ja noch viel weniger kompliziert oder komplex als ein No Code/ Low Code-Programm und soll die dann weiterentwickeln und die dann pflegen und ich habe keine Dokumentation dazu und das sind dann vielleicht fünf oder sechs Exceldateien, die dann angeschlossen werden, mit so und so vielen Untertabellen. Das kann einen dann schon mal um den Schlaf bringen und kann ganz viele, viele Tage dauern, bis man das durchschaut hat, wenn man es überhaupt durchschaut. Und ähnlich ist es dann auch hier. Und wenn dann noch ein Satz dazu steht: "Diese Variable habe ich da und da." dann ist es leichter und hat also mit Programmierung gar nichts zu tun, sondern nur Dokumentation, hier greife ich die Information auf und werde sie da einbinden. Und das wäre eine Sache, die müsste mit bedacht werden.

- 36 [0:46:40.2] **I: RS** Bei Einführungsprojekten kommt es ja häufig auch dazu, dass entweder aus dem Projekt heraus oder von außen, also im Unternehmen, an das Projekt Änderungswünsche geäußert werden. Einfach weil sich Sachen ergeben haben, im Laufe der Zeit. Wie gehst du damit um? Aus deiner Erfahrung heraus, wie geht man am besten mit diesen Änderungswünschen um?
- 37 [0:47:07.2] **B: PF** Genau. Also, die Änderungswünsche, wir müssen jetzt mal einen Schritt zurückgehen und sagen: Ich bin die Fachabteilung und ich, Telekom ist ja nicht unbedingt ein kleiner Konzern, ich weiß das als Projektleiter, diese vermittelnde Rolle zwischen den Fachabteilungen und den IT-Abteilungen, aber halt auch den Betrieb. Also das weiß ich sehr, sehr gut aus der eigenen Erfahrung. Hier ist es jetzt so, ich habe diesen Weg. Die Fachabteilung findet dieses oder jenes nur nicht so ganz gelungen und sie hätte gerne folgendes. Oder folgender Fehler, das kann durchaus auch sein, ist in diesem IT-Programm aufgetreten, das sollte korrigiert werden. Das ging ja jetzt bisher in der Regel, wenn man es in den alten Strukturen sagt. Es ging über die entsprechenden Hefte, die man angelegt. Also man hat gesagt: "Das und das hätte ich jetzt gerne in der Umsetzung." Dann wurde ein Pflichtenheft daraus abgebildet und das Pflichtenheft wurde dann umgesetzt und das Pflichtenheft, das haben wir dann häufig bei der T-Systems Fachkonzept genannt und diese Umsetzung der Kundenwünsche in das Fachkonzept, oder in das Pflichtenheft, die haben häufig zu Informationsdefiziten geführt. Und da genau setzt jetzt No Code/ Low Code ein. Diese Informationsdefizite sollen jetzt vermieden werden. Derjenige, der an der Tastatur sitzt und dort aus einer großen Vielzahl von Angeboten, Bibliotheken und Funktionen grafisch dann sein Programm zusammenklicken kann. Wie gesagt, das ist keine Sache, die man innerhalb von 15 Minuten lernt, sondern da braucht man schon einige Wochen für, bis man das gut kann und in der entsprechenden Qualität auch machen kann. Und mit Aspekten des abstrakten Niveaus. Derjenige, der ist ja aus dem Fachbereich, der versteht eigentlich sehr genau, was da getan werden

muss. Und in diesem Zusammenhang ist der Informationsverlust vom Fachbereich bis zu demjenigen, der es umsetzt, eigentlich möglichst kurz. Vielleicht gibt es immer wieder Missverständnisse, auch innerhalb der Fachbereiche, wo der eine dies oder jenes unter einem gewissen Begriff versteht, und insofern haben sie dann einen Dissens. Das kann natürlich da auch auftreten, aber das gehört zur allgemeinen Kommunikation und nicht zu diesem Übertrag von einer anderen Anforderung eines Kunden in ein Pflichtenheft. Sondern hier sitzt der an der Tastatur und hat direkt die Möglichkeit, das zurückzuspiegeln an den Fachbereich und das ist der große Vorteil, der hier gegeben ist. Und da ergeben sich die Ersparnisse, die so No Code/ Low Code mit sich bringt. Ich hoffe, ich habe die Frage jetzt richtig beantwortet. Nicht ganz?

- 38 [0:50:32.7] **I: RS** Also du hast ja jetzt viel aus der Perspektive, wenn No Code/ Low Code schon in den Abteilungen bekannt ist, gesagt. Mir geht es wirklich mehr darum: Ich bin als Unternehmen dabei, das erste Mal eine Plattform einzuführen und dort gehen ja aus den Abteilungen, die noch nicht genau wissen, was ist jetzt No Code/ Low Code, kann ja trotzdem Wünsche, was für Änderungen jetzt noch vorgenommen werden sollen. Können ja an das Projekt herangetragen werden oder während des Entwicklungsprozesses bzw. Einführungsprozesses kann ja auch aus dem Projekt selbst Änderungswünsche für den Projektablauf oder für die Einführung kommen. Wie man am besten damit dann umgeht? Wie du empfehlen würdest, was sollte man das jetzt am besten in ein Backlog einpflegen? Also darauf hat diese Frage jetzt eigentlich abgezielt.
- 39 [0:51:24.8] **B: PF** Genau. Also, wenn man diesen Backlog dann so definiert, als einige Karten, die ich zum Beispiel in ein Kanbanboard einsetze, dann wäre das eine Möglichkeit, das wäre eine Ebene zu sagen: "Ich als User hätte gerne und dann so und so, kann man es umsetzen. Kannst du das machen?" Das ist richtig, aber das würde dann innerhalb des Projektes, innerhalb der Fachabteilung, wie ich es gerade beschrieben habe, laufen. Also da gäbe es dann ein Backlog, sozusagen, des Fachbereichs für den Fachbereich und der würde es dann umsetzen und würde es direkt zeigen können auch.
- 40 [0:52:01.4] **I: RS** Ok, also du siehst es mehr als eine dezentralisierte Sache. Es geht nicht alles ...
- 41 [0:52:06.7] **B: PF** Ja, ja, das ist so. Das ist der große Vorteil. Ich habe eben nicht mehr diese Übersetzungsschwierigkeiten, sondern kann sofort mich an die Arbeit machen und kann es vor allem Dingen auch sofort zeigen.
- 42 [0:52:18.0] **I: RS** Ok, dann was ich auch noch gerne wissen würde ist: Welche wichtigen Rollen siehst du für so ein Projekt? Also, was sind die wichtigen Stakeholder, wichtige Kompetenzen, die beim Projekt liegen müssen? Kannst du da mal näher drauf eingehen?
- 43 [0:52:35.4] **B: PF** Genau. Also nehmen wir mal an, ich greife jetzt mal ein Beispiel aus der Praxis heraus, wo ich nicht in diesem Projekt, weil die Telekom keine Fahrräder produziert, involviert war, aber was eigentlich in der Literatur immer relativ häufig berichtet wird. Es gibt also eine Fahrradmanufaktur, möchte ich fast sagen, aber die produzieren schon ganz ordentlich Fahrräder, und die haben sich dann entschlossen, diesen Weg der No Code/ Low Code-Applikation zu beschreiten. Und die haben dann tatsächlich gesagt, von dem Einkauf, über Lagerhaltung, Produktion, Qualitätssicherung, Verkauf, Kundenkontakt, Abrechnung wollen wir alles in einem Workflow darstellen. Und das haben die dann auch gemacht und wichtige Player sind natürlich dann in diesem Bereich die Anforderer, das sind die jeweiligen Fachbereiche. Dann gibt es innerhalb dieser Abläufe, gibt es dann unterschiedliche Rollen, zumindest PMI sieht die vor. Ich greife mal einen raus, das ist der Citizen Developer, der dann tatsächlich in der Lage ist, zu sagen: "So und so könnte diese Applikation umgesetzt werden." Dann gibt es aber auch den Citizen Developer Strategist. Der Strategist hat die Aufgabe, aus Unternehmenssicht heraus zu sagen: "In dieser und jener Form wollen wir diese

Applikationen aufbauen. Da und darauf sollen die ja einzahlen." Also, auch wenn die No Code/ Low Code-Applikationen durch den Fachbereich betrieben werden können, ist das keine App für 14,90 €, sondern das sind veritable Investitionen, die ein Unternehmen leisten muss, die sind teuer. Also das muss man ganz klar sagen. Die sind jetzt eine Ersparnis gegenüber Einzelentwicklungen, aber sie sind erst mal keine Ersparnis, wenn ich auf das Konto gucke. Da ist erst mal ein gewisser Geldbetrag, der wandert von der einer Hand zur anderen und die andere Hand ist halt nicht mehr die Hand des Unternehmens, die bezahlt, sondern das Geld ist weg. Und der Strategist, der sagt: "Das und das ist eigentlich das Einsatzziel, was wir dann haben." Dann gibt es den Moderator, der diesen Prozess eigentlich noch weiterentwickelt. Dann gibt es denjenigen innerhalb des Unternehmens, der sich für diese technischen Aspekte sehr stark macht. Also der ist schon so ein bisschen halb in dieser IT-Rolle drin, in diesem IT-Businessmanagement. Also der dann tatsächlich hinget und sagt: "Das und das muss man eigentlich an Hardware haben. So und so müsste eigentlich die Software beschaffen sein, die man darauf laufen lässt, um das zu machen." Also das sind so ganz wesentliche Rollen. Dann ganz wichtig, die Mitarbeiter im Fachbereich. Die sind diejenigen, die die Produktionsideen bringen. Die die Ideen bringen zur Lösung. Die die Fragen aufwerfen. Die die besten Betrachter sind. Dann diejenigen, die dann halt für die Kommunikation der unterschiedlichen Fachbereiche zuständig sind. Bei so einem Gewerk vom Einkauf über Lager, Produktion, Verkauf, Kunde, Abrechnung, da sind natürlich unterschiedliche Bereiche in einem Unternehmen betroffen. Wenn so eine Fahrradmanufaktur auch jetzt nicht über tausende Menschen verfügt, aber es können schon einige hundert sein, die da eingebunden sind. Und da ist dann auch immer eine Abstimmung notwendig. Und das wird dann zum Beispiel mit dem Moderator durchgeführt.

- 44 [0:56:26.0] **I: RS** Als Frage: Du hast ja gesagt der Ansatz, den ihr verwendet, ist hauptsächlich agil. Habt ihr dann auch solche Rollen wie Scrum Master und Project Owner? Ich meine Product Owner.
- 45 [0:56:38.7] **B: PF** Product Owner haben wir in diesem Sinne nicht. Einen Project Owner wäre ganz gut, wenn man den hat. Also das ist ja dann der Sponsor, der sagt: "Ja ich stehe dafür gerade, dass ihr zweimal in der Woche einen Workshop von vier Stunden machen könnt." Und das wird ja nicht über jahrzehntelang laufen, aber während der Entwicklungszeit wäre das zum Beispiel eine ganz gute Sache, dass man da zwei Vormittage, zwei Nachmittage einfach mal an dieser Sache arbeitet gemeinsam als Fachbereich, das wäre ganz gut. Product Owner, Scrum Master -> Nein. Also ist erst mal in dieser Form nicht vorgesehen. Dass Aspekte des Scrum Masters, wenn man den unter der Maßgabe des Enablers sieht, dass die von einzelnen in diesem Bereich wahrgenommen werden, das ist klar. Also das könnte zum Beispiel der Citizen Developer sein, der sagt: "Ich kann das hier versuchen, zu lösen." "Ich nehme das mit als Fragestellung usw., kann da Lösungsvorschläge machen." Das ist richtig, aber dass wir dann ein tägliches Daily haben, der ganze Fachbereich sich aufstellt. "Ich habe gestern folgendes gemacht..." "Ich mache heute dieses..." und "Das und das hindert mit." Das ist eigentlich da bei dieser Low Code/ No Code-Programmierung für den Fachbereich nicht vorgesehen. Also du musst immer wirklich dich darauf fokussieren: Wir sind hier im IT-Bereich. Wir machen eine IT-Applikation, aber wir sind nicht die IT. Wir haben eine Vorgabe und wir erarbeiten die innerhalb des Fachbereichs. Und innerhalb des Fachbereichs kommt auch das Feedback und innerhalb des Fachbereichs kommen auch die Lösungen. Dass die Abnahme dann durch dritte noch erfolgen sollte, ist eine ganz andere Sache, aber da sind wir festgezurrt und insofern die Kollegen und die Kolleginnen, die da im Fachbereich arbeiten, die werden gestern ganz andere Sachen auch gemacht haben, als sich nur um dieses Projekt zu kümmern. Das ist dann eher der Entwickler, der dann an der Tastatur sitzt und sich mit der grafischen Oberfläche der No Code/ Low Code-Applikation oder Plattform beschäftigt hat.
- 46 [0:59:00.1] **I: RS** Hast du dann einen klassischen Projektleiter im Kopf?

- 47 [0:59:05.4] **B: PF** Nein, das wäre dann hier der, sagen wir mal, der Moderator, der da vorgehen ist, von PMI. Also ein klassischer Projektleiter wäre dann tatsächlich jemand, der auch von, der da aus dem Projektbereich kommt und dann sagt: "Ich weiß, wie Workshops funktionieren. Ich weiß, wie dies und jenes funktioniert." Die die Fachbereiche ja dann halt auch nutzen und wo die dann sagen, der macht das dann. Das wäre eine Möglichkeit. Wie man den dann nennt, Moderator oder Scrum Master oder Projektleiter ist ja eigentlich unabhängig davon. Aber er hätte dann diese Aufgabe, diese Dinge mit vorzubereiten.
- 48 [0:59:51.5] **I: RS** Was ich auch noch gerne wissen würde, ist: Wenn du eine Low Code-Plattform einführst, kommst du ja irgendwann an den Punkt, dass du sie in das Unternehmen, in die Firma aufskalieren möchtest. Also, dass die Plattform von allen Abteilungen nach Möglichkeit auch eingesetzt wird, um die genannten Vorteile nutzen zu können. Wie gestaltest du diese Skalierung von dem Prototypen, von der Prototypenfachabteilung, hoch ins Unternehmen?
- 49 [1:00:28.5] **B: PF** Ja, ich sag mal, von der Einzelapplikation, Fachabteilung, das ist ja genau die Sache, die mit diesem Ideationworkshop, zumindest initial, angedacht werden soll. Das heißt, wer sind die Stakeholder? Wer sind die Beteiligten? Wo sind die Touchpoints? Wie ist die Customer Journey? Oder die Product Journey? Je nachdem, welche Perspektive man einnehmen kann. Dass es die Userstories dafür gibt, dass es die Personas dafür geben sollte, die man aufzeichnet. Das sind alles, ich will nicht sagen Banalitäten aus dem Bereich des Service Designs, aber es sind alles auf jeden Fall die Pflichtaufgaben, die man erfüllen muss. Und wenn ich dann erkenne, in der Anlage dieser Applikation auch schon im minimal viable product. Das soll eigentlich im Endeffekt da und dazu führen, verwendet werden, dass wir uns auch austauschen, mit dem Einkauf, mit dem Lager, mit der Produktion, dann sind diese Dinge zumindest vorzubereiten. Und diese Skalierung findet dann dadurch statt, dass man innerhalb des Entwurfs dieses Programms zumindest diese Schnittstellen mit bedenkt. Wenn jetzt dann, sagen wir mal, klar, eine andere Abteilung dazu kommt und sagt: "Wir haben es erst abgelehnt und wollten es gar nicht haben, aber jetzt sehen wir den Vorteil. Wir wollen da auch rein." Dann muss man halt schauen, wie man dann über einen entsprechenden Workshop, über den Austausch, den man dann noch moderieren kann, wie man dann von einer gegebenen Lösung zu einer anderen Lösung kommen kann. Und das ist wie in der jetzigen IT auch, also das sind dann Entwicklungsaufgaben für den Citizen Developer, der dann an der Tastatur sitzt und sagt: "Das und das kann ich umsetzen."
- 50 [1:02:11.7] **I: RS** Bei Skalierung geht es ja auch darum, die Mitarbeiter zu befähigen. Man führt ja so eine Plattform nicht ein und alle Mitarbeiter wissen sofort, wie funktioniert das, sondern die müssen ja erst mal auf diese Plattform befähigt werden. Das hast du ja auch selber auch vorhin gesagt. Wie gestaltest du diese Befähigung? Was sind da gute Tools da?
- 51 [1:02:33.6] **B: PF** Ja, es werden nicht alle Mitarbeiter auf die No Code/ Low Code-Plattform befähigt. Also das hieße, man würde einer Fachabteilung noch mal eine zusätzliche Aufgabe aufoktruieren. Das ist zu viel. Es muss ja auch nicht jeder in der Fachabteilung ein Excelspezialist sein. Dass wir alle rudimentär mit Excel umgehen können oder zumindest mit einer Tabellenkalkulation, um es neutral auszudrücken. Es gibt ja eins, zwei, drei und wie sie alle heißen. Das ist selbstverständlich, gerade auch in Bereichen, wenn du mit Finanzen zu tun hast, dann ist das, glaube ich, deren tägliches Handwerkszeug. Und die sind auch schon Experten, aber nicht alle in den Fachbereichen müssen das tun. Und so ist es auch hier. Das heißt, die Menschen müssen, wie bei den IT-Programmen auch, nicht befähigt werden, mit Kobold umzugehen oder mit C++ oder was auch immer. Sondern die müssen befähigt werden, mit der Applikation umzugehen, die hieraus erstellt worden ist. Das muss man tun über übliche Schulungen, das sind halt dann natürlich auch die Touchpoints, die man bedenken muss. Wo kommt eigentlich ein Stakeholder mit dieser Applikation in Berührung. Was muss er wissen? Wie muss er sich verhalten? Was müssen wir ihm auch bieten, anbieten an

Eingabemöglichkeiten? Und dass man das dann eh schnell ändern kann, ist halt einer der Vorteile von No Code/ Low Code, aber grundsätzlich, die müssen befähigt werden, mit der Applikation umzugehen und nicht mit der Plattform.

- 52 [1:04:12.0] **I: RS** Du sagst ja, das sind quasi klassische Schulungen. Siehst du das als klassischen Unterricht, wie man das im Studium erwartet, wo man einem Professor vor sich hat, der erklärt einem das? Oder würdest du sagen, da kann man auch noch andere Wege gehen? Und wenn andere Wege, welche würdest du sagen, sind da durchaus klug zu gehen?
- 53 [1:04:43.7] **B: PF** Genau. Jetzt nehme ich mal wieder diese Chef-Haltung ein, die ich eben angedeutet hatte. Wir stellen kluge Leute ein, damit sie uns sagen, was wir zu tun haben. Jetzt gehe ich genau diesen Weg mal. Das heißt, innerhalb der Anlage dieser Applikation kommen halt auch Fragestellungen auf. Wie kann ich das denn lernen? Wie kann ich mich denn an die Wege erinnern? Dazu müssten innerhalb des Teams die klugen Leuten, die ja für die Fachbereiche schon arbeiten, müssten Lösungen geschaffen werden. Ob man dann sagt: "Mensch, die Applikation, die ist jetzt schon so umfangreich geworden", das es ja sein kann. Vom Einkauf bis zur Abrechnung ist ja schon relativ viel, was man da alles bedenken muss. Das ist jetzt zu viel geworden, das muss jetzt richtig beschult werden, sonst verlieren wir die Anwender. Dann ist das halt eben so, dann muss eine richtige Schulung durchgeführt werden. Wenn aber gesagt wird: "Mensch, hier hilft jetzt einfach ein Sheet", dann geht das auch, dass man einfach so eine Art Spickzettel hat und sagt: "Ok, wenn ich das und das machen will, dann muss ich so und so vorgehen. Alles andere wird mir dann in der Applikation erklärt." Das kann man ja auch dann durchführen, dass man dann das sensitiv macht. Wo die Maus sich befindet, dass man Erklärungstexte einblendet. Das ist eine Sache, die dann durch das Team besprochen wird, das das dann auch realisiert.
- 54 [1:06:14.7] **I: RS** Ok, es ist ja auch wichtig sichtbar zu machen, ob ein Projekt erfolgreich war oder nicht. Wie sicherst du die Qualität eines Projektes bzw. wie misst du am Ende des Projektes den Erfolg? Was sind da Keypoints, wo du sagst: "Daran sehe ich, ich habe alles richtig gemacht und die Qualität hat gestimmt."
- 55 [1:06:40.7] **B: PF** Ja, also das ist der Punkt, ich habe ja, sozusagen während der Entwicklung, habe ich ja ganz viele Veränderungen. Ich habe eine Basis, auf der ich aufsetze, dann fange ich an zu entwickeln. Lerne dazu, ändere es wieder ab, baue das immer ein. Dadurch ist die Evaluation, das ist der Vergleich des Ergebnisses mit dem angestrebten Ziel, diese Evaluation ist dann zumindest nicht auf einem Status, der mal vorher in der Historie war, einfach durchzuführen. Weil eben moving Target, das verändert sich eben auch und auf einmal habe ich eine ganz andere Anforderung, in der sechsten/ siebten Woche, als ich zu Anfang hatte. Ist bei der IT-Abteilung üblicherweise nicht anders. Auch die machen ja viele, viele Dinge wieder neu. Der Vergleich, bin ich erst mal mit der Funktionalität zufrieden, die kommt dann, wie im Review im agilen Bereich, der kommt dann durch den Kundenkontakt. Das heißt: "The proof pudding is in the eating", also hier muss dann tatsächlich der Pudding gegessen werden und so ist es hier auch. Das heißt, es wird angewandt. Dadurch kommt erst mal heraus, habe ich das gut gemacht oder nicht. Die Qualität, das ist natürlich eine schwierige Frage. Qualität in welcher Hinsicht, Qualität in der Hinsicht auf die Schnelligkeit oder Qualität auf die nicht funktionale Ebene oder Qualität auf die funktionale Ebene. Bei der funktionalen Ebene kann ich relativ schnell sagen, ja, erfüllt oder erfüllt nicht. Bei den nicht funktionalen, da muss man halt Messungen durchführen, Metriken einführen, um zu sagen, ich komme jetzt endlich an die Zeiten ran. Ich weiß gar nicht, ob die No Code/ Low Code-Plattformen das mitbringen. Aber das ist dann eine Sache, die ist ein bisschen schwierig nachzuweisen. Meines Erachtens ist es so, hier haben wir Anwender in diesem Bereich, in den Fachbereichen, und die sind dann am ehesten in der Lage zu beurteilen, ob sie selber eine gute und sie befriedigende Arbeit geleistet haben oder nicht. Und wir brauchen auf jeden Fall, das hatte ich ja eben schon ausgeführt, jemand, der mit Verstand auf die Sicherheit

dieser Applikation schaut und dann einen Test, den man durchführt, denn ungetestete Software, egal aus welcher Feder sie stammt, auf einem Markt zu haben, geht nicht. Also es kann sein, dass man eine fehlerhafte Software am Markt hat und dass sie nachgearbeitet werden muss. Da muss dieser Fehler sozusagen akzeptiert worden sein, aber die Software selber muss immer getestet werden und sie muss ein Testat bekommen: "Ja, darf raus." Das dürfte aber nicht durch die Fachabteilung passieren, da müsste meines Erachtens, bevor sie dann aufschlägt im Außenuniversum, im Metaverse, müsste dann tatsächlich ein Test durchgeführt werden.

- 56 [1:09:58.1] **I: RS** Würdest du sagen, die Tester sollten ausserhalb oder innerhalb trotzdem des Unternehmens sein?
- 57 [1:10:05.1] **B: PF** Beides, also die Tester sollen sowohl in dem Fachbereich sein, ich grenze das jetzt mal auf den Fachbereich ab. Also die sollen ja genau gucken: "Haben wir das jetzt gut gemacht oder nicht?" Also sie sollen ja selber ihre Arbeit bewerten und ihre Arbeit da auch beurteilen. Und dann soll es natürlich innerhalb des Unternehmens soll es Leute geben, die in der Lage sind, diese Applikation zu testen. Wenn ich jetzt innerhalb der IT-Abteilung meines Unternehmens niemand mehr habe, der frei ist zum Testen, dann muss ich mir am Markt jemand suchen, der das testet. Aber das muss passieren, weil, das ist bei einer IT-Abteilung so. Das ist, wie wenn ich ein Auto baue. Also es kann auch sein, dass ich ein Auto per Hand baue und sage: "Ich lass es auf die Straße los." Aber getestet werden muss es, sonst darf es nicht raus.
- 58 [1:11:00.7] **I: RS** Wir nähern uns jetzt dem Ende des Interviews. So als meine letzte Frage: Hast du noch Anmerkungen? Hast du noch Best Practices to Share? Oder sagst du, ich möchte gerne noch etwas sagen, was jetzt nicht im Interview gefragt wurde, was ich aber als wichtig empfinde?
- 59 [1:11:18.7] **B: PF** Na, ich will einfach nur mal betonen, dass, ähnlich wie beim agilen Vorgehen, auch bei Citizen Development, der Fachbereichsentwicklung, das Mindset im Unternehmen ein ganz wichtiger Erfolgsfaktor ist. Also man kann, sollte wirklich den Mitarbeitern die Freiheit, aber auch das Vertrauen geben, Lösungen zu suchen, Lösungen auch umzusetzen. Sollte in der Lage sein, eine gute Teamstruktur zu schaffen. Sicherheit für alle Beteiligten. Was nicht heißt, ein Raum, in dem gepampert wird. Auch ein Raum, in dem Sicherheit herrscht, da können schon mal klare Worte gesprochen werden, das sollte auch so sein, aber eben halt nicht, dass die von oben herab gesprochen werden, sondern dass man klar in der sachlichen Ebene bleibt. Das sollte man versuchen alles umzusetzen, und wenn es eben geht, als Geschäftsführung, diese Leistungen, die dann on top erbracht werden, honorieren. Jetzt nicht unbedingt mit Geld, sondern halt mit den Möglichkeiten, die man sonst im Unternehmen hat. Auf jeden Fall Vertrauen und einer ganz klaren Akzeptanz gegenüber denjenigen, die das tun und diese Arbeit auf sich nehmen.
- 60 [1:12:41.8] **I: RS** Dann danke ich dir ganz sehr für deine Zeit und ich würde jetzt die Aufnahmen beenden.
- 61 [1:12:48.5] **Ende**

## Einwilligungserklärung Interview

Ich erkläre hiermit mein Einverständnis zur Nutzung der personenbezogenen Daten, die im Rahmen des folgenden Gesprächs erhoben wurden:

Datum des Interviews	08.07.2022
Namen der zu interviewenden Person	Peter Fey
Name der interviewenden Person	Rocco Schenk
Kurzbeschreibung des Forschungsprojekts	Einführung von No Code/ Low Code aus Sicht des Projektmanagements
Durchführende Hochschule	Hochschule Mittweida – University of Applied Science

Die Daten werden im Rahmen eines mündlichen Gesprächs erhoben, das mit einem Aufnahmegerät und Videoaufnahmegerät aufgezeichnet wurde. Zum Zwecke der Datenanalyse werden die mündlich erhobenen Daten verschriftlicht (Transkription), wobei die Daten nicht anonymisiert werden.

Vorhandene Kontaktdaten, der interviewten Person, werden aus Dokumentationsgründen in einem separaten Schriftstück lediglich den Gutachter\*innen der wissenschaftlichen Ausarbeitung zur Verfügung gestellt. Nach dem Abschluss des Projekts werden diese Daten gelöscht.

Der Speicherung der personenbezogenen Daten zu Dokumentationszwecken kann durch die interviewte Person jederzeit widersprochen werden. Die Teilnahme an dem Gespräch erfolgt freiwillig. Das Gespräch kann zu jedem Zeitpunkt abgebrochen werden. Das Einverständnis zur Aufzeichnung und Weiterverwendung der Daten kann jederzeit widerrufen werden.

Peter Fey  
Vorname und Name in Druckbuchstaben

Peter Fey  
Unterschrift

8.7.2022, Bonn  
Datum, Ort

## Anlagen 10, Fallzusammenfassungen

### EB01 Eckhard Hauenherm

- Sieht NCLC als Möglichkeit für Nicht-Programmierer, eigene Applikationen in einer Plattform umzusetzen.
- Citizen Development ist das Nutzen von NCLC durch Nicht-Programmierer.
- Sieht Vorteile in Lösung Kommunikationsproblem, eigene Prozesse besser umsetzen, verbesserte Geschwindigkeit.
- Sieht Nachteile in fraglicher Qualität der Applikation, die entwickelt wird, sowie Fragen zu Application Live Cycle, Betreuungssituation, Security, Governance.
- Realistische Erwartungen sind, dass IT unterstützt werden kann, nicht ersetzt.
- Problematik, dass in vielen Unternehmen NCLC von den IT-Abteilungen eingesetzt wird und nicht von den Fachabteilungen.
- Citizen Developer sollten logisch und prozessorientiert denken können.
- Als Voraussetzungen spielen Themen wie Größe und Finanzen untergeordnete Rolle.
- Mindset wird als wichtig angesehen und Vertrauen in MA muss da sein.
- Hürden bei der Einführung sind Governance-Fragen und technische Risiken (Datenverlust)
- Team sollte gemischt aus IT-MA und Non-IT MA bestehen.
- Projektorganisationsform Matrixorganisation, wo MA teilweise aus ihren Abteilungen gelöst werden.
- Bevorzugt agile Vorgehensweise bei Einführung.
- Bei hybridem Vorgehen wird nicht als Mehrwert empfunden.
- Vorteile bei agilem Vorgehen sind schnelles Feedback, schnelle Reaktionsgeschwindigkeit, geringere Vorinvestitionen
- Hürden bei agilem Vorgehen bei häufigen Wechseln von Teammitgliedern und motivierte Teammitglieder zu finden.
- Benutzt gerne aus dem agilen Bereich Scrum und Kanban.
- Änderungswünsche werden ins Product Backlog geschoben und dann dem Scrumverfahren folgend entsprechend bearbeitet.
- Als wichtige Rollen/ Mitglieder im Projekt wird der Product Owner, IT-MA, Business-MA, Scrummaster (unter Bedingungen eher Moderative Rolle),
- Wichtige Stakeholder sind im Projekt. Stakeholder sind Geschäftsführung, IT-Abteilung und IT-Leitung, Business MA.
- Alle Mitarbeiter und Geschäftsleitung informiert halten.
- Skalierung sollte schrittweise erfolgen und eventuell, wenn alles gut läuft in späteren Abteilungen parallelisiert werden.
- Befähigung der Mitarbeiter kommt über Initialschulung und weiterführenden Schulungen. Einsatz von Multiplikatoren ebenfalls sinnvoll.
- Qualitätssicherung durch Motivation der Teammitglieder.
- Erfolgsmessung durch Evaluation, ob Ziele erreicht wurden. Später dann KPI's bilden, wie viele Applikationen werden genutzt pro Gesamtzahl erstellter Applikationen.
- Besonders hervorgehoben wird, dass ein starkes Bewusstsein für ein Projektvorgehen vorhanden ist und eine NCLC-Plattform nicht nebenbei eingeführt.

### EB02 Robin Rosengrün

- Definiert NCLC dadurch, dass wenig Code benötigt und sich durch grafische Tools eine Anwendung zusammengesetzt wird.
- Setzt aber trotzdem voraus, dass fundierte IT-Kenntnisse vorhanden sein müssen.



- Sieht Citizen Developer als die Nutzer NCLC Plattformen, in der Rolle der Entwickler von Anwendungen.
- Als Vorteil wird gesehen die Geschwindigkeit, mit der Anwendung erstellt werden könne, geringerer Initialaufwand bei der Entwicklung kleinerer Fachanwendungen und Vermeidung von Kommunikationsverlusten.
- Als Problemstellung wird betrachtet, dass ohne entsprechende Governance eine Schatten IT 2.0 entstehen kann.
- Fehlende IT-Kenntnisse können sich negativ auf die Performance der entwickelten Anwendungen auswirken.
- Als Einsatzbereich wird die persönliche Produktivitätssteigerung der MA betrachtet und die Verbindung zu anderer Software im Unternehmen.
- Weiteres Einsatzgebiet ist die Verbesserung von Kommunikation zwischen Abteilungen.
- Was nicht Teil des Einsatzgebietes sein sollte, sind komplexe Anwendungen, mit großen Datenmengen, für die es andere Lösungen braucht.
- Eine Voraussetzung für die Einführung als wichtig erachtet wird eine funktionsfähige Governance, sowie ein Bewusstsein zu etablieren für Daten und wie mit diesen umgegangen werden muss.
- Als unwichtiger Faktor wird die Unternehmensgröße betrachtet und als unabhängig davon zu sehen.
- Mindset wird als ein wichtiger Faktor wahrgenommen und man sollte ein agiles Mindset besitzen in Bezug auf Citizen Development. Es sollten keine zu hohen Erwartungen am Anfang herrschen und den Citizen Developern der Raum gegeben werden, um sich zu entwickeln.
- Falsche Erwartungen werden auch als Problemstellung definiert.
- Als wichtiger Teil des Projektteams wird eine Person mit Erfahrung betrachtet. Dabei spielt es keine Rolle, ob diese von außen oder von innen kommt. -> Governance Administration
- Wichtig ist auch Enabler, der für die Community Implementierung benötigt wird.
- Projektgovernance Administration sollte in IT-Abteilung angesiedelt sein, um dadurch die Struktur aus der IT-Abteilung heraus zu entwickeln.
- Für das Vorgehensmodell wird aus Erfahrungswerten angegeben, dass eine gewisse vorausgehende Planung unterstützend wirken kann, aber nicht muss.
- Einschränkungen sollten dabei allerdings so gering wie möglich gehalten werden, um dem Projektteam den Raum zum Selbstentwickeln geben zu können.
- Bei einem Einführungsprojekt wird in der Weise betrachtet, dass es kein richtiges Ende gibt. Es werden immer neue Sachen hinzugefügt oder müssen weiterentwickelt werden, weshalb ein agiler Ansatz als optimal betrachtet wird.
- Phasen werden als notwendige Ergänzung während des Einführungsprojekts gesehen. Diese dienen dem Aufbau von Governance und dem Enabling. Diese Einführungsphasen sollten aber nicht länger als 1 Jahr gehen.
- Bei zu schneller Einführung kann sich das negativ auf die Prozessstabilität auswirken oder verschiedene wichtige Stakeholder werden zu wenig berücksichtigt.
- Aufgrund des Größe des Projekts sind keine klaren agilen Vorgehensweisen in Verwendung, sondern eher einzelne Tools wie Weeklys.
- Es wird versucht, Änderungen, wenn möglich, über ein Sprint System (Scrum) zu bearbeiten. Das klappt nicht immer, weshalb Änderungen unter anderem auch über ein Ticketsystem abgearbeitet werden von den einzelnen Teammitgliedern.
- Als wichtige Rolle im Projekt wird der Product Owner gesehen, damit jemand den Gesamtüberblick behält.
- Skalierung erfolgt nicht durch Manpower, sondern soll durch Automatisierung erfolgen. Vergabe von Lizenzen oder Enabling. Dies bedeutet zwar für den Anfang einen größeren Personalaufwand, aber mit steigender Skalierung verringert sich dieser erheblich.

- Für die Befähigung von MA wird angestrebt, dass diese sich das Wissen selbst aneignen durch Videokurse. Jedoch für Rückfragen soll man trotzdem zur Verfügung stehen, weshalb ein gewisses Wachstum auch Personal einkalkuliert muss.
- Als wichtiger Faktor außerhalb des Projektteams wird gesehen, dass das Thema NCLC in der IT-Strategie des Unternehmens vorhanden ist.
- Neue Arbeitsweisen wie die Microsoft Fusion Teams werden als erstrebenswert angesehen. Dort müssen die entsprechenden Abteilungen angesprochen werden, im Speziellen die IT-Abteilung.
- Weitere wichtige Stakeholder sind die Fachabteilungsvorsitzenden, da diese wichtig für die bereitgestellten personellen Ressourcen sind.
- Als wichtiges Kriterium zur Qualitätsmessung wird das Monitoring betrachtet, von zum Beispiel der Anzahl der aktiven Citizen Developer, Anzahl der aktiv genutzt Anwendungen.
- Die qualitativen Ziele werden in gleichem Maße als wichtige Faktoren für die Qualitätssicherung betrachtet.
- Als wichtiger Erfolgsfaktor wird gesehen, dass das Projektteam aus vielen verschiedenen Fachrichtungen kommt und dass es jedem Mitglied des Projektteams mitgetragen werden muss.
- Hervorgehoben wurde noch mal, dass NCLC-Plattformen kein Gegenmodell zur klassischen IT-Abteilung sind, sondern eine ergänzende Wirkung haben sollten.

### EB03 Fabian Gackstatter

- NCLC wird als Applikationsentwicklung definiert, für die es keine Voraussetzung ist, selbst Entwickler zu sein, welche Programme wie Visual Studio nutzen.
- Entwickler für NCLC eher Experten in Programmen wie PowerPoint oder Excel.
- Citizen Development kann potenziell jeder MA sein, wenn er mithilfe der richtigen Tools seine eigenen NCLC-Anwendungen erstellt.
- Als Vorteil wird betrachtet, die IT-Abteilung, als Bottleneck zu entlasten, in dem diese nicht mehr jeden kleinen Anwendungsfall selbst erstellen müssen.
- Probleme werden hingegen gesehen, wenn der Fall mangelnder Kontrolle eintritt. Citizen Developer erhalten durch NCLC die Fähigkeit, eigene Anwendungen zu entwickeln und auf Daten des Unternehmens zuzugreifen.
- Eine weitere Problemstellung könnte eine unvorbereitete Supportorganisation sein, welche mit den eingehenden Anfragen überfordert ist.
- Möglichkeiten, die durch NCLC realisiert werden können, sind, dass eine wesentlich höhere Anzahl an Ideen umgesetzt werden können, die im Zweifelsfall durch höher priorisierte Aufgaben im Backlog der IT-Abteilung nicht in die Entwicklungsphase gehen.
- Es wird klar abgegrenzt, dass ein Citizen Developer kein Ersatz für Softwarelösungen, wie SAP, erstellen wird.
- Ein weiterer Anwendungsfall ist die gemeinsame Entwicklung mit der IT-Abteilung für komplexere Lösungen, welche ein weitgehendes IT-Wissen benötigen.
- NCLC wird den Fachkräftemangel in der IT-Branche nicht beheben können, aber lindern. Das begründet sich darin, dass IT-Mitarbeiter nicht nur entwickeln, sondern auch weitere Rollen erfüllen.
- Als wichtige Voraussetzung wird ein passendes Mindset angesehen. Das bedeutet, dass man sich darauf einstellt, dass die Einführung weitreichende Folgen hat und diese getragen werden müssen. Das geht einher mit einem höheren Vertrauen gegenüber den Mitarbeitern, da diese die Anwendungen entwickeln sollen.
- Eine weitere wichtige Voraussetzung ist eine entsprechende Planung für das Vorgehen mit den Anwendungen und mit einhergehend ein Risikomanagement.
- Als wenig wichtig werden die Faktoren Manpower und Unternehmensgröße gesehen.
- Das Thema Finanzen wird als wichtig angesehen, allerdings nicht primär für das Einführungsprojekt, sondern für die nötigen Strukturen um das Projekt herum.

- Bei der Einführung werden drei Risiken als besonders wichtig betrachtet. Das sind mangelndes Personal und dass Unternehmen im Vorfeld wissen müssen, wie sie die Plattform einführen möchten. Das dritte Risiko ist eine zu große Bürokratie.
- Bei der Projektorganisation werden die meisten Projektbeteiligten auf Unternehmensseite nicht vollständig aus ihren Abteilungen gelöst, aber sie werden zu einem hohen Anteil ihrer Arbeitszeit angebonden. Einige werden allerdings auch zu 100 % im Projekt benötigt.
- Als bevorzugte Vorgehensweise wird ein agiles Vorgehen angegeben.
- Es wird kein Mehrwert in einem hybriden Modell gesehen.
- Als Vorteil wird betrachtet, dass sehr leicht unabhängig gearbeitet werden kann
- Problemstellungen, die entstehen können, sind, dass Unternehmen noch nicht in einer agilen Methodik gearbeitet haben.
- Innerhalb des agilen Vorgehens wird auf eine Abwandlung des Scrum-Modells gesetzt. Ergänzend wird auch Kanban eingesetzt.
- Rollen wie Scrum Master, Product Owner und Development Team werden als wichtig angesehen. Ergänzende Rollen sind Changemanagement Koordinator und Architekt.
- Für Änderungswünsche wird der klassische Scrum-Weg über das Product- und Sprint Backlog gewählt.
- Wichtige Stakeholder sind der CTO, Changemanagement Lead, Betriebsrat, Supportstrukturverantwortliche.
- Skalierung erfolgt über eine einheitliche Grundlage der Strukturen und Plattformenab-  
lement.
- Für die Befähigung von Mitarbeitern werden verschiedene Schulungs- und Workshopangebote als sinnvoll angesehen, sowie eine Community als Hilfestellung zwischen den MA. Multiplikatoren spielen in beiden Vorgehen eine Rolle.
- Qualitätssicherung erfolgt durch Monitoring verschiedenen qualitativer und quantitativer Daten.
- Für die Erfolgsmessung werden verschiedene KPI's gemessen und analysiert.
- Am Ende wurde noch einmal besonders hervorgehoben, dass ein Unternehmen wirklich wissen sollte, worauf es sich einlässt und ein hohes Maß an Commitment vorhanden sein sollte.

#### **EB04 Simone Göttl**

- NCLC wird als Bausteinsystem verstanden, bei dem ein Fachbereichsmitarbeiter auch ohne IT-Ausbildung sehr einfache Anwendungen erstellen kann.
- Citizen Developer wird als der Fachbereichsmitarbeiter aufgefasst, der die Anwendungen erstellt.
- Als Vorteil wird empfunden, dass es bei NCLC einfacher, kostengünstiger und schneller ist, Anwendung umzusetzen, als über den traditionellen Weg über die IT-Abteilung.
- Es wird die Auffassung vertreten, dass NCLC nur schwer traditionelle Systeme, wie SAP, ablösen kann.
- NCLC eignet sich, als Unterstützung für die persönliche Arbeit, aber auch fachbereichsspezifische Prozesse zu automatisieren.
- Als Hürde wird betrachtet, dass bestimmte Leitplanken geschaffen werden müssen, für Datenschutz, Berechtigungsmanagement und Governance im Allgemeinen.
- Als Befürchtung wird geäußert, dass falsche Erwartungen entstehen aufgrund von Werbeversprechen von Herstellern.
- Als Voraussetzung wird Größe des Unternehmens als untergeordnete Voraussetzung betrachtet.
- Als wichtige Voraussetzung wird das Mindset formuliert, einerseits von der Belegschaft, sowie auf der anderen Seite vom Unternehmen.
- Es wird alles förderlich empfunden, wenn NCLC ein Teil der Digitalisierungsstrategie ist.
- Während der Einführung wird als Problemstellung gesehen, dass verschiedene Sicherheits- und Governancefragen nicht geklärt sind.

- Es wird darauf hingewiesen, dass es eine große Auswahl an Plattformen gibt und die Auswahl gründlich erfolgen sollte.
- Es wird beobachtet, dass der Wasseranteil in Softwareprojekten zurückgeht, zugunsten von agilen Ansätzen.
- Für die Projektorganisation wird ein Zwiespalt gesehen zwischen agilem Vorgehen und den praktischen Erwägungen in Bezug auf die zeitliche Zuordnung von Fachbereichsmitarbeitern zum Projekt.
- Als ideale Vorgehensweise wird eine Mischform aus agilem und traditionellem Vorgehen gesehen.
- Als Bestandteile von traditionellen Ansätzen in der Mischform wird unter anderem Kennzahlenmessung wie Fortschrittsmessung angebracht.
- Bestandteile des agilen sind ein Scrum-Vorgehen und Kanban-Boards.
- Als Problemstellung bei diesem Vorgehen wird gesehen, dass eine Unerfahrenheit im agilen Bereich vorliegen kann.
- Als gute Methodik, um mit Änderungen umzugehen, wird das Backlog empfunden und diese in Sprints zu bearbeiten.
- Als wichtige Rollen neben Scrum Master und Product Owner werden auch Schnittstellen zur Compliance-Abteilung und Betriebsrat gesehen. Diese sind auch wichtige Stakeholder, falls nicht Teil des Teams.
- Als wichtige Stakeholder werden generell alle Personen im Unternehmen definiert, die mit der Plattform zu tun haben.
- Eine Idee für eine mögliche Skalierung wird in Multiplikatoren, Communitys und Intranet gesehen.
- Für die Befähigung werden Multiplikatoren, Communitys und Schulungen als wichtig angesehen.
- Als nützliche Messerwerte für die Erfolgsmessung werden die klassischen Kennzahlen herangezogen. Es können aber auch noch weitere KPI's gebildet werden.
- Als wichtig wird die Plattformauswahl noch einmal erwähnt.

#### **EB05 Christian Rohrbeck**

- NCLC wird gesehen, als ein System, mit optischen Mitteln Algorithmen abzubilden.
- Citizen Development wird als Möglichkeit gesehen, MA die keine Kenntnisse von Programmierung haben, zu befähigen, einfache Algorithmen selbst zu erstellen.
- Als Vorteil wird gesehen, dass fachliche Anforderungen ohne Reibungsverluste in IT-Lösungen umgesetzt werden können.
- Als Nachteil wird betrachtet, dass bei MA häufig ein grundsätzliches Verständnis für IT-Systeme fehlt. Dadurch kommt es zu erhöhten Zeitaufwänden.
- Zum Thema Erwartungshaltung wird angemerkt, dass die Erwartung nicht zu hoch sein sollte.
- Es wird angenommen, dass die Einführung nicht zur Entlastung von IT-Abteilungen beiträgt, sondern der Aufwand gleichbleibt.
- Als ein voraussetzender Faktor für Unternehmen bei der Einführung wird eine gute Firmenkultur gesehen.
- Eine weitere Voraussetzung ist ein entsprechendes Mindset der MA.
- Als eine Problemstellung wird hervorgehoben, dass das Commitment zum Projekt, in einzelnen Abteilungen fehlt.
- Finanzielle Mittel sollten in der benötigten Menge bereitgestellt und eingeplant werden.
- Ein weiterer Faktor ist ein Projektmitglied, welches Erfahrung mit NCLC hat und die Projektmitglieder mitnimmt.
- Die Größe des Unternehmens spielt keine Rolle.
- Problemstellungen, die auftauchen können, sind ein fehlendes Verständnis der eigenen Prozesse im Unternehmen.
- Als optimale Organisationsform wird eine Form betrachtet, in der die Mitarbeiter zu 100 % dem Projekt zugeordnet werden, dies wird aber als nicht realistisch betrachtet.

- Als weitere Organisationsform wird die Möglichkeit gesehen, die Mitarbeiter zu einem gewissen Teil dem Projekt zuzuordnen.
- Als bevorzugtes Vorgehensmodell wird ein hybrider Ansatz betrachtet, mit dem Schwerpunkt auf dem traditionellen Ansatz.
- Der agile Ansatz wird als Möglichkeit für Unternehmen gesehen, die bereits weitreichende Erfahrung mit diesem Vorgehen haben.
- Als Vorteil des agilen Anteils wird die Selbstbestimmtheit und das schnelle Feedback gesehen.
- Als Vorteil des traditionellen Ansatzes wird die Planungssicherheit für Budget, Scope und Time gesehen.
- Nachteilig bei einem hybriden Modell ist, dass in der entsprechenden Gewichtung sich auch die Nachteile der zugrunde liegenden Vorgehensmodelle teilweise auswirken.
- Veränderungen werden nach Impact auf das Projekt klassifiziert.
- Große Veränderungen werden eingeplant, was Auswirkungen auf den Zeitplan haben kann.
- Kleine Änderungen werden in den Backlog gepackt und nach Priorität abgearbeitet.
- Als wichtige Rollen werden der Sponsor/ Product Owner und der Projektleiter angeführt.
- Scrum Master wird eher als weniger wichtig betrachtet, um zu 100 % der Rolle zugeordnet zu werden.
- Wichtige Stakeholder sind die Fachexperten aus technischer Sicht.
- Weitere wichtige Stakeholder sind die Verantwortlichen in den mit dem Projekt korrelierenden Abteilungen.
- Für Skalierung wird es als vorteilhaft angesehen, die Mitarbeiter, die später mit NCLC arbeiten, früh abzuholen und von einer Pilotenabteilung aus zu skalieren.
- Es wird darauf hingewiesen, dass Schwierigkeiten entstehen können, wenn mit einem neuen System gearbeitet werden muss.
- Befähigung sollte zyklisch erfolgen.
- Zur Qualitätssicherung werden messbare Größen herangezogen, wie mit in Time, in Scope und in Budget.
- Die gleichen Faktoren werden auch für Erfolgsmessung verwendet, plus qualitative Daten.
- Am Ende wird darauf hingewiesen, dass bei so einem Projekt die emotionale Seite besonders wichtig ist und darauf sollte geachtet werden.

#### EB06 Moritz Hesse

- NCLC wird definiert als Möglichkeit, mithilfe von interaktiven Oberflächen Applikationen zusammensetzen, ohne Quellcode selber schreiben zu müssen.
- Citizen Development beschreibt die Entwicklung der Applikationen in Fachabteilungen durch die Fachmitarbeiter.
- Als Vorteil wird betrachtet, dass eine langjährige Entwicklung durch die IT-Abteilung und Kommunikationsverluste in den Anforderungen vermieden werden.
- Weitere Vorteile, die genannt werden, sind eine Erhöhung der Produktivität und andere betriebliche Effekte.
- Als nachteilig wird betrachtet, dass für solche Applikationen eine gewisse Struktur beim Erstellen durch die Citizen Developer mitgebracht werden muss.
- Weiter wird gesagt, dass dies in das Feld Governance mit hineinreicht.
- Als Anwendungsszenario wird genannt, dass viele betriebliche Auxiliar- und Hilfsprozesse abgebildet werden können.
- Als Voraussetzung wird betrachtet, dass die technischen Systemvoraussetzungen da sein müssen.
- Begünstigend kann eine entsprechende Vorbildung der MA sein.
- Unternehmensgröße wird als irrelevant eingestuft, solange die entsprechenden personellen Ressourcen vorhanden sind.

- Mindset wird hervorgehoben als stark begünstigender Faktor.
- Problemstellungen können sein fehlende Governance.
- Die Organisationsform hängt hauptsächlich vom Anwendungsfall der Plattform ab.
- Als bevorzugtes Vorgehensmodell wird ein agiler Ansatz genannt, da Wasserfall zu starr ist.
- Vorteilhaft aus dem Wasserfall wird nur gesehen, die bessere Strukturierung des Projekts.
- Als nachteilig bei einem agilen Vorgehen wird gesehen, dass eine mangelnde Übersicht entstehen kann.
- Aus den agilen Methoden wird das Scrum-Model und das Kanban-Model bevorzugt verwendet. Nicht zwingend zusammen.
- Mit Änderungen wird im Scrum Rahmen so verfahren, dass diese in das Backlog übernommen und in Sprints abgearbeitet werden, je nach Priorität.
- Als wichtige Rolle im Projekt werden gesehen der Citizen Developer (optional), der PMO, Projektleiter sowie technische Rollen.
- Weitere Stakeholder im Projekt sind die Geschäftsleitung, IT-Verantwortlicher und ein Citizen Development-Beauftragter.
- Als Befähigungsmaßnahmen für die MA werden Schulungsmaßnahmen und Community-Bildung definiert.
- Skalierung erfolgt aus einer Pilotabteilung heraus in verschiedene andere Abteilungen. Dabei wird mit Multiplikatoren gearbeitet.
- Erfolgsmessung wird realisiert über technische Funktionalität des Systems sowie qualitative Messwerte, wie Kundenzufriedenheit.

#### EB07 Tino Janke

- NCLC wird als Ansatz zur Entwicklung von Software definiert, bei dem der Anwender keine Programmiersprachen beherrschen muss.
- Citizen Development wird als Ansatz gesehen, damit nicht Entwickler in die Lage versetzt werden, Applikationen zu erstellen.
- Vorteilhaft wird dabei gesehen, dass es den Aufgabenbacklog der IT entlastet.
- Ein weiterer Vorteil, der genannt wird, ist, dass die Entwicklung von Lösungen schneller geht.
- Als Nachteil wird gesehen, dass ein erhöhter Aufwand für Governance und Security betrieben werden muss.
- Zur vollen Potenzialentwicklung sollte NCLC organisationsweit ausgerollt werden.
- NCLC kann Innovationen fördern, bei entsprechender Verankerung im Unternehmen.
- Es wird darauf eingegangen, dass auch schnelles Reagieren auf geänderte Rahmenbedingungen möglich ist.
- NCLC dient der Auflösung von Flaschenhälsen.
- Als Voraussetzung wird gesehen, dass ein Umdenken im Unternehmen stattfinden muss.
- Der Umgang mit Mitarbeitern muss sich verändern.
- Unternehmensgröße und finanzielle Aspekte werden als zweitrangig betrachtet.
- Als Problemstellung wird definiert, dass das Thema unterschätzt wird.
- Wird als reines IT-Thema gesehen, was es nicht ist.
- Governance wird ebenfalls als eine potenzielle Hürde definiert.
- Ein falsches Bewusstsein für das Thema wird ebenfalls als problematisch eingeordnet.
- Als wichtig für das Projekt wird ein entsprechendes Commitment der Stakeholder zum Projekt gesehen.
- Vorteilhaft ist, wenn die Mitglieder des Projektteams zu Beginn in ihren Fachabteilungen verbleiben, aber im Laufe der Zeit teilweise voll für das Projekt zur Verfügung stehen.
- Für die Wahl des Vorgehensmodells ist ein entscheidender Faktor, welches Vorgehen das Unternehmen selbst kennt.

- Bevorzugt wird aber aus Sicht des Experten ein hybrider Ansatz.
- Die Gewichtung des hybriden Modells würde je nach Unternehmen angepasst.
- Als Tools des traditionellen Vorgehens werden Phasen hervorgehoben.
- Bei agilen Methodiken wird in der Umsetzung auf Sprints verwiesen.
- Für die Skalierung in andere Bereiche wird auf den iterativen Charakter des agilen verwiesen, weshalb dies dort zu bevorzugen wäre.
- Änderungswünsche werden mit der Zielstellung abgeglichen und werden dann in der Umsetzung eingeplant.
- Eine wichtige Rolle im Projekt ist jemand, der das Thema in der Organisation aus eigener Motivation vorantreibt.
- Wichtig sind auch motivierte Projektmitglieder, die an der Umsetzung beteiligt sind.
- Sponsor im Projektteam wird als optional angesehen, aber als wichtig für das Projekt.
- Der Sponsor muss in jedem Fall ein offenes Mindset mitbringen.
- Für die Skalierung sollten Erfolge schnell im Unternehmen gezeigt werden.
- Bei der Skalierung darf die Governance nicht vergessen werden.
- Wichtiger Bestandteil der Skalierung ist auch das Bilden von Communitys
- Skalierung sollte von Abteilung zu Abteilung erfolgen, in Iterationen.
- Retrospektiven werden für die Erfolgsmessung als vorteilhaft empfunden.
- Bildung von vergleichenden KPI's wird ebenso genannt.
- Hervorgehoben wird, dass das ganze Projekt einen Mehrwert schafft, für die Mitarbeiter und Kunden.
- Es wird als wichtig betrachtet, dass das Thema Governance nicht vernachlässigt wird.

#### EB08 Peter Fey

- NCLC ist die Möglichkeit, über eine grafische Benutzeroberfläche zu programmieren.
- Es ist nicht wichtig, dabei Codezeilen schreiben.
- Citizen Development ist die Vorgehensweise, um die Fachbereiche dazu zu ermächtigen, NCLC einzusetzen.
- Wichtige Faktoren, die beachtet werden sollten, sind Teammotivation, Datensicherheit und Governance.
- Vorteilhaft wird gesehen, dass die Anwendungen, die durch NCLC erstellt werden, ohne Kommunikationsverluste auf die Bedürfnisse der Fachabteilung angepasst werden können.
- Der Rahmen für die Entwicklung der Applikationen wird durch die Plattform vorgegeben, was eine Standardisierung ermöglicht.
- Als Problemstellung wird gesehen, dass ohne Rahmenbedingungen die Applikationen sehr viel Freiheiten ermöglichen, was zu Probleme führen kann.
- Ein besonderes Augenmerk sollte auf die Datensicherheit gelegt werden.
- Mithilfe von NCLC-Applikationen können Workflows im IT-Bereich auf Basis der Handlungsmöglichkeiten des Fachbereichs entwickelt werden.
- Es wird ermöglicht, komplexe Anwendungen zu entwickeln, die auch für den Außengebrauch einsetzbar sind.
- Es wird darauf hingewiesen, dass die Anwendungen schneller entwickelt werden und trotzdem innerhalb eines Standards sind.
- Das Einarbeiten in die Plattform muss ausführlich erfolgen und kann nicht nur so nebenbei geschehen.
- In der IT-Abteilung kann es durch NCLC zu einer Entlastung führen.
- Es kommen allerdings auch neue Aufgaben für die IT-Abteilung durch eine Einführung hinzu.
- Das Mindset sollte im Unternehmen kooperativ sein.
- Es wird als wichtig erachtet, dass für Problemlösungen eine gewisse Freiheit bei der Entwicklung herrscht.
- Geschäftsleitung muss den Mitarbeitern mehr vertrauen und weniger Vorgaben machen.

- Offenheit für Lösungsvorschläge von MA, ohne diese direkt abzulehnen.
- Governance wird als wichtige Voraussetzung gesehen, die aufgebaut werden muss.
- Ein Risiko ist, dass es eine zu geringe Fehlertoleranz gibt.
- Gute Feedback-Kultur wird als wichtig angesehen, um eine ständige Weiterentwicklung der Applikationen, die entwickelt werden sollen, zu ermöglichen.
- MA im Projekt sollten nicht aus ihren Fachabteilungen herausgelöst werden.
- Regelmäßige Projektteam-Zusammenkünfte werden angestrebt und dann wird zurück in die originäre Abteilung gegangen.
- Es wird auch als in der Praxis schwierig angesehen, Mitarbeiter dauerhaft aus ihren Abteilungen herauszulösen.
- Bevorzugtes Vorgehensmodell hat einen agilen Schwerpunkt, es werden aber auch Bestandteile des traditionellen Vorgehens beigemischt.
- Es wird die Frage aufgeworfen, ob sich einige Methoden der Vorgehensmodelle wirklich grundlegend unterscheiden.
- Beim agilen Vorgehen wird auf Feedbackschleifen gesetzt.
- Sprint Backlog wird als teilweise zu starr wahrgenommen.
- Product Owner gibt es nicht, die Fachbereiche selber erfüllen diese Rolle.
- Das Vorgehen ist eher ein hybrides, aber mit Schwerpunkt auf agilem Vorgehen.
- Phasen finden Anwendung im hybriden Vorgehen.
- Es sollte aufgepasst werden bei der Frage der Dokumentation.
- Berücksichtigt und geplant werden muss, dass die Plattform sich ständig weiterentwickelt und dafür Maßnahmen ergriffen werden.
- Änderungen werden nicht über den starren, langen Weg des Changemanagements durchgeführt.
- Bei Änderungswünschen sollen diese in ein Backlog eingetragen werden. Dieser Backlog soll allerdings in der Fachabteilung liegen, nicht zentralisiert im Projektteam.
- Die Fachabteilungen selbst werden als wichtige Stakeholder gesehen.
- Citizen Developer werden als wichtig angesehen, da diese später die Anwendung einsetzen.
- Es wird die Rolle des Citizen Developer Strategist aufgeworfen
- Als vorteilhaft wird auch ein Kommunikationsbeauftragter betrachtet.
- Sponsor anstatt eines Product Owners als Rolle verwendet.
- Scrum Master findet keine Verwendung, es wird eher ein Enabler genutzt.
- Es wird auch kein klassischer Projektleiter verwendet, sondern ein Moderator.
- Skalierung erfolgt über das initiale Produkt hin zur Fachabteilung. Das gewählte Tool sind Ideation Workshops.
- Abteilungen werden schrittweise hinzugenommen.
- Es werden nicht alle Mitarbeiter auf NCLC befähigt, sondern nur bestimmte in den Fachabteilungen.
- Befähigung wird über Schulungen realisiert.
- Zur Qualitätssicherung wird ein ständiger Soll-Ist-Vergleich zwischen aktuellem Zustand und Zielstellungen durchgeführt.
- Qualitative Faktoren wie Zufriedenheit spielen dort ebenfalls eine Rolle.
- Es wird unterschieden dabei zwischen funktionaler Ebene und nicht funktionaler Ebene.
- In funktionale Ebenen lassen sich Metriken einführen und Messungen durchführen.
- Eine weitere Möglichkeit der Qualitätssicherung soll über Tester geschehen.
- Es wird hervorgehoben, wie wichtig das Mindset in den Bereichen agiles Vorgehen, Citizen Development und Fachbereichsentwicklung ist.



# Anlagen 11, Codebuch

## Codesystem

1 - Vorbereitung Unternehmen	0
1.1 - Auswahl Kriterien Devisen	9
1.2 - Motivation Stakeholder	9
1.3 - Projektbewusstsein	3
1.4 - Vorbereitung der Organisation im Unternehmen	10
1.5 - Umgang mit Mitarbeiter der Unternehmenskultur	5
1.6 - Vorteile NCLC Plattformen	0
1.6.1 - Lösung Kommunikationsprobleme	7
1.6.2 - Prozessverbesserung	12
1.6.3 - Verbesserung der Arbeitsabläufe MA	9
1.6.4 - Steigerung der Qualität von Lösungen	3
1.6.5 - Höhere Entwicklungsgeschwindigkeit	5
1.7 - Nachteile NCLC Plattformen	0
1.7.1 - schlechte Qualität der Anwendung	2
1.7.2 - Falsche Erwartungen	8
1.8 - Governance Einfluss	0
1.8.1 - Fehlende Governance	8
1.8.2 - Governancestrategie	11
1.9 - IT Abt. und NCLC	0
1.9.1 - Negative Auswirkungen für IT	3
1.9.2 - Entlastung IT Struktur	5
1.9.3 - Fachkräftemangel IT	2
2 - EP: Eigenschaften Unternehmen	0
2.1 - Mindset	9
2.1.1 - Fehlerkultur	2
2.2 - Kapital	5
2.3 - Unternehmensgröße	9
2.4 - Technische Planung	4
2.5 - Erfahrenes Projektmanagement	2
3 - EP: Risiken und Hürden vor der Einführung	0
3.1 - Schatten IT	4
3.2 - Cybersecurity und Datenschutz	3
3.3 - Thema unterschätzt	9
3.4 - Weitere Risiken	4

4 - EP: Projektorganisationsform	9
5 - Wah Vorgehensmode	0
5.1 - Ag es Vorgehensmode	1
5.1.1 - Option auf Wasserfall Elemente	3
5.1.2 - Begründung ag es Vorgehen	7
5.2 - Hybr des Vorgehensmode	0
5.2.1 - Ausprägung Hybr des Mode	5
5.2.2 - Begründung hybr des Vorgehen	4
6 - Vorteile gewähltes Vorgehensmode	0
6.1 - Vorteile ag es Vorgehen	3
6.1.1 - Vorteile ag schnelle Feedback	6
6.1.2 - Vorteile ag Reaktionsgeschwindigkeit	11
6.1.3 - Vorteile ag Motivation	3
6.2 - Vorteile traditiones Vorgehensmode	2
6.2.1 - Vorteile trad. Planungssicherheit	4
6.3 - Vorteile hybr des Vorgehensmode	2
7 - Nachteile gewähltes Vorgehensmode	0
7.1 - Nachteile ag es Vorgehensmode	0
7.1.1 - Nachteile ag fehlende Planungssicherheit	4
7.1.2 - Nachteile Projektcommitment	3
7.1.3 - Nachteile ag Fehlende Erfahrung	2
7.1.4 - Nachteile ag schlechte Dokumentation	1
7.2 - Nachteile traditiones Vorgehensmode	0
7.2.1 - Nachteile trad. zu starr	5
7.2.2 - Nachteile trad. Dauer	3
7.3 - Nachteile hybr des Vorgehensmode	1
7.3.1 - Nachteile hybr d zu starr	1
7.3.2 - Nachteile hybr d Kosten	1
7.3.3 - Nachteile hyb. Fehlende Erfahrung mit Ereignissen der Mode	1
7.3.4 - Nachteile hybr d. komb. Probleme neue Vorgehensweisen	2
8 - Tools Vorgehensmode	0
8.1 - Tools ag es Vorgehen	14
8.2 - Tools trad. Vorgehen	6
9 - Änderungswünsche	0
9.1 - Changemanagement ag es Vorgehen	3
9.2 - Changemanagement hybr des Vorgehen	6
10 - Rollen und Stakeholder im Projekt	0
10.1 - Projektverantwortlicher	9
10.2 - Technische Rollen	6
10.3 - andere Rollen	3

10.4 - Enabler	2
10.5 - Governance Rolle	3
10.6 - Projekt-Schnittstellen-Rolle	7
10.7 - Changemanagement Rolle	2
10.8 - Scrum Master	0
10.8.1 - Scrum Master soziale Projektseite	2
10.8.2 - Vorbereitende Scrum Master Rolle	4
10.9 - Content Developer/ Fachbereichsmitarbeiter	5
10.8.1 - Vorbereitende Content Developer	1
11 - Wichtige Stakeholder außerhalb des Projekts	0
11.1 - Organisatorische Vorgesetzte	7
11.2 - Sonstige Beträge	3
11.3 - Geschäftstätung	3
11.4 - Abteilungen	5
11.5 - Mitarbeitervertretung	3
11.6 - AEMitarbeiter im Unternehmen	2
11.7 - Support Abteilungen	3
12 - Skalierung Projekt	0
12.1 - Sequenzelles Vorgehen	2
12.2 - Paralleles Vorgehen	4
12.3 - Sequenzelles -Paralleles Modell	2
12.4 - Vorbereitende Maßnahmen	10
13 - Befähigung Mitarbeiter	0
13.1 - Befähigung durch Schulung/ Workshop	10
13.2 - Befähigung durch Community	5
13.3 - Befähigung durch Multiplikatoren	6
13.4 - Befähigung durch Selbstlernen	8
14 - Qualitätssicherung Projekt	0
14.1 - Qualitätssicherung durch emotionale Aspekte	2
14.2 - Qualitätssicherung durch Monitoring	4
14.3 - Qualitätssicherung durch Feedback	1
15 - Erfolgsmessung Projekt	0
15.1 - Quantitative Größen	8
15.2 - Qualitative Größen	9

## 1 - Vorbereitung Unternehmen

Dieser Code wird verwendet, wenn eine Aussage sich darauf bezieht, was Unternehmen, bevor sie NCLC-Plattformen einführen, beachten, wissen oder tun sollten.

Entstehung des Codes: Deduktiv

### 1.1 - Auswahl Citizen Developer

Dieser Code wird verwendet, wenn der Interviewpartner darüber spricht wer sich als potentieller Citizen Developer eignet.

Entstehung des Codes: Induktiv

### 1.2 - Motivation Stakeholder

Dieser Code wird verwendet, wenn die Aussage sich auf wichtige Aspekte bezieht, die beachtet werden müssen durch das Unternehmen und die sich mit der Motivation von Stakeholdern beschäftigt.

Entstehung des Codes: Induktiv

### 1.3 - Projektbewusstsein

Dieser Code wird verwendet, wenn eine Aussage getroffen wird, dass in der Vorbereitung der Einführung von NCLC ein Projektbewusstsein in Unternehmen in Bezug für die Auswirkungen von NCLC vorhanden sein muss.

Entstehung des Codes: Induktiv

Beispiel: Naja gefragt, gefragt wurde es schon, aber es gibt tatsächlich einen Punkt, der mir immer wichtig ist bei dem Thema. Geht bewusst an das Thema ran. Also macht daraus ein Projekt. Geht bewusst ran.

### 1.4 - Vorbereitung der Organisation im Unternehmen

Dieser Code wird verwendet, wenn Aussagen getroffen werden, welche organisatorischen vorbereitenden Maßnahmen notwendig sind, vor der Einführung von NCLC.

Entstehung des Codes: Induktiv

### 1.5 - Umgang mit MA als Teil der Unternehmenskultur

Dieser Code wird verwendet, wenn eine Aussage getroffen wird über die Kultur eines Unternehmens in Bezug auf den Umgang mit den Mitarbeitern. Oder der Umgang der Mitarbeiter untereinander.

Entstehung des Codes: Induktiv

Beispiel:

und auch das Vertrauen seinen Mitarbeitern gegenüber, weil, ansonsten lasse ich halt viel Potenzial von so einer No Code/ Low Code- Plattform liegen. Wir haben Kunden, die es tatsächlich einführen, aber dann nur mit tatsächlichen Entwicklern bestücken. Das heißt, die ganzen Citizen Developer bleiben nach wie vor außen vor. Was natürlich dann viel Potenzial von so einer Plattform wieder verschenkt. Zum einen dem Misstrauen geschuldet, dass man sagt: "Ich traue das meinen Mitarbeitern einfach nicht zu." Oder: "Wir können die nicht in den entsprechenden Kontrollmechanismus hineinbringen, wie ich es mit meiner zentralen IT machen kann." oder weil man sagt: "Ich habe einfach nicht die Manpower, um das zu betreuen,

die Geschichte. Wir haben nicht genügend Leute für Projekte, dann können wir jetzt nicht nebenher noch eine neue Plattform betreiben." Dann lass ich halt viel Potenzial liegen, weil, wie gesagt, es geht nicht darum, ein neues Office Paket einzuführen, was ich mit ein, zwei Mann betreiben kann und tragen, sondern No Code/ Low Code-Plattform heißt halt eben auch, ganz neue Aufgabengebiete, die es so vorher nicht gab

### **1.6 - Vorteile NCLC Plattformen**

Dieser Code wird verwendet, wenn Aussagen, die zu positiven Aspekten von NCLC-Plattformen getroffen werden, welche Unternehmen erzielen können durch den Einsatz von NCLC-Plattformen.

Entstehung des Codes: Deduktiv

#### **1.6.1 - Lösung Kommunikationsproblem**

Dieser Code wird verwendet, wenn eine Aussage erfolgt, die sich auf die Lösung des Kommunikationsproblems zwischen Abteilungen bezieht, in Zusammenhang mit NCLC-Plattformen.

Entstehung des Codes: Induktiv

Beispiel:

Die Vorteile sehe ich natürlich insbesondere darin, dass ich kein Kommunikationsproblem mit irgendwelchen Entwicklern habe, denen ich erst mal deutlich machen muss, was ich eigentlich will. Ich kann sozusagen selber meinen eigenen Prozess, so wie er mir vorschwebt, ohne Code-Kenntnisse mit den relativ einfachen Werkzeugen als logischen Prozess besser umsetzen. Also, das ist ein großer Vorteil.

#### **1.6.2 - Prozessverbesserung**

Dieser Code wird verwendet, wenn eine Aussage dazu erfolgt, wie der positive Nutzen von NCLC für Prozesse ist.

Entstehung des Codes: Induktiv

Beispiel:

Realistische Erwartungen aus meiner Sicht, was man heutzutage sehr gut schafft, ist, einfache Prozesse zu automatisieren. So klassische Geschäftsprozesse, wie Rechnungseingang und solche Dinge, Prozesse automatisieren

#### **1.6.3 - Verbesserung der Arbeitsabläufe MA**

Dieser Code wird verwendet, wenn sich eine Aussage auf die Verbesserung von Arbeitsweisen für oder durch MA bezieht, in Zusammenhang mit NCLC.

Entstehung des Codes: Induktiv

Beispiel:

Wenn es aber darum geht, sich einerseits Helferlein im Alltag zu bauen, vielleicht auch, um Sachen zu automatisieren, die man tagtäglich macht oder um fachbereichsspezifische Prozesse geht, dann ist das, glaube ich schon etwas, wo die Endentwickler auch Vorteile erhalten können.

#### **1.6.4 - Steigerung der Qualität von Lösungen**

Dieser Code wird verwendet, wenn eine Aussage sich auf die Verbesserung der Qualität von Problemlösungen bezieht, was erreicht wird durch NCLC.

Entstehung des Codes: Induktiv

Beispiel:

Aber jetzt gerade, weil dein Thema ja auch das Projektmanagement ist, da Sachen, die jetzt bei uns nicht im ERP-System abgebildet werden, wäre ich immer sofort dabei, mach das bitte, bitte, bevor ihr anfangt, es über Excel-Dateien etc. zu machen, nutzt da ein Low Code-Tool, eine Low Code-Plattform dafür. Man hat immer ein besseres Ergebnis, als wenn man das über Kalkulationstabellen macht oder so was.

### **1.6.5 - Höhere Entwicklungsgeschwindigkeit**

Dieser Code wird verwendet, wenn in einer Aussage als positiver Aspekt von NCLC die Steigerung von Entwicklungsgeschwindigkeit angeführt wird.

Entstehung des Codes: Induktiv

Beispiel:

Das hat auch was mit Geschwindigkeit zu tun. Ich bin in den meisten Fällen dann schon relativ schnell fertig. Die Plattformen sind heute wirklich so mächtig und so gut, dass man wirklich in ein paar Stunden schon sehr gute Dinge zustande bringt.

### **1.7 - Nachteile NCLC Plattformen**

Dieser Code wird verwendet, wenn Aussagen getroffen werden darüber, worauf Unternehmen im Vorfeld der Einführung von NCLC achten sollten, was ansonsten zu negativen Konsequenzen führen kann.

Entstehung des Codes: Deduktiv

#### **1.7.1 - schlechte Qualität der Anwendung**

Dieser Code wird verwendet, wenn eine Aussage erfolgt darüber, dass die Qualität der NCLC-Programme schlechter sein kann als die von anderen nicht NCLC-Programmen.

Entstehung des Codes: Induktiv

Beispiel:

Also es gibt natürlich erst mal aus Programmsicht so ein bisschen, sagen wir mal, die Qualität dessen, was da rauskommt, hängt natürlich sehr stark davon ab, wie gut auch das technische und das logische Verständnis desjenigen ist, der programmiert. Es ist wirklich unterschiedlicher Qualität oder, sagen wir mal so, die Qualitätsschwankungen im Vergleich zu einer, sagen wir mal, wirklich von Entwicklern getriebenen Entwicklung ist sicherlich da.

#### **1.7.2 - Falsche Erwartungen**

Dieser Code wird verwendet, wenn Aussagen getroffen werden über zu hohe Erwartungshaltung in Bezug auf NCLC.

Entstehung des Codes: Induktiv

Beispiel:

Wo ich sage, ist die Erwartungshaltung noch ein bisschen übertrieben, ist wirklich das Applikationen erstellen. Also dass ich so Applikationen im Fachbereich erstelle, die extern zum Beispiel angewendet wird, in der Kundenkommunikation und so was. Da, glaube ich, sind wir

noch nicht so weit und da weiß ich auch gar nicht, ob das wirklich so eine Zielsetzung sein kann.

### **1.8 - Governance Einfluss**

Dieser Code wird verwendet, wenn Aussage getroffen werden, die sich auf den Einfluss von NCLC auf Governance Themen beziehen und umgekehrt.

Entstehung des Codes: Induktiv

#### **1.8.1- Fehlende Governance**

Dieser Code wird verwendet, wenn eine Aussage getroffen wird, mit dem Thema Governance-Probleme, darunter fallen auch die Begriffe Schatten-IT und Wildwuchs.

Erstellung des Codes: Induktiv

Beispiel:

Klar, also wenn es aus dem Citizen Development raus betrieben wird, dann ist natürlich die große Gefahr, dass wir Schatten-IT 2.0, nenne ich das immer, haben, also dass wir keine Governance aufgesetzt haben und das eigentlich, ja, sowas wie Excel Script ist, keiner weiß Bescheid, es gibt keine Doku zu gar nichts. Das ist ein großes Thema

#### **1.8.2 - Governancestrategie**

Dieser Code wird verwendet, wenn Aussagen getroffen werden zum Thema Governance mit hinweisendem Charakter, ohne dabei auf positive oder negative Aspekte einzugehen.

Entstehung Code: Induktiv

Beispiel:

Und ich muss auch zwingend mir Gedanken machen über das Thema Governance. Ansonsten ist es toll, dass die Menschen damit arbeiten können, aber ich glaube, dann geht mit der Zeit auch der Nutzen verloren, weil es einfach zu unübersichtlich wird, das Ganze zu verwalten und es geht nur miteinander.

### **1.9 - IT Abteilung und NCLC**

Dieser Code wird verwendet, wenn Aussagen getroffen werden zum Einfluss von NCLC auf die IT-Abteilung.

Entstehung Code: Induktiv

#### **1.9.1 - Negative Auswirkungen für IT**

Dieser Code wird verwendet, wenn eine Aussage getroffen wird zu negativen Auswirkungen von NCLC, in Beziehung auf die IT-Abteilung.

Entstehung des Codes: Induktiv

Beispiel:

Oder falls meine Supportorganisation nicht darauf vorbereitet ist, dass da jetzt auf einmal statt zehn Entwicklern hundert Entwickler in meinem System unterwegs sind, die dann im Zweifelsfall geflutet werden von Serviceanfragen, von Supportanfragen, auf die sie nicht vorbereitet sind, habe ich natürlich auch ein Problem.

#### **1.9.2 - Entlastung IT-Situation**

Dieser Code wird gebildet, wenn Aussagen darüber getroffen werden, wie eine Entlastung der IT durch NCLC erfolgen kann.

Entstehung des Codes: Induktiv

Beispiel:

Und natürlich könnte man jetzt auch sagen, die überlasteten IT-Leute werden da entlastet. Aber das sind alles Nebenvorteile, sage ich mal. Hauptsächlich geht es darum, dass keine Reibungsverluste entstehen

### 1.9.3 - Fachkräftemangel IT

Dieser Code wird verwendet, wenn sich eine Aussage auf den Fachkräftemangel in der IT bezieht und wenn eine Einschätzung getroffen wird, wie NCLC und Citizen Development in diesem Zusammenhang Einfluss haben können.

Entstehung des Codes: Induktiv

Beispiel

Also ich glaube nicht, dass man die IT damit ersetzen kann. Ich glaube, dass man die IT damit sehr stark unterstützen kann.

## 2 - EP: Eigenschaften Unternehmen

Dieser Code wird verwendet, wenn Aussagen darüber getroffen werden, welche Voraussetzungen Unternehmen erfüllen müssen, um sich für eine Einführung zu qualifizieren.

Entstehung des Codes: Deduktiv

### 2.1 - Mindset

Dieser Code wird verwendet, wenn der Interviewpartner über Mindset und Unternehmenskultur spricht aus Sicht des Unternehmens.

Entstehung des Codes: Deduktiv

Beispiel:

Was es aber bedarf ist, wie Sie es gesagt haben, auch der Mindset, aber das Mindset, glaube ich, kommt von 2 Seiten. Einerseits von der Belegschaft selber, dass man dort auch wirklich eine Akzeptanz fördert und dort auch deutlich macht, was die Vorteile sind oder in gewisser Weise, man wird nicht alle überzeugen, gerade bei großen Unternehmen, sondern ein paar Leuchttürme hat, die dafür stehen und die anderen mitziehen. Das Zweite ist aber, was ganz wichtig ist, das Commitment von oben. Was meine ich damit? Es muss natürlich der Vorstand bzw. ein IT-Vorstand auch für diese Transformation stehen und auch sagen, das ist etwas, was wir jetzt machen und es darf ihm nicht aufgedoktriniert sein. Das heißt, am besten, was heute gut funktioniert und was ich schon gesehen habe, ist das einzubetten in eine digitale Agenda oder in eine grundsätzliche Digitalisierungsstrategie. Und wenn es da mit drin ist und verankert ist, dann funktioniert das auch gut.

#### 2.1.1 - Fehlerkultur

Dieser Code wird verwendet, wenn sich Aussagen auf die Fehlerkultur in einem Unternehmen beziehen.

Entstehung des Codes: Induktiv

Beispiel:



Die Risiken sind die, dass man zu wenig fehlertolerant ist. Das heißt, wenn ich jetzt hingehere und probiere etwas aus und das funktioniert jetzt nicht sofort, dass man dann sagt: "Mensch, das ist jetzt wieder alles schiefgelaufen und das ist alles Mist." Also diese Einstellungssache, Mindset gehört da auch rein und dass man dann sagt: "Ok, wir haben hier etwas entdeckt, was jetzt noch nicht so ganz gut funktioniert. Macht es noch Sinn, in dieser Richtung weiterzugeben? Oder müssen wir was ganz anderes als Lösung versuchen?" Also diese Start-up-Mentalität, die muss dann rein.

## 2.2- Kapital

Dieser Code wird verwendet, wenn der Interviewpartner über Geld spricht, welches ein Unternehmen haben sollte.

Entstehung des Codes: Deduktiv

Beispiel:

Also, fangen wir vielleicht mal mit dem Kapital an. Also, prinzipiell sollte von der obersten Unternehmensführung, egal welche Größe, da gesagt werden, das ist eine Investition, dafür stellen wir die finanziellen Mittel bereit. Also das ist eine Basisvoraussetzung, damit da einfach kein Druck herrscht, weil, man kann einfach nicht auch entwickeln, wenn man in einem Spitzdruck ist. Das ist einfach auch eine Sache, Projekte sind auch so eine Sache, wo Geld verbrannt wird. Das muss klar sein.

## 2.3 - Unternehmensgröße

Dieser Code wird verwendet, wenn es um die Anzahl der Mitarbeiter oder die Form des Unternehmens und dessen Größe geht.

Entstehung des Codes: Deduktiv

Beispiel:

Die betrifft im Zweifelsfall das ganze Unternehmen und genau so muss ich die auch aufhängen. Dass es eben nicht mit den drei Admins getan ist, die das jetzt noch nebenhermachen, sondern es ist ja ein organisationsweiter Change, den ich da einführe und den muss ich auch mit der entsprechenden Manpower unterfüttern.

## 2.4 - Technische Planung

Dieser Code wird verwendet, wenn eine Aussage getätigt wird über technische Eigenschaften von Unternehmen, die in Zusammenhang mit NCLC wichtig sind.

Entstehung des Codes: Induktiv

Beispiel:

Ja, also ich sage jetzt mal, tatsächlich harte Voraussetzungen sind eigentlich nur technische Systemvoraussetzung um die Plattform, also den Designer zu bedienen, diese App zu bauen und dann die entsprechenden Betriebssysteme und Endgeräte, um die Plattform zu betreiben. Ich würde sagen, ansonsten gibt es A priori keine Voraussetzungen.

## 2.5 - Erfahrenes Projektmitglied

Dieser Code wird verwendet, wenn als Voraussetzung ein erfahrenes Projektmitglied benannt wird.

Entstehung des Code: Induktiv

Beispiel:

### 3 - EP: Risiken und Hürden vor der Einführung

Welche Risiken und Hürden sind zu beachten

Entstehung des Codes: Deduktiv

#### 3.1 – Schatten-IT

Dieser Code wird verwendet, wenn als Hürde/ Nachteil/ Problemstellung während des Einführungsprojektes die Schatten-IT oder Wildwuchs oder ähnliche Begriffe genannt wird.

Entstehung des Codes: Induktiv

Beispiel:

Also, dass diese Schatten-IT 2.0 eben nicht passiert, muss man ein gewisses Level an Governance aufbauen. Das kann ja auch ein bisschen wachsen mit der Low Code- Plattform. Aber man sollte sich von Anfang an bewusst sein, dass man so was braucht und dass jemand dafür abgestellt ist, der sich damit beschäftigt von: Kritische Konnektoren blocken oder überwachen, Neuerungen in der der Plattform dann entsprechend den Policies umsetzen/ einbinden, also da braucht man einfach jemanden, sollte man jemanden haben, der das macht. Ist das Erste, und das Zweite ist, woran es immer scheitert, es muss ein Bewusstsein für Daten geben, finde ich. Also es muss Leute geben, die Datenstrukturen in Datenbanken denken können, sonst funktioniert es nicht, weil, damit kann ich dann eben auch den Vorteil realisieren, dass ich halt Sachen nicht in einem Excel Spreadsheet abbilde, sondern in einer relationalen Datenbank.

#### 3.2 - Cybersecurity und Datenschutz

Dieser Code wird verwendet, wenn der Interviewpartner über die Themen Cybersecurity und Datenschutz redet, in Verbindung mit Problemen und Risiken.

Entstehung des Codes: Induktiv

Beispiel:

wenn ich jetzt einen Fachbereich ermächtige, große Workflows für die Abläufe auf IT-Basis für Unternehmen zu erstellen, dann muss ich halt, unabhängig davon, ob die Leute Code schreiben wollen oder nicht, muss ich die ganzen Umgebungsbedingungen, die muss ich halt auch bedenken. Das heißt, es geht um Datensicherheit bei der Richtung, dass einerseits die Sicherheit der Daten, aber andererseits die Verweigerung von unberechtigten Zugriffen auf die Daten, also das ist die Datenkonsistenz zum Beispiel, das gehört auch noch zur Datensicherheit.

#### 3.3 - Thema unterschätzt

Dieser Code wird verwendet, wenn bei Risiken und Hürden sich die Aussage darauf bezieht, dass NCLC falsch eingeschätzt oder unterschätzt wird.

Entstehung des Codes: Induktiv

Beispiel:

Falsche Erwartungen ist ein Problem, ein Risiko, das einfach nicht, also es gibt ja klassischerweise den Anforderer dann an Problemen, ist jetzt vielleicht auch der Chef irgendwie. Der hat dann von der Low Code-Plattform gehört und verdonnert seinen Mitarbeiter, damit was zu machen und hat einfach falsche Erwartungen, was man damit machen kann und was auch ein Mitarbeiter, der neu in so einer Rolle ist oder so etwas zum ersten Mal macht, was er damit bewirken kann.

### 3.4 - Weitere Risiken

Dieser Code wird verwendet, wenn Einzelrisiken genannt werden, die interessant sind, aber selten genannt wurden.

Entstehung des Code: Induktiv

### 4 - EP: Projektorganisationsform

Welche Projektorganisationsform sollte für das Projekt gewählt werden.

Entstehung des Codes: Deduktiv

### 5 - Wahl Vorgehensmodell

Welches Vorgehensmodell sollte gewählt werden.

Entstehung des Codes: Deduktiv

#### 5.1 - Agiles Vorgehensmodell

Dieser Code wird verwendet, wenn sich für ein agiles Vorgehensmodell entschieden wird.

Entstehung des Codes: Induktiv

##### 5.1.1 - Option auf Wasserfallelemente

Diese Code wird gebildet, wenn ein Interviewpartner klar ein agiles Modell favorisiert, aber dort eventuell auch Methoden der klassischen Modelle davor oder danach nicht ausschließt.

Entstehung des Code: Induktiv

Beispiel:

Zu einem hybriden Modell... Kann man muss man aber nicht. Also man könnte natürlich tatsächlich für so ein Projekt auch ein Wasserfall drum herum legen, sage ich mal, für so ein hybrides Modell.

##### 5.1.2 - Begründung agiles Vorgehen

Dieser Code wird verwendet, wenn eine Begründung für ein agiles Vorgehen angeführt wird und Gegenargumente erbracht werden gegen ein traditionelles Vorgehen.

Entstehung des Code: Induktiv

Beispiel:

Also, wir sind normalerweise agil unterwegs. Das heißt, wir fangen an mit einem Sprint Null, in dem wir erst mal eine Reihe von Workshops durchführen, um den Kunden auf ein gewisses Niveau aufzuschauen und nach diesem Sprint Null planen wir dann wirklich Entwicklungssprints, um zu sehen: "Ok, das sind jetzt die Arbeitspakete, die wir tatsächlich implementieren möchten, in dem und dem Zeitraum."

#### 5.2 - Hybrides Vorgehensmodell

Dieser Code wird verwendet, wenn ein hybrides Vorgehensmodell gewählt wird.

Entstehung des Codes: Induktiv

### 5.2.1 - Ausprägung Hybrides Modell

Dieser Code wird verwendet, wenn Aussagen darüber getroffen werden, wo der Schwerpunkt des hybriden Modells liegen soll.

Entstehung des Codes: Induktiv

Beispiel:

Also, da ich es auch ein Projekt sehe, was immer am Laufen ist, hat halt keinen klaren Abschluss. Also, eine Low Code-Plattform an sich, ich sag nicht irgendwann, die ist eingeführt also, sondern es gibt dann immer neue Herausforderungen. Es gibt neue Funktionalitäten, die man vielleicht einführen/ schmackhaft machen will. Man hat also gerade speziell auch in der Microsoft Power Platform, aber ich glaube auch in allen Low Code-Plattformen, hat man sehr hohe Entwicklungs-, also, die Plattform selber hat noch eine hohe Veränderungsgeschwindigkeit, wie die sich verändert. Von daher, denke ich, kommt man um einen agilen Ansatz da gar nicht drumherum, weil, Microsoft schmeißt dann einfach noch mal irgendwo einen Stock in die Speichen irgendwas ändert sich und dann musst du schnell darauf reagieren, also da kann ich nicht klassisch in 3 Phasen das durchziehen, denke ich.

### 5.2.2 - Begründung hybrides Vorgehen

Dieser Code wird verwendet, wenn Aussagen getroffen werden, warum ein hybrides Modell sich für die Einführung besser eignet als andere Modelle.

Entstehung des Code: Induktiv

Beispiel:

Das kann man nicht, oder ich würde eher jetzt spontan zu einer hybriden Lösung tendieren. Aus meiner Sicht ist es so, dass eine rein agile Lösung im klassischen Sinne, ein reines Modell, unvermischt, laut Lehrbuch, findet man in den meisten Projekten nicht. Es wäre maximal in so einer projektorientierten Umgebung möglich, aus meiner Sicht, wo wir wirklich Experten haben, die auch in diesem Modus schon zusammengearbeitet haben. Die also genau wissen, was sie zu tun haben. Wo ihr Fokus darauf ist und noch in der Lage sind, selbsttempowert, sage ich mal, Entscheidungen zu treffen und, zum Beispiel, jetzt die Sprintziele festzulegen und das Ganze dann auch halt selbst motiviert voranzutreiben.

## 6 - Vorteile gewähltes Vorgehensmodell

Dieser Code wird verwendet, wenn es um Vorteile der verschiedenen Vorgehensmodelle geht.

Entstehung des Codes: Deduktiv

### 6.1 - Vorteile agiles Vorgehen

Dieser Code wird verwendet, wenn Vorteile für ein agiles Vorgehens genannt werden.

Entstehung des Codes: Induktiv

#### 6.1.1 - Vorteil agil schnelles Feedback

Dieser Code wird verwendet, wenn schnelles Feedback als Vorteil des agiles Vorgehens genannt wird.

Entstehung des Codes: Induktiv

Beispiel:

Also, die Vorteile vom agilen sind, ich habe es schon ein bisschen so erwähnt, tatsächlich die Tatsache, dass ich relativ schnell Feedback kriege über das, was ich mache. Ich arbeite ja sozusagen in so einem Projekt typischerweise nicht einfach so, dass ich erst die Plattform einführe und dann mit Anwendern arbeite, sondern ich arbeite mit einigen Anwendern daran, die Plattform einzuführen und zu gucken, wie kann man die nutzen in ihrem Arbeitsbereich. Dann habe ich ein schnelles Feedback und kann damit wieder überlegen, was muss ich in der Konfiguration der Plattform machen für die nächste Gruppe von Anwendern. Also diese kurzen Feedbackschleifen sind ein extremer Vorteil in solchen Projekten.

### **6.1.2 - Vorteil agil Reaktionsgeschwindigkeit**

Dieser Code wird verwendet, wenn eine Aussage darüber getroffen wird, was vorteilhaft beim agilen Vorgehen, in Bezug auf die Reaktionsgeschwindigkeit, ist.

Entstehung des Codes: Induktiv

Beispiel:

Und ich kann auch mit relativ wenig Kosten schnell wieder aussteigen, wenn ich sage: "Nee, das ist für uns nichts." Oder "Die Plattform ist für uns nichts. Wir nehmen mal eine andere." Das kann ich bei einem fertig geplanten, also bei einem klassischen, sequenziellen Projekt habe ich das nicht. Da habe ich am Ende alles eingeführt und keiner kann damit arbeiten. Das wollen wir nicht.

### **6.1.3 - Vorteil agil Motivation**

Dieser Code wird verwendet, wenn als Vorteil einer agilen Vorgehensweise die Motivation angeführt wird.

Entstehung des Codes: Induktiv

Beispiel:

Der motivatorische Faktor bei agilem Vorgehen spielt auch noch eine Rolle. Mit einer kleinen Gruppe von Anwendern schnell mal eben was umzusetzen, passt eigentlich wunderbar zu Low Code/ No Code. Also da würde ich immer sehr stark dazu tendieren, das ist agil. PMI nennt das ja sogar hyper agil.

## **6.2 - Vorteile traditionelles Vorgehensmodell**

Dieser Code wird verwendet, wenn Vorteile des traditionellen Vorgehensmodells genannt werden.

Entstehung des Codes: Induktiv

### **6.2.1 - Vorteil trad. Planungssicherheit**

Dieser Code wird gebildet, wenn eine Aussage dazu erfolgt, dass durch Verwendung von traditionellen Anteilen eine Erhöhung der Planungssicherheit entsteht.

Entstehung des Codes: Induktiv

Beispiel:

Einfach um Dinge, diese konzeptionellen Probleme relativ frühzeitig erkennen zu können. Als aber auch hier eine Direktive vorzugeben. Wo wollen wir denn hin und was sind unsere Meilensteine? Also diese Definition, damit wir auch wissen, was sind unsere Deliverables in der Zeit X. Das muss ganz klar auf dem Schirm sein und meine Erfahrung ist einfach, auch wenn man agil im Prinzip den Scope immer wieder entscheidet, das große Ganze, dieser Endtermin,

das darf nicht aus dem Blick geraten und hier bedarf es schon einer klassischen Unterstützung von einem Projektmanager an der Stelle.

### 6.3 - Vorteile hybrides Vorgehensmodell

Dieser Code wird verwendet, wenn Vorteile des hybriden Vorgehensmodells genannt werden.

Entstehung des Codes: Induktiv

Beispiel:

Auf der anderen Seite würde ich von der Seite der agilen Methoden, kann ich mir vorstellen, dass man durchaus auch in Sprints arbeitet, also wenn man zum Beispiel darüber spricht, dass man in einer iterativen Vorgehensweise einzelne Fachbereiche in dieses Projekt hinzuholt und die darin schult. Das wäre eine Sache, wo man iterativ vorgehen könnte, einfach damit man die Gelegenheit hat, aus den Erfahrungen des einen Onboarding des Fachbereiches für den nächsten zu lernen. Ich würde bei den agilen Methoden auch Wert darauf legen, schnell Erfolge sichtbar zu machen, dass man also nicht mit einem beispielsweise Big Bang arbeitet und alles passiert im Kämmerchen bis das ganze Thema leicht geht, sondern man muss eben zwischendrin auch die Erfolge sichtbar machen. Also immer ein Stückchen mehr, ja, Inkrement sozusagen, ein Stückchen mehr zeigen, was man geschaffen hat, um die Projektmitglieder einfach am Ball zu behalten und auch den Stakeholdern zu zeigen, dass man hier konsequent Mehrwert schafft für die Organisation.

## 7 - Nachteile gewähltes Vorgehensmodell

Welche Nachteile hat gewähltes Vorgehensmodell?  
Worauf muss geachtet werden?

Entstehung des Codes: Deduktiv

### 7.1 - Nachteile agiles Vorgehensmodell

Dieser Code wird verwendet, wenn Nachteile des agilen Vorgehensmodells genannt werden bzw. Dinge, auf die bei diesem Vorgehen geachtet werden muss.

Entstehung des Codes: Induktiv

#### 7.1.1 - Nachteil agil fehlende Planungssicherheit

Dieser Code wird verwendet, wenn im Zusammenhang mit agilen Vorgehen eine negative auf Aussage auf die Planungssicherheit getätigt wird.

Entstehung des Codes: Induktiv

Beispiel:

So, dem gegenüber steht natürlich auch der gewisse Projektdruck, den man hat, ich sage mal, Zeit und Budget hängen einem im Nacken. Das ist immer ein Problem. Mit der Zeit, das ist wie Raum und Zeit bei Einstein, die korrelieren miteinander. Überschreite ich die Zeit, überschreite ich quasi auch das Budget, das ist ein Punkt letztendlich. Und man hat natürlich auch die Umsetzungsqualität im Nacken, weil man natürlich als Projektmanager weiß, dass die schweren Probleme, die schweren Fehler, die werden ganz am Anfang gemacht. Das sind in der Regel Designfehler, konzeptionelle Fehler und hier, genau hier in diesem Punkt, halte ich das total wichtig, eine Waterfall-Schiene mit einzubauen.

### 7.1.2 - Nachteil Projektcommitment

Dieser Code wird verwendet, wenn als Nachteil für agile Methoden angeführt wird, dass dort Probleme im Projektablauf entstehen können, durch den agilen Charakter.

Entstehung des Codes: Induktiv

Beispiel:

Stolpern, also eine Hürde ist bei so einem Vorgehen natürlich, da man bei solchen Projekten häufig auch mit wechselnden Teilnehmern, je nachdem, mit welchem Fachbereich man gerade zu tun hat, arbeitet. Dass man so ein bisschen die Konstanz in der Arbeit verliert. Dass manche Leute punktuell mal was machen und dann bleibt das alles liegen. Dann habe ich so meine halbfertigen Applikationen, oder Applikationen, die gut genug sind, dass sie sie anwenden können und dann kümmert sich keiner mehr darum. Also das ist natürlich eine Sache, die bei agilem Vorgehen schnell mal passiert. Insbesondere nämlich, wenn das Team nicht konstant ist.

### 7.1.3 - Nachteil agil Fehlende Erfahrung

Dieser Code wird verwendet, wenn eine Aussage dazu getroffen wird, dass es in Unternehmen zu Problemen mit der Anwendung einer agilen Vorgehensweise kommen kann, wenn diese wenig oder keine Erfahrung mit diesem Vorgehen haben.

Entstehung des Codes: Induktiv

Beispiel:

Ja, ist jetzt aber weniger der Power Platform geschuldet als der agilen Methodik insgesamt. Wenn ich jetzt in ein Unternehmen komme, die mit der agilen Methode bisher noch nie zusammengearbeitet haben, die tun sich natürlich erst mal schwer, weil ich dann erst mal eine agile Projektvorgehensweise mit dem Kunden klären muss, bevor es überhaupt an die Inhalte gehen kann. Also, das ist so die Hürde. Ist aber losgelöst von dem Thema Power Platform an sich. Das Gleiche hätte ich, wenn es jetzt um eine Dynamics Einführung geht, zum Beispiel. Im schlimmsten Fall geht man dann halt auch wirklich auf ein Wasserfallmodell über, wenn der Kunde sagt: "Agil ist definitiv nicht unseres. Wir brauchen den kompletten Plan von vorn bis hinten." Dann machen wir das auch.

### 7.1.4 - Nachteil agil schlechte Dokumentation

Dieser Code wird verwendet, wenn eine Aussage darüber erfolgt, das fehlende Dokumentation ein Nachteil für agiles Vorgehen sein kann.

Entstehung des Codes: Induktiv

Beispiel:

Genau. Die Nachteile sind in der Regel ähnlich wie bei der IT. Niemand, der ein Programm schreibt, liebt es zu dokumentieren. Also es werden Dokumentationen gemacht, aber ich habe bisher noch keinen getroffen, der außer Lippenbekenntnissen zur Dokumentation wirklich dann auch dafür brennt, dass was er tut, da gut dokumentieren. Dass das getan wird, das ist ein Ausdruck der Professionalität und das ist ganz gut. Das heißt, je besser ein Programm, was ich beschrieben habe, dokumentiert ist, desto einfacher ist es und jetzt kommt das Stichwort, das ganz wichtig ist: zu pflegen. Wenn ich ein Programm schlecht dokumentiere und ich bin dann weg, wo auch immer hin.

### 7.2 - Nachteile traditionelles Vorgehensmodell

Dieser Code wird verwendet, wenn Nachteile des traditionellen Vorgehensmodells genannt werden bzw. Dinge, auf die bei diesem Vorgehen geachtet werden muss.

Entstehung des Codes: Induktiv

### **7.2.1 - Nachteil trad. zu starr**

Dieser Code wird verwendet, wenn als Nachteil für ein traditionelles Vorgehen die zu starre Struktur oder Organisation angeführt wird.

Entstehung des Code: Induktiv

Beispiel:

Das heißt, wenn ich die jetzt alle in ein Wasserfall versuchen würde reinzupacken. Der Wasserfall ist immer nur so schnell wie das langsamste Teammitglied und in der agilen Methode habe ich eben die Chance, Sachen auch vernünftig zu parallelisieren und zu sagen: "Ok, im Entwickler-Stream bauen wir jetzt diese zehn Applikationen. Das ist unser Entwicklungs-stream und im Thema Changemanagement, das sind die drei Communitys, die wir aufsetzen, die parallel dazu laufen."

### **7.2.2 - Nachteil trad. Dauer**

Dieser Code wird verwendet, wenn eine Aussage zum traditionellen Vorgehen getätigt wird, welche die negativen Auswirkung dieses Vorgehens auf die Dauer hervorhebt.

Entstehung des Code: Induktiv

Beispiel:

Es ist halt unsere Empfehlung das zu tun, weil wir damit in der Regel bessere Ergebnisse erzielen, wenn der Kunde den Weg aber so nicht mitgehen kann, dann machen wir es auch ganz normal im Wasserfall. Die Geschichte dauert in der Regel länger und ist dann halt auch schwieriger für den Kunden zu greifen, weil es dann halt auch einfach länger dauert, bis ich mal was in der Hand habe hinterher.

## **7.3 - Nachteile hybrides Vorgehensmodell**

Dieser Code wird verwendet, wenn Nachteile des hybriden Vorgehensmodells genannt werden bzw. Dinge, auf die bei diesem Vorgehen geachtet werden muss.

Entstehung des Codes: Induktiv

### **7.3.1 - Nachteil hybrid zu starr**

Dieser Code wird verwendet, wenn eine Aussage darüber erfolgt, dass es bei hybridem Vorgehen nachteilig sein kann, dass dieses zu starr ist.

Entstehung des Codes: Induktiv

Beispiel:

Das würde ich bei einem plangetriebenen Ansatz oder auch bei einem hybriden Ansatz habe ich es häufig so, dass ich sehr viele Vorinvestitionen habe, um erst mal einen Plan aufzubauen, um erst mal die Plattform zu checken, um erst mal die Plattform auszuwählen und so weiter und so fort. Das wird deutlich teurer und führt auch eher zu Frustration.

### **7.3.2 - Nachteil hybrid Kosten**

Dieser Code wird verwendet, wenn eine Aussage darüber getroffen wird, dass die Kosten von der Anwendung höher sind als bei einem anderen reinen Modell.

Entstehung des Codes: Induktiv



Beispiel:

Das würde ich bei einem plangetriebenen Ansatz oder auch bei einem hybriden Ansatz habe ich es häufig so, dass ich sehr viele Vorinvestitionen habe, um erst mal einen Plan aufzubauen, um erst mal die Plattform zu checken, um erst mal die Plattform auszuwählen und so weiter und so fort. Das wird deutlich teurer und führt auch eher zu Frustration.

### **7.3.3 - Nachteil hyb. Fehlende Erfahrung mit einem der Modelle**

Dieser Code wird gebildet, wenn Aussagen darüber getroffen werden, dass es zu Problemen kommen kann, wenn das Unternehmen keine Erfahrung hat, mit entweder dem agilen oder dem traditionellen Vorgehen.

Entstehung des Codes: Induktiv

Beispiel:

Na gut, das kommt darauf an. Also wenn eine Unternehmung schon mehrere agile Projekte gefahren hat und weiß, wie es geht, was meine ich damit, die Mitarbeiter haben schon Erfahrung damit, dann sehe ich da keine Hürden. Aber wenn das sozusagen als Vehikel benutzt werden würde, um eines der ersten agilen Projekte zu nutzen, dann ist das schon kompliziert, weil, es kennt noch keiner so richtig die Rollen. Scrum Master, was macht der eigentlich den ganzen Tag? Für was ist der eigentlich zuständig? Ach so, der soll so Daily moderieren, aber da brauch ich doch keinen für. Also Daily, wieso soll ich jeden Morgen sagen, welche Sachen ich mache und wieso soll ich mich jetzt committen, was ich in zwei- oder vierwöchigen Sprints erledige. Und das sind, glaube ich, Punkte, das ist natürlich eine Hürde, die dann definitiv mit reinspielt, wie weit ist dieses Mindset dafür und was natürlich auch eine Hürde sein kann, ist immer bisschen der Punkt, dass Agilität so ein bisschen manchmal als Flexibilität wahrgenommen wird. Das heißt, dass vielleicht gewisse Stakeholder oder Personen auch das Backlog stark beeinflussen möchten und auch vielleicht die Priorisierung des Backlogs dabei. Was natürlich nicht immer so an der Stelle vorgesehen ist, wenn man es einmal abgestimmt hat. Und das sind so ein bisschen die Hürden, die man schon so sieht und die früher in einem klassischen Prinzip, da weiß halt jeder, was er zu tun hat, was er sein lassen darf und muss oder was er machen darf.

### **7.3.4 - Nachteil hybrid. kombi. Probleme reine Vorgehensweisen**

Dieser Code wird verwendet, wenn eine Aussage darauf hinweist, dass bei einem hybriden Vorgehen zu einem gewissen Anteil auch die Nachteile der jeweiligen reinen Modelle eingebracht werden und diese sich nicht komplett aufheben.

Entstehung des Codes: Induktiv

Beispiel:

Also, definitiv ist das auch das nicht die optimale Lösung. Ich kann dir jetzt einfach nur sagen, die Nachteile sind halt die, die jetzt auch die beiden Einzel- oder reinen Modelle mitbringen würde jeweils auf ihrer Seite. Also es gibt jetzt keinen zusätzlichen hybriden Nachteil.

## **8 - Tools Vorgehensmodell**

Welche Tools des Vorgehensmodells werden genutzt.

Entstehung des Codes: Deduktiv

### 8.1 - Tools agiles Vorgehen

Dieser Code wird verwendet, wenn sich für ein agiles Vorgehen entschieden wurde und Tools aus dem agiles genannt werden.

Entstehung des Codes: Induktiv

Beispiel:

Aber, wir haben unsere Weekly's, wo wir die Themen besprechen, dann bisschen langfristige Planung, wo wir hinwollen, wo wir zusammenarbeiten, aber nutzen da jetzt auch keine festen Tools aus dem agilen Baukasten.

### 8.2 - Tools trad. Vorgehen

Dieser Code wird verwendet, wenn Aussagen getätigt werden, welche Tools sich für das Vorgehensmodell eignen und dort Bestandteile des traditionellen Vorgehens gewählt werden.

Entstehung des Codes: Induktiv

Beispiel:

Also insgesamt in der Projektarbeit, weil man davon ausgehen muss, dass die Kollegen, die das machen, dann tatsächlich diesem Daily Business verhaftet sind, ihren täglichen Routinen. Also da fehlt die Projekterfahrung, deswegen wäre es eher ein Waterfall-Ansatz. Dass es jemanden gibt, der so ein bisschen den Rahmen festlegt und die nächsten Schritte, aber und das kommt dazu vom hybrid her, durchaus hybride Teile einstreut. Das ist, glaube ich, das was ich sehe, was sich bewährt.

## 9 - Änderungswünsche

Wie wird mit Änderungswünschen an das Projekt umgegangen?

Entstehung des Codes: Deduktiv

### 9.1 - Changemanagement agiles Vorgehen

Dieser Code wird verwendet, wenn sich für ein agiles Vorgehen entschieden wurde und Aussagen getroffen werden, wie mit Änderungen umgegangen wird.

Entstehung des Codes: Induktiv

Beispiel:

Wir arbeiten auch agil. Die Änderungswünsche werden in den entsprechenden Backlog überführt bzw. mit den bestehenden Backlog Items abgeglichen. Also entweder es ist eine neue Anforderung oder es ist eine geänderte Anforderung oder auch der Wegfall der Anforderung. Das wird entsprechend im Backlog vermerkt und dann findet die wieder Einsortierung in den regulären agilen Fluss, in das agile Projektmanagement. Also Bewertung der Items ziehen in den aktuellen Sprint und Umsetzen.

### 9.2 - Changemanagement hybrides Vorgehen

Dieser Code wird verwendet, wenn sich für ein hybrides Modell entschieden wurde und Aussagen über Änderungswünsche, die an das Projekt gestellt werden, getätigt werden.

Entstehung des Codes: Induktiv

Beispiel:

Genau. Also, wenn man diesen Backlog dann so definiert, als einige Karten, die ich zum Beispiel in ein Kanbanboard einsetze, dann wäre das eine Möglichkeit, das wäre eine Ebene zu

sagen: "Ich als User hätte gerne und dann so und so, kann man es umsetzen. Kannst du das machen?" Das ist richtig, aber das würde dann innerhalb des Projektes, innerhalb der Fachabteilung, wie ich es gerade beschrieben habe, laufen. Also da gäbe es dann ein Backlog, sozusagen, des Fachbereichs für den Fachbereich und der würde es dann umsetzen und würde es direkt zeigen können auch.

## **10 - Rollen und Stakeholder im Projekt**

Welche wichtigen Rollen gibt es im Projekt?

Entstehung des Codes: Deduktiv

### **10.1 - Projektverantwortlicher**

Dieser Code wird verwendet, wenn jemand die Rolle eines Projektverantwortlichen als wichtig empfindet. Dieser vertritt das Team in der Organisation und im Unternehmen und stellt finanzielle Mittel zu Verfügung. Er gibt auch Input, in Form von Wünschen, die an das Projekt gestellt werden, hat aber in Abgrenzung zu Projektorganisationsrollen keine planende oder organisatorische Funktion. Dies entspricht dem Product Owner im agilen Vorgehen und dem Sponsor im traditionellen Vorgehen.

Entstehung Code: Induktiv

Beispiel:

Also schon einen Product Owner, Low Code-Plattform hätte ich gerne, hatten wir nicht, hätte ich gerne, der über alle Themen so ein bisschen darüber guckt und danach guckt, dass die Fäden alle richtig zusammen laufen.

### **10.2 - Technische Rollen**

Dieser Code wird verwendet, wenn als wichtige Rolle im Projektteam eine technische/ fachliche Rolle genannt wird.

Entstehung des Codes: Induktiv

Beispiel:

Dann habe ich meinen String des Developer enablements und hier trainiere ich jetzt tatsächlich die Kundenressourcen und auch die Citizen Developer, bereite Applikationen vor.

### **10.3 - andere Rollen**

Dieser Code wird verwendet, wenn eine Aussage darüber getätigt wird, dass eine Rolle im Projekt vorhanden sein soll, diese Rolle aber nicht näher beschrieben wird und nur grob umschrieben wird.

Entstehung des Codes: Induktiv

Beispiel:

Dann gibt es aber auch den Citizen Developer Strategist. Der Strategist hat die Aufgabe, aus Unternehmenssicht heraus zu sagen: "In dieser und jener Form wollen wir diese Applikationen aufbauen. Da und darauf sollen die ja einzahlen." Also, auch wenn die No Code/ Low Code-Applikationen durch den Fachbereich betrieben werden können, ist das keine App für 14,90 €, sondern das sind veritable Investitionen, die ein Unternehmen leisten muss, die sind teuer. Also das muss man ganz klar sagen.

#### **10.4 - Enabler**

Dieser Code wird verwendet, wenn die Rolle des Enablers als Teil des Projektteams gesehen wird.

Entstehung des Codes: Induktiv

Beispiel:

Da hilft es, also ich habe da immer ein offenes Ohr für die Leute, die können sich direkt an mich wenden und ich versuche schnell zu helfen und durch die Rolle von einem Enabler haben wir auch eine Community etabliert,

#### **10.5 - Governance Rolle**

Dieser Code wird verwendet, wenn die Rolle eines Governance-Beauftragten im Projektteam als sinnvoll angesehen wird.

Entstehung des Codes: Induktiv

Beispiel:

Also, es muss mindestens eine Person mit Erfahrung reinkommen. Natürlich, die das Ganze ein bisschen steuert. Ob extern oder intern ist da egal, es muss von Anfang an eine Governance Struktur mit dabei sein, also Administrierung von der Plattform muss es dafür geben und muss auch eine eigene Rolle sein, die nicht unbedingt, also es muss kein Entwickler sein, sondern dieser Person muss dafür eine gewisse Zeit eingeräumt werden.

#### **10.6 - Projekt-Schnittstellen-Rolle**

Dieser Code wird verwendet, wenn eine Rolle im Team erwähnt wird, die sich mit Schnittstellen-Aufgaben beschäftigt, zwischen Projektteam und Organisation.

Entstehung des Codes: Induktiv

Beispiel:

wir setzen uns normalerweise zusammen, ein Projektteam auf unserer Seite, mit einem insgesamt Architekten. Also jemand wie ich, der die Fäden zusammenhält und dann habe ich verschiedene Consultants und weitere Architekten dabei, die die einzelnen Teilströme abdecken.

#### **10.7 – Changemanagement-Rolle**

Dieser Code wird verwendet, wenn ein Rolle erwähnt wird für den Bereich Changemanagement im Projektteam.

Entstehung des Codes: Induktiv

Beispiel:

Dann der Bereich des Changemanagements, also wie muss sich die Organisation vorbereiten, im Sinne von Training? Und als Letztes den Bereich die Supportorganisation. Wie verknüpfe ich jetzt meine neue Plattform, meine Citizen Developer, mit den entsprechenden internen Supportstrukturen.

#### **10.8 - Scrum Master**

Dieser Code wird verwendet, wenn die Rolle des Scrum Master erwähnt wird.

Entstehung des Codes: Induktiv

Beispiel:

**I: RS** Du hast ja jetzt schon Rollen angesprochen. Es ist ja ein Scrum-ähnliches. Hast du dann trotzdem diese gleichen Rollen: Scrum Master, Product Owner, das Developmentteam?

**B: FG** Ja die nutzen wir auch.

#### 10.8.1 - Scrum Master sollte im Projekt sein

Dieser Code wird verwendet, wenn der Scrum Master Teil des Projektes sein sollte.

Entstehung des Codes: Induktiv

Beispiel:

Du hast ja jetzt schon Rollen angesprochen. Es ist ja ein Scrum-ähnliches. Hast du dann trotzdem diese gleichen Rollen: Scrum Master, Product Owner, das Developmentteam?

**B: FG** Ja die nutzen wir auch.

#### 10.8.2 - Vorbehalte reine Scrum Master Rolle

Dieser Code wird verwendet, wenn eine Aussage getroffen wird dazu, dass die Rolle des Scrum Masters weniger wichtig im Projekt ist.

Entstehung des Codes: Induktiv

Beispiel:

Derjenige, den ich tatsächlich nicht so sehr im Fokus sehe, wäre die Rolle des Scrum Master.

#### 10.9 - Citizen Developer/ Fachbereichsmitarbeiter

Dieser Code wird verwendet, wenn das Vorhandensein der Citizen Developer und Fachbereichsmitarbeiter im Team befürwortet wird.

Entstehung des Code: Induktiv

Beispiel:

Und ansonsten vor allem die Leute, die es umsetzen sollen. Die Citizen, die Developer.

#### 10.8.1 - Vorbehalte Citizen Developer

Dieser Code wird verwendet, wenn es Vorbehalte gegenüber dem Einsatz von Citizen Developern im Einführungsteam gibt.

Entstehung des Codes: Induktiv

Beispiel:

Für die Einführung der Lösung im Unternehmen braucht es ganz dringend nicht nur den Citizen Developer, sondern hier ist es ganz wichtig, dass wir diese Restriktion, Konventionen im Blick behalten, das heißt, dass diese Governancebrille aufgenommen wird, damit eben nicht diese Nachteile und Schwächen, die wir vorhin schon mal diskutiert haben, eintreten. Insofern halte ich es für essenziell, ich würde aber auch gleichzeitig behaupten, dass der Citizen Developer diese Notwendigkeit nicht sieht und es im Gegenteil als einen Knüppel zwischen den Beinen empfindet.

## 11 - Wichtige Stakeholder außerhalb des Projekts

Welche wichtigen Stakeholder im Unternehmen gibt es außerhalb des Projekts?

Entstehung des Codes: Deduktiv

### 11.1 - Organisatorische Vorgesetzte

Dieser Code wird verwendet, wenn eine Aussage erfolgt dazu, dass organisatorische Vorgesetzte gebraucht werden.

Entstehung des Codes: Induktiv

Beispiel:

Und wenn es dann schwierig wird, muss vielleicht mal so eine Bereichsleitung oder so was mit reingeholt werden, als wichtige Stakeholder, weil, die müssen natürlich Ressourcen geben. Also die Leute, die da aus dem Business kommen, die müssen da Zeit dafür haben.

### 11.2 - Sonstige Beteiligte

Dieser Code wird verwendet, wenn es Stakeholder gibt, die nicht originär zum Unternehmen gehören, oder eine sehr spezielle Sonderrolle haben.

Entstehung des Codes: Induktiv

Beispiel:

Ich überlege gerade, wo vielleicht dann noch Dinge sind, wenn Sie ein Unternehmen haben, das selber keine wirkliche IT oder keine große IT hat, dann brauche ich natürlich so was wie den Provider oder so was mit drin, also ist das natürlich ein wichtiger Stakeholder, der darin eine Rolle spielt. Das ist aber nur eine andere Version von IT, die ich dann im Projektteam habe.

### 11.3 - Geschäftsleitung

Dieser Code wird verwendet, wenn die Geschäftsführung als wichtiger Stakeholder außerhalb des Projekts gesehen wird.

Entstehung des Codes: Induktiv

Beispiel:

Natürlich sollte so was wie die Geschäftsführung informiert darüber sein, weil wie gesagt, das ist durchaus eine Änderung der Arbeitsweise im Unternehmen.

### 11.4 - Abteilungen

Dieser Code wird verwendet, wenn eine Aussage zu ganzen Abteilungen erfolgt, wo der Stakeholder nicht allein der Abteilungsleiter sein kann.

Entstehung des Codes: Induktiv

Beispiel:

Natürlich muss ich eventuell, das fällt mir gerade ein, das wäre vielleicht noch so ein Stakeholder, wenn man so an Personalabteilungen denkt. Ich gebe natürlich Leuten Verantwortungsbereiche, also manche Unternehmen haben ja so Personalrollenbeschreibungen, die sehr strikt sind und da könnte das auch ein Punkt sein.

### 11.5 - Mitarbeitervertretung

Dieser Code wird verwendet, wenn der Betriebsrat als wichtiger Stakeholder außerhalb des Projektes definiert wird. Dabei ist nicht wichtig, ob die Option dabei besteht das dieser auch im Projekt sein könnte.

Entstehung des Codes: Induktiv

Beispiel:

Ich muss meinen Betriebsrat mit abholen, der in den meisten Projekten klassischerweise nicht beteiligt ist, aber dadurch, dass jetzt meine Plattform prinzipiell jeden Mitarbeiter betrifft, muss der Betriebsrat halt involviert sein, muss die Entscheidungen auch mitgehen, muss sich auch darauf einstellen, was heißt das denn im Sinne von: Da kommen jetzt Änderungen auf uns zu. Sachen, die wir vielleicht in der klassischen IT-Entwicklung gemacht haben, funktionieren jetzt mit einer No Code/ Low Code-Plattform ganz anders, wie das Beispiel was ich vorhin gebracht habe mit Unternehmen, die sagen: "Jede Applikation, die ausgerollt wird, muss durch den Betriebsrat genehmigt werden." Ist für eine No Code/ Low Code-Plattform mit Citizen Developern ein Todesstoß an der Stelle, wenn ein Betriebsrat da nicht sauber abgeholt wird, weil, das kann nicht funktionieren.

### 11.6 - Alle Mitarbeiter im Unternehmen

Dieser Code wird verwendet, wenn alle Mitarbeiter im Unternehmen als wichtige Stakeholder außerhalb des Projekts angesehen werden.

Entstehung des Codes: Induktiv

Beispiel:

Ja, natürlich. Also, informiert halten würde ich so viele Leute wie möglich.

### 11.7 - Support Abteilungen

Dieser Code wird verwendet, wenn Aussagen getroffen werden bzgl. wichtiger Stakeholder, die nach der Einführung für das Projekt weiter Zuständigkeiten haben und in irgendeiner Form damit weiter arbeiten..

Entstehung des Code: Induktiv

Beispiel:

Das sollte man auf jeden Fall in der Projektphase schon anfangen und alle die, die dann später involviert sind, sei es ein User Help Desk, diese Software geht kaputt und irgendjemand möchte jemand qualifiziert anrufen, der ihm dann hilft. Dann muss dieses User Help Desk informiert werden. Also es ist von Unternehmen zu Unternehmen und der Größe unterschiedlich und dem, was man macht. Ganz wichtig, aus meiner Sicht, in der Projektphase schonmal sich Gedanken machen, wie es später in der Business as Usual- Phase aussieht. Wer dann für Störungen verantwortlich ist? Wie lange die Reparaturfristen sind? Ist das vertraglich abgesichert? Dafür die richtigen Stakeholder herauszufinden, das ist schon ganz wichtig.

## 12 - Skalierung Pilotprojekt

Wie wird das Projekt skaliert?

Entstehung des Codes: Deduktiv

### 12.1 - Sequenzielles Vorgehen

Dieser Code wird verwendet, wenn bei der Skalierung des Projektes ein sequenzielles Vorgehen bevorzugt wird.

Entstehung des Codes: Induktiv

Beispiel:

B: Schrittweise.

I: Heißt Abteilung für Abteilung für Abteilung, oder?

B: Tatsächlich, also mal gucken, weitere Anforderungen sammeln. Weitere Anforderungen sammeln, weitere Möglichkeiten sammeln, wo gibt es Bedarf oder wo gibt es Möglichkeiten, das, was wir jetzt gelernt haben, weiter zu treiben, Effizienzgewinne herauszuholen und solche Dinge, wo gibt es Wünsche? Aber tatsächlich schrittweise.

### **12.2 - Paralleles Vorgehen**

Dieser Code wird verwendet, wenn bei der Skalierung ein paralleles Vorgehen aufgezeigt wird.

Entstehung des Codes: Induktiv

Beispiel:

eine zweite Vorgehensweise, mit einem Piloten anzufangen und zu gucken, wenn der funktioniert, dann stellen wir es flächendeckend erst mal allen Abteilungen zur Verfügung. Das wird man aber nach meinem Dafürhalten tatsächlich nicht so betreuen können, dass die Leute wirklich anfangen damit zu arbeiten. Es sei denn, es gibt wirklich Schmerzen in den Abteilungen, wo die sagen: "Die können wir nur damit lösen." Dass die sozusagen freiwillig darin sind. Sonst wird das nach meiner Erfahrung eher so enden, wenn ich sofort in die Breite gehe: "Ja hier hat sich schon wieder mal einer ein neues Tool überlegt, aber eigentlich ich komme mit meinen Excel-Dateien eigentlich ganz gut klar. Warum soll ich mich jetzt damit beschäftigen."

### **12.3 - Sequenziell-Paralleles Modell**

Dieser Code wird verwendet, wenn sowohl ein sequenzielles als auch ein paralleles Vorgehen bei der Skalierung empfohlen werden.

Entstehung des Codes: Induktiv

Beispiel:

Ich würde mir sozusagen die nächsten Fachabteilungen nehmen, die in dieses ganze Projekt oder über dieses Projekt begleiten, die mit den Themen vertraut machen und sie schlussendlich onboarden, also sie in die Lage versetzen, dass sie selber die Low Code/ No Code-Plattform nutzen können. Und da auch wieder der Rückschluss zu dem agilen Arbeiten, also aus dieser ersten Iteration lernen und die nächste Iteration mit der nächsten Fachabteilung dann besser machen und das wäre dann so eine Art Inkubator, wo man sagen kann, man lässt die Fachabteilung nacheinander dieses Programm durchlaufen. Danach sind die Fachabteilungen in der Lage, selber diese Plattformen zu nutzen. Sie wissen, welche Governance, also welche Regeln, sie einhalten müssen, welche Verantwortlichkeiten es für das Thema gibt, sie wissen, wohin Sie sich mit Fragen wenden müssen, nämlich an die Community.

### **12.4 - Vorbereitende Maßnahmen**

Dieser Code wird verwendet, wenn Aussagen darüber getroffen werden, wie eine Skalierung vorbereitet werden könnte.

Entstehung des Codes: Induktiv

Beispiel:

Da gibt's verschiedene Varianten und Lösungen, was durchaus auch immer ein Ansatz bei uns ist, um diesen Wissenstransfer, also gar nicht mal die Skalierung der Lösung oder die



Diffusion in die verschiedenen Bereiche, sondern mehr so der Wissenstransfer, dass wir so ein Train-The-Trainer-Konzept versuchen herbeizuführen. Das heißt, dass wir sagen, wir versuchen nicht nur die Leute zu befähigen, es selber umzusetzen, sondern darüber hinaus die auch noch zu befähigen, andere Leute zu befähigen. Einfach um da quasi die Abhängigkeit zu verringern, in dem wir quasi sagen, Train-The-Trainer und der soll dann bitte auch die Leute befähigen. Dass wir so einen Multiplikations- und Skalierungseffekt dadurch erzielen.

### **13 - Befähigung Mitarbeiter**

Wie werden die Mitarbeiter für NCLC-Plattformen befähigt?

Entstehung des Codes: Deduktiv

#### **13.1 - Befähigung durch Schulung/ Workshop**

Dieser Code wird verwendet, wenn in Zusammenhang der Mitarbeiter über Befähigung von Schulungen/ Workshops gesprochen wird.

Entstehung des Codes: Induktiv

Beispiel:

Und sonst, wie gesagt, Communitys, ganz klassische Trainings und was man sich auch anschauen sollte, das klingt jetzt wieder aber YouTube- Videos, also es gibt ein paar Koryphäen für einzelne Plattformen und wenn die erzählen, was möglich ist, dann muss ich persönlich sagen, da bin ich mit einem 5-Minuten-Video schneller, als mich einmal selbst durchzuprobieren und das auch natürlich zu kommunizieren und sagen, das sind so die best of links.

#### **13.2 - Befähigung durch Community**

Dieser Code wird verwendet, wenn in Zusammenhang mit der Befähigung von Mitarbeitern eine Befähigung durch Communities angestrebt wird.

Entstehung des Codes: Induktiv

Beispiel:

Die Communitys sind insofern wichtig, weil sie die erste Anlaufstelle für die User sein sollten. Dass eben ein User, der eine Frage hat oder Probleme hat, nicht als erstes zum Telefon greift und den internen Support anruft, die ihm im Zweifelsfall sowieso nicht helfen können, sondern dass es wirklich interne Communitys gibt oder externe Communitys auch, in denen dann Fragen beantwortet werden können, von einem Citizen Developer zum anderen.

#### **13.3 - Befähigung durch Multiplikatoren**

Dieser Code wird verwendet, wenn die Befähigung der Mitarbeiter für die Plattform durch Multiplikatoren erfolgt.

Entstehung des Codes: Induktiv

Beispiel:

Man kann verschiedene Dinge machen. Also man kann tatsächlich intern, wie ich das eben schon mal kurz gesagt habe, so eine Art Multiplikatoren aufbauen. Also manchmal kommt man in der Plattform auch ohne explizite Schulung aus, also nach der Einführung, wenn genug Leute im Unternehmen schon damit gearbeitet haben, kann sich das Know-how tatsächlich auch schon so verbreiten, da muss man nicht unbedingt zwingend eine Schulung machen. Ich würde trotzdem immer noch Schulungen für komplexere oder für so einen Gesamtüberblick immer noch anbieten.

### **13.4 - Befähigung durch selbst lernen**

Dieser Code wird verwendet, wenn in Zusammenhang mit der Befähigung von Mitarbeitern über eine Befähigung durch eigenständiges lernen gesprochen wird.

Entstehung des Codes: Induktiv

Beispiel:

Also von daher denke ich, ist es wichtig, dass es wenig, also dass es keine Restriktionen gibt, das Ding zu nutzen an sich, natürlich Governance Restriktionen, was man machen darf, was man nicht machen darf, aber dass es erst mal zugänglich für alle verfügbar ist und dann eher über so eine Experimentierphase geht, wo die Leute das möchte dann einfach selber ausprobieren können.

### **14 - Qualitätssicherung Projekt**

Wie erfolgt die Qualitätssicherung des Projektes während des Verlaufs?

Entstehung des Codes: Deduktiv

#### **14.1 - Qualitätssicherung durch emotionale Aspekte**

Dieser Code wird verwendet, wenn eine Aussage zur Qualitätssicherung des Projektes die Position vertritt, dass emotionale Faktoren herangezogen werden sollten.

Entstehung des Codes: Induktiv

Beispiel:

Gute Frage. Motivation, Motivation, Motivation. Wie mache ich das am besten.

#### **14.2 - Qualitätssicherung durch Monitoring**

Dieser Code wird vergeben, wenn eine Aussage in Bezug auf die Qualitätssicherung im Projekt den Punkt Monitoring betrachtet.

Entstehung des Codes: Induktiv

Beispiel:

Ist jetzt auch unabhängig von Power Platform-Projekten, würde ich sagen. Wir haben regelmäßige Statuschecks. Wir haben auch die entsprechenden Quality Gates, wo wir sagen: "Ok, das und das sind jetzt eigentlich die Punkte, die wir adressiert haben sollen."

#### **14.3 - Qualitätssicherung durch Feedback**

Dieser Code wird verwendet, wenn Feedback bei der Qualitätssicherung des Projektes eine Rolle spielt.

Entstehung des Codes: Induktiv

Beispiel:

Regelmäßige Feedbackrunden mit dem Kunden: Ist das jetzt auch verstanden? Sind die entsprechenden Arbeitspakete auch implementiert? Haben wir unsere Checkliste, die wir am Anfang setzen, haben wir die tatsächlich abgearbeitet? Sehen wir, dass der Kunde jetzt in der Lage ist, diese Plattform selber zu betreiben? Oder sehen wir, die haben zwar in den Workshops mit dringesessen aber sich nur, wie soll ich sagen, bespaßen lassen, aber es nicht wirklich verinnerlicht, worauf es tatsächlich ankommt.

## 15 - Erfolgsmessung Projekt

Wie wird der Erfolg des Projekts gemessen?

Entstehung des Codes: Deduktiv

### 15.1 - Quantitative Größen

Dieser Code wird verwendet, wenn die Aussage sich auf Erfolgsmessung bezieht und diese über quantitative Werte ermitteln möchte

Entstehung des Codes: Induktiv

Beispiel:

Wann ist ein Projekt denn erfolgreich? Das ist ja eigentlich im Sinne so: in Time, in Budget, in Scope, hat man ja immer so schön gesagt, im klassischen. Zu Projektabschluss, kann man sagen, hat man all das erreicht, was man in seinem Backlog drin hatte, zu einem gewissen definierten Zeitpunkt, dann ist das abgeschlossen worden und dann bin ich auch, das muss ganz klar sagen, in dem Kostenpart drin.

### 15.2 - Qualitative Größen

Dieser Code wird genutzt, wenn die Aussage sich um Erfolgsmessung dreht und über qualitative Arten der Messung spricht.

Entstehung des Codes: Induktiv

Beispiel:

Aber, dann auch tatsächlich, dann gehts dann eher in die qualitativen Ziele, dass wirklich auch sich das weiterentwickelt, dass Anwendungen mit immer größerem Impact auch auf die Plattform gehoben werden

## Anlagen 12, Summery Grid

Code	Codierte Segmente	Summary
1 - Vorbereitung Unternehmen\1.1 - Auswahl Citizen Developer	<p>Das ist im Gegensatz zu der Idealform, wie es eigentlich gedacht ist, nicht unbedingt der normale Geschäftsanwender. Das sind schon nach meiner Erfahrung im Unternehmen eher die Leute, die eh schon so ein bisschen technisch orientiert sind. Also halt die Leute, die früher auch Makros in Excel geschrieben haben oder so was. Die gehen eher in die Richtung Citizen Development und manchmal sind es auch wirklich IT-Leute, die nicht wirklich Anwendungsentwickler sind und das jetzt nutzen, um ihre eigenen Prozesse zu verbessern oder zu automatisieren. Also ich mache viel in der Prozessautomatisierung damit, das kann ich schonmal sagen</p> <p>EB01_Eckhard Hauenherm: 11 - 11 (0)</p> <p>Die müssen, sage ich mal, also ich muss immer so ein bisschen aufpassen. Wie gesagt, ich mache sehr viel mit Prozessen und Prozessautomatisierung, deswegen denke ich immer so in Prozesslogik. Also ein Verständnis für Prozesslogik, was halt ein guter Applikationsentwickler auch hat. Also wann brauche ich eine Schleife? Was bilde ich über Verzweigungen ab? Wie gehe ich mit Variablen um? Das sind ja so Dinge, soviel technisches Verständnis muss tatsächlich da sein, um wirklich gute, sagen wir mal, gute Anwendungen oder gute Umsetzungen in Low Code/ No Code machen zu können. Also ich muss einerseits ein gutes Verständnis davon haben, wie arbeitet eigentlich eine Applikation, also so ein gewisses technisches Grund-Know-how. Das ist logisches Verständnis. Eigentlich ist es, ich würde mal sagen, so ein klassisches mechanisches Verständnis. Wer mir erklären kann, wie ein Fahrrad funktioniert und warum unterschiedliche Übersetzungen unterschiedliche Geschwindigkeiten bewirken,</p>	<p>Citizen Developer sind in der Praxis eher MA die Technik affin sind oder MA aus der IT-Abteilung, welche keine Anwendungsentwickler sind und NCLC nutzen, um eigene kleine Anwendungen zu erstellen oder ihre eigenen Prozesse zu automatisieren.</p> <p>Ein Citizen Developer sollte die Fähigkeit haben in Prozesslogiken zu denken und ein gewisses technisches Verständnis besitzen und logisch aufbauende Funktionalitäten nachvollziehen können.</p> <p>Wissen über die Plattform ist sehr wünschenswert, da eine Low Code-Plattform nicht ohne gewisse Code-Kenntnisse auskommt, dazu kommt ein mathematisches Verständnis, um gewisse Berechnungen zu verstehen.</p> <p>EB01_Eckhard Hauenherm</p>

der kann in der Regel auch einen Prozess ganz gut darstellen, sage ich mal. Das ist das eine. Also wirklich, das müssen Leute sein, die halt das, was sie machen, auch tatsächlich in solche Prozesse umsetzen können. Das ist eine Umsetzungsleistung, die erst mal in der Ausbildung vielleicht mit drin sein muss oder ähnlichen Dingen. Das Zweite ist natürlich, was gebraucht wird, ein Wissen über die Plattform selber. Das wird häufig auch unterschätzt. No Code/ Low Code, das sind ja auch nicht zwingend simple Plattformen. Also wenn ich an die Umgebung, in der ich am meisten arbeite, die Power Platform, wenn ich da in die Formeln und solche Dinge, Azure, Functions und so was reingucke, dann ist da ja schon noch Code drin. Also ich muss schon so mathematische Klammerung verstehen, Funktionsaufrufe verstehen und solche Sachen. Also so was kommt auch noch mit rein und überhaupt, was ich in der Plattform machen kann.

EB01\_Eckhard Hauenherm: 23 - 23 (0)

An sich gibts meiner Meinung nach, eigentlich fast gar nicht oder ist ein irreführender Begriff, weil damit von vielen verbunden wird, dass ich dann keine IT-Kenntnisse haben müsste, das ist aber aus meiner Sicht komplett falsch, sondern ich brauche trotzdem fundierte Kenntnisse, wie ich eine Datenstruktur mir zum Beispiel erstelle oder so was und meistens, also ich habe jetzt noch keine App geschafft, die produktiv im Einsatz ist, wo ich wirklich keine Zeile Code für geschrieben habe. No Code von daher gibts für mich nicht und Low Code dann eher, dass das Interface schnell und einfach zusammengeklickt wird, ohne dass ich ein grafisches Interface habe und eben nicht in Texteditor-Form.

EB02\_Rosengrün, Robin: 7 - 7 (0)

Also nur um das klarzustellen. Citizen Developer sind für dich Leute, die quasi eigentlich keine IT-Kenntnisse haben, die aber dazu befähigt werden, also mit solchen Plattformen umzugehen oder wie

Citizen Developer benötigen gewisse IT-Kenntnisse, wie zum Beispiel eine Datenstruktur funktioniert.

Wichtig ist auch dass sie Fachkenntnisse aus ihrer Fachabteilung mitbringen, um Anwendungen spezifisch für diese entwickeln zu können.

EB02\_Rosengrün, Robin

würdest du Citizen Developer mal mit 2 Sätzen definieren?

**B: RR** Nicht keine IT-Erkenntnisse, sondern also, als Grunddefinition würde ich sagen, Leute, die nicht in der IT-Abteilung sitzen, sondern aus der Fachabteilung raus selber Anwendungen entwickeln.

EB02\_Rosengrün, Robin: 10 - 11 (0)

Leute, die sich mit PowerPoint auskennen, Leute, die sich mit Excel auskennen, die dann tatsächlich mit den entsprechenden Editoren oder Wizards sich einfache Applikationen, einfache Automationen, zusammenklicken können, ohne eine Zeile Code schreiben zu müssen.

EB03\_Fabian Gackstatter: 7 - 7 (0)

Citizen Developer kann potenziell jeder Mitarbeiter werden, weil jeder Mitarbeiter, wenn ich ihm die richtigen No Code/ Low Code-Tools an die Hand gebe, kann er damit seine eigenen Applikationen bauen. Seine eigenen Flows bauen, eigene Automationen, ohne sich jetzt großartig in Entwicklungsthemen einarbeiten zu müssen.

EB03\_Fabian Gackstatter: 9 - 9 (0)

Also vielleicht noch mal ganz kurz mein Verständnis von Citizen Development, das beschreibt einfach, dass die Entwicklung der erforderlichen Anwendungen dort durchgeführt wird, durchgeführt werden kann, mit Hilfe von z. B. No Code/ Low Code-Lösungen, wo die Probleme tatsächlich anfallen, also der Begriff, also quasi auf Basis der Bürger und nicht auf Basis irgendeiner höher geschalteten Autorität oder höher geschalteten Instanz, die diese Applikationen umsetzt, sondern das ist tatsächlich Bottom up von den Bürgern oder, im unternehmerischen Kontext, von den fachlichen Mitarbeitern selber entwickeln kann.

**I: RS** Meinen Sie mit fachliche Mitarbeiter IT-Mitarbeiter?

Citizen Developer sollten Leute sein, welche sich mit tiefergehenden Funktionen von Anwendungssoftware, wie zum Beispiel Excel, beschäftigen und diese auch nutzen. Ansonsten sieht er keine Einschränkungen und ist der Meinung es kann potentiell jeder Mitarbeiter werden.

EB03\_Fabian Gackstatter

Citizen Developer sind für ihn explizit nicht die IT-Mitarbeiter, sondern Mitarbeiter aus den jeweiligen Fachabteilungen.

EB06\_Hesse, Moritz

	<p><b>B: MH</b> Nein, sondern ganz im Gegenteil die fachlichen Mitarbeiter aus den Fachabteilungen, aus den IT-fremden Abteilungen, also aus dem Einkauf, Warenwirtschaft, Materialhaltung. Also die IT-fachfremden Abteilungen. So dass nicht die IT dafür verantwortlich sein muss, die Applikation zu entwickeln, sondern die Anwender selber.</p> <p>EB06_Hesse, Moritz: 13 - 15 (0)</p>	
	<p>Ja, das geht ja stark mit dem Thema Low Code/ No Code einher. Citizen Development ist dabei aus meiner Sicht die oder der Ansatz, dass eben Nicht-Entwickler in die Lage versetzt werden, selber Applikationen und Workflows zu bauen, also in diese Digitalisierung reinzugehen und da geht es dann natürlich um Themen, wie Befähigung der Nutzer, die bisher keine Entwicklererfahrungen haben. Und typischerweise diese Citizen Developer, das sind typischerweise die Kollegen aus IT-fernen Abteilungen, zum Beispiel Fachabteilungen, irgendwie eine Einkaufsabteilung oder so was.</p> <p>EB07_Janke, Tino: 11 - 11 (0)</p>	<p>Citizen Developer sind die MA aus der Fachabteilung dich nicht Teil der IT-Abteilung sind.</p> <p>EB07_Janke, Tino</p>
	<p>Ja, es werden nicht alle Mitarbeiter auf die No Code/ Low Code-Plattform befähigt. Also das hieße, man würde einer Fachabteilung noch mal eine zusätzliche Aufgabe aufoktruieren. Das ist zu viel. Es muss ja auch nicht jeder in der Fachabteilung ein Excelspezialist sein. Dass wir alle rudimentär mit Excel umgehen können oder zumindest mit einer Tabellenkalkulation, um es neutral auszudrücken. Es gibt ja eins, zwei, drei und wie sie alle heißen. Das ist selbstverständlich, gerade auch in Bereichen, wenn du mit Finanzen zu tun hast, dann ist das, glaube ich, deren tägliches Handwerkszeug. Und die sind auch schon Experten, aber nicht alle in den Fachbereichen müssen das tun. Und so ist es auch hier.</p> <p>EB08_Fey, Peter: 51 - 51 (0)</p>	<p>Er betont, dass nicht jeder Mitarbeiter in der Abteilung auf die NCLC-Plattform befähigt werden muss, da dies ein zu großer Aufwand wäre und MA abschrecken kann, die das gar nicht wollen. Er vertritt den Standpunkt, es sollten die MA sein, welche sowieso tiefgreifend mit Anwendungsprogrammen arbeiten, wie Excel.</p> <p>EB08_Fey, Peter</p>
<p>1 - Vorbereitung Unternehmen\1.2 - Motivation Stakeholder</p>	<p>Tatsächlich ist meine Erfahrung aus den Unternehmen, dass Low Code/ No Code schwerpunktmäßig in der IT eingesetzt wird und weniger in den Fachbereichen, tatsächlich, also das, was man eigentlich will, da sind wir, glaube ich, noch nicht.</p>	<p>Eine grundsätzliche Hürde ist es, die Leute für neue Vorgehensweisen zu begeistern und diese im Vorfeld des Projekts zum Mitmachen zu bringen. Das ist eine Herausforderung. Erfahrungsbeispiele zeigen, dass auch ein</p>

Das hat wahrscheinlich auch was mit einem Rollenverständnis zu tun, denn das klassische Rollenverständnis ist so, für Applikationen ist die IT zuständig und deswegen machen die das auch. Es ist natürlich auch ein bisschen so, dass ich meine Anwender dann sozusagen dahin bringen muss

EB01\_Eckhard Hauenherm: 17 - 17 (0)

Eine grundsätzliche Hürde ist aber natürlich, die Leute zu finden, die mitmachen. Die Zeit haben, die mitmachen, die Spaß daran haben, die Interesse daran haben. Ich weiß aber gar nicht, ob das nur bei einem agilen Projekt so wäre. Ich bin wahrscheinlich näher im Team und deswegen kommt gerade das da eher vor, weil bei einem klassischen Projekt bestimme ich das Team und dann müssen sich halt alle darum kümmern, so ungefähr. Könnte man eventuell als Hürde in der Vorgehensweise festhalten.

EB01\_Eckhard Hauenherm: 49 - 49 (0)

Falsche Erwartungen ist ein Problem, ein Risiko, das einfach nicht, also es gibt ja klassischerweise den Anforderer dann an Problemen, ist jetzt vielleicht auch der Chef irgendwie. Der hat dann von der Low Code-Plattform gehört und verdonnert seinen Mitarbeiter, damit was zu machen und hat einfach falsche Erwartungen, was man damit machen kann und was auch ein Mitarbeiter, der neu in so einer Rolle ist oder so etwas zum ersten Mal macht, was er damit bewirken kann.

EB02\_Rosengrün, Robin: 25 - 25 (0)

Also für uns war Erfolgsfaktor, glaube ich, dass wir, hatte ich vorher glaube ich auch schon gesagt, dass wir von allen Seiten kommen, also dass wir jemand haben, der auch von den Citizen Developern kommt, dass es nicht ein Projekt ist, was für die Leute wie von der IT übergestülpt wirkt. Und vor allem, dass auch von der bei uns, von den Personen gerade von der IT mitgetragen wird, also das es nicht so ein unliebsames Projekt ist, was man halt macht, weil Low Code-Plattformen hip sind, sondern was von

falsches Rollenverständnis schädlich für die Motivation sein kann, sich für das Thema zu begeistern. Wird dieses lediglich als IT-Thema gesehen, kann das die Motivation selbst aktiv zu werden, negativ beeinflussen.

EB01\_Eckhard Hauenherm

Es wird als wichtig empfunden, dass die Mitarbeiter nicht gezwungen werden mit so einer Plattform zu arbeiten und aufgrund falscher Erwartungen schlechte Ergebnisse erzielt werden. Als wichtiger Erfolgsfaktor wird gewertet, wenn alle Mitarbeiter motiviert das Projekt unterstützen und auch die IT-Abteilung entsprechend eingebunden wird. Es sollte nicht der Eindruck entstehen, dass die Mitarbeiter das Projekt aufgezwungen bekommen, aber auch die IT-Abteilung sollte nicht das Gefühl haben, dass sie durch NCLC und Citizen Development ersetzt wird. Es sollte als zusätzliche Bereicherung und Unterstützung für alle Beteiligten betrachtet werden.

EB02\_Rosengrün, Robin



quasi allen Seiten mitgetragen wird und dann sich, da arbeiten wir noch ein bisschen dran, und dann tatsächlich auch in der IT, da ein Bewusstsein verwurzelt ist, dass es kein Gegenmodell zur klassischen IT ist, sondern dass das halt einfach ein weiterer Pfeil im Köcher ist, dass man einfach neue Möglichkeiten dadurch hat. Für alle, von allen Richtungen, für professionelle Entwicklung, für kleinere Entwicklung, für gemeinsame Projekte zwischen IT und Fachbereichen.

EB02\_Rosengrün, Robin: 69 - 69 (0)

das klappt am besten, wenn man selber dahintersteht, also im besten Fall hat der, ich nenne ihn jetzt mal Projektleiter für dieses Thema, eben eine intrinsische Motivation für das Ganze, weil er selber zum Beispiel da ein Leidensthema hatte, das irgendwie eigene Anforderungen von ihm, in einem seiner Projekte bei der IT-Abteilung zu lange lag, oder wie auch immer, aber er muss selber den Mehrwert sehen und das Ganze auch dahin treiben und zum anderen auch die übrigen Projekt-/ Teammitglieder, die müssen eben auch im besten Fall selber darauf Lust haben, selber den Mehrwert erkennen. Und es darf nicht so dieses Gefühl haben, es wird ihnen jetzt übergestülpt, diese Verantwortlichkeit und jetzt müssen sie das das nächste Jahr mal machen und dass es so das ungeliebte Projekt wird in meinem Aufgabenkalender, sondern es muss eben, ich glaube, da muss man jemanden finden, der einfach Lust auf das Thema hat und darum mitzieht. Genau, das wären so die 2 wesentlichen Rollen. Augenmerk ist ganz klar auf dem Projektleiter, der das Thema einfach treibt und vorantreibt und sich Unterstützung in der Organisation dafür sucht.

EB07\_Janke, Tino: 39 - 39 (0)

Ich glaube, es wurde alles gefragt, beziehungsweise würde meinen, es ist irgendwie alles untergebracht. Wichtig ist mir nur, in diesem ganzen Projekt, sind zum einen die Menschen, also die Mitarbeiter, die Kunden, das ganze Thema muss

Eine gute Motivationsunterstützung für das Einführungsprojekt ist, wenn der Projektleiter oder andere Projektmitglieder eine persönliche Erfahrung haben, bei der sich zum Beispiel über lange Wartezeiten in der IT geärgert wurde. Diese zusätzliche Motivation, dass solche Probleme nicht mehr auftreten und ein tatsächlicher Mehrwert für einen persönlich geschaffen wird, sollte Vorteilhaft für eine bessere Motivation wirken. Im Weiteren wird hervorgehoben, dass das Projekt nicht aufgezwungen wirken sollte und die Motivation, es zu realisieren und weiterzubetreiben verschwindet. Sollte der Mehrwert nicht erkennbar sein, wird die Befürchtung geäußert, dass das Projekt nicht den gewünschten Erfolg hat. Explizit ist auch wichtig das alle, welche die Plattform nutzen sollen, auch motiviert sind, diese zu nutzen.

EB07\_Janke, Tino

einen Mehrwert für die haben. Ansonsten fliegt es nicht.

EB07\_Janke, Tino: 47 - 47 (0)

Also mir bringt die beste Governance nichts, wenn ich die Menschen nicht mit an Bord hole, in dem Projekt, wenn es also niemand nutzt. Und wie gesagt, mir bringen die begeisterten Nutzer der Plattform nichts, wenn ich das nicht schaffe, in sinnvolle Leitplanken zu geben, diese Projekte.

EB07\_Janke, Tino: 47 - 47 (0)

Und das ist der Bereich von Citizen Development und in diesem Citizen Development sind dann viele, viele Vorgehensweisen, die man halt aus dem agilen Umfeld kennt, aber halt auch aus der Teammotivation heraus kennt, aus der Teamführung, transformationelle Führung zum Beispiel. Die spielen da eine große Rolle und die sollten im Citizen Development entsprechend auch berücksichtigt werden. Zum Beispiel bei Ideation Workshops, aber auch in der Kommunikation von Lösungsvorschlägen, einer Psychological safety, die dann gegeben sein muss. So dass jeder seinen Vorschlag machen kann, ohne dass Gefahr zu laufen, als Nichtfachmann dann diskreditiert zu werden, sondern genau diese Ideen bringen die Abteilungen weiter. Die sozusagen sich in neues Terrain vorwagen und altbekannte Möglichkeiten verlassen, ist halt auch nicht im Rahmen der Low Code/ No Code-Plattformen ein Bestandteil, aber im Citizen Development, als umfassendere Vorgehensmethode, soll es berücksichtigt werden.

EB08\_Fey, Peter: 9 - 9 (0)

Na, ich will einfach nur mal betonen, dass, ähnlich wie beim agilen Vorgehen, auch bei Citizen Development, der Fachbereichsentwicklung, das Mindset im Unternehmen ein ganz wichtiger Erfolgsfaktor ist. Also man kann, sollte wirklich den Mitarbeitern die Freiheit, aber auch das Vertrauen geben, Lösungen zu suchen, Lösungen auch umzusetzen. Sollte in der Lage sein, eine gute Teamstruktur zu schaffen. Sicherheit für alle Beteiligten.

Als ein Motivationsfaktor wird die Sicherheit genannt, dass Vorschläge, die zur Lösung eines Problems unterbreitet werden, nicht abgelehnt werden und schlecht geredet werden, nur weil der Citizen Developer keinen IT-Hintergrund hat. Es sollte unterstützt werden, dass die Citizen Developer sich in neue Themenfelder vorwagen und dort bei der Lösungsfindung unterstützen. Das bedeutet im Umkehrschluss nicht, dass jemand geschont wird, sondern das Kritik auf einer sachlichen und respektvollen Weise erteilt wird und gemeinsam an einer Lösung gearbeitet wird. Dafür ist eine Unternehmenskultur Voraussetzung, die den Citizen Developern gewisse Freiheiten und das Vertrauen gewährt, nach eigenen Lösungen zu suchen. Dafür ist eine entsprechende Teamstruktur sinnvoll.

EB08\_Fey, Peter

Was nicht heißt, ein Raum, in dem gepampert wird. Auch ein Raum, in dem Sicherheit herrscht, da können schon mal klare Worte gesprochen werden, das sollte auch so sein, aber eben halt nicht, dass die von oben herab gesprochen werden, sondern dass man klar in der sachlichen Ebene bleibt. Das sollte man versuchen alles umzusetzen, und wenn es eben geht, als Geschäftsführung, diese Leistungen, die dann on top erbracht werden, honorieren. Jetzt nicht unbedingt mit Geld, sondern halt mit den Möglichkeiten, die man sonst im Unternehmen hat. Auf jeden Fall Vertrauen und einer ganz klaren Akzeptanz gegenüber denjenigen, die das tun und diese Arbeit auf sich nehmen.

EB08\_Fey, Peter: 59 - 59 (0)

1 - Vorbereitung Unternehmen\1.3 - Projektbewusstsein

Naja gefragt, gefragt wurde es schon, aber es gibt tatsächlich einen Punkt, der mir immer wichtig ist bei dem Thema. Geht bewusst an das Thema ran. Also macht daraus ein Projekt. Geht bewusst ran. Bindet die IT mit ein, weil, du stellst deinen Anwendern eine Plattform zur Verfügung, die in der Regel sehr viel kann, aber auch zu sehr viel Chaos führen kann. Wirklich, meistens wird die Mächtigkeit unterschätzt und die Komplexität der Plattform auch unterschätzt. Also entweder stelle ich es einfach zur Verfügung und die Anwender nutzen es nicht, weil sie es einfach nicht kennen. Oder die fangen dann an es zu nutzen, weil ich es zur Verfügung gestellt habe und wissen nicht so richtig, was sie tun, da passieren komische Dinge. Also, wirklich kann ich nur raten, daraus wirklich ein Einführungsprojekt zu machen. Also es wirklich bewusst anzugehen. Klein, iterativ, wirklich auszuprobieren.

EB01\_Eckhard Hauenherm: 79 - 79 (0)

Ich muss mir natürlich die Zeit dafür nehmen. Also das ist kein Selbstläufer, wenn ich jetzt eine No Code/ Low Code-Plattform einführe und der Schwerpunkt liegt jetzt wirklich auf Plattform. Es geht ja jetzt nicht um eine einzelne Applikation, die ich einstelle, wo ich sage: "Ok, ich

Ein sehr wichtiger Aspekt ist, dass für das Thema ein Projektbewusstsein herrscht. Eine solche Plattform kommt mit weitreichenden Tools, die bei falscher Verwendung negative Konsequenzen für das gesamte Unternehmen haben, aber auch einen finanziellen Verlust bedeuten können, wenn die Plattform eingeführt, aber nicht optimal genutzt wird. Aus diesem Grund sollte die Einführung als Einführungsprojekt realisiert werden, mit den entsprechenden zugehörigen Maßnahmen. NCLC ist nichts, was ich nebenbei einführen kann.

EB01\_Eckhard Hauenherm

Bei der Einführung der Plattform darf nicht unterschätzt werden, welche weitreichenden Konsequenzen die Einführung einer NCLC-Plattform hat für das gesamte Unternehmen. Es ist abzugrenzen von Softwareprojekten, bei der ein überschaubarer Personenkreis für

habe jetzt hier das neue System implementiert. Dafür brauche ich jetzt zwei Admins, die sich darum kümmern." Wenn ich eine Plattform zur Verfügung stelle, da sind eben viel breitere Themen dran, um die ich mich kümmern muss. Das heißt, ich brauche eine entsprechend große Admintruppe, die tatsächlich diese ganzen Aktivitäten der User managen kann. Ich brauche aber auch jetzt eine entsprechende Changemanagementorganisation, die tatsächlich damit umgehen kann. Also in den Implementierungen, die wir sehen, ist die technische Zurverfügungstellung der Plattform maximal 20 % vom gesamten Volumen, was so ein Projekt ausmacht und die restlichen 80 % sind tatsächlich, wie bereite ich denn meine Organisation darauf vor, das jetzt tatsächlich umzusetzen, das anzunehmen, die Leute zu trainieren, die Leute aber auch bei der Stange zu halten. Mit vernünftigen Informationen zu füllen, mit den entsprechenden Aktivitäten, wie einem Hackathon, wie mal eine Expertenrunde mit Walk-in-Clinic und allem. Diese Sachen, die ich zur Verfügung stellen muss. Also da muss man als Unternehmen natürlich auch Geld in die Hand nehmen, weil diese Rollen müssen gestacked werden. Das sind in der Regel keine Sachen, die ich zu einem anderen Job nebenher machen kann. Und damit tun wir uns auch am Anfang schwer, wenn ich jetzt so ein Team zusammenstelle auf Kundenseite und sozusagen: "Wen brauch ich denn jetzt? Ok, ich brauche die drei Admins für den technischen Teil und dann brauche ich aber noch die zehn Leute, die sich um das Thema Changemanagement und Service Organisation kümmern." Wo die meisten Kunden sagen: "Ich dachte, das wäre ein IT-Thema." Und das ist so eine Plattformentscheidung halt nicht. Die betrifft im Zweifelsfall das ganze Unternehmen und genau so muss ich die auch aufhängen. Dass es eben nicht mit den drei Admins getan ist, die das jetzt noch nebenhermachen, sondern es ist ja ein organisationsweiter Change, den ich da einführe und den muss ich auch mit der entsprechenden Manpower unterfüttern.

die Bedienung und den Betrieb benötigt wird. Nur ein geringer Teil beim Einführungsprojekt ist tatsächlich technischer Natur, der wesentlich größere Teil ist organisatorischer Natur und beschäftigt sich mit Fragen, wie die Organisation vorbereitet wird, wie Mitarbeiter befähigt werden auf die Plattform, wie eine Akzeptanz gefördert werden kann, wie Informationen verteilt werden, aber auch die Motivation zur Nutzung des Tools hoch zu halten. Wichtig ist auch, dass entsprechende finanzielle und personelle Ressourcen zur Verfügung gestellt werden. Der personelle Aufwand ist groß und kann nicht in einer Nebentätigkeit erfüllt werden. Dies wird als eine große Hürde zu Beginn des Projekts beschrieben, dass wichtige Fragen nicht geklärt sind und Ressourcen nicht zu Verfügung stehen. Es muss klar sein, dass dies nicht nur ein IT-Projekt ist, sondern ein Projekt, in dem viele verschiedene Fachbereiche eines Unternehmens involviert sind. Eine Plattform-Entscheidung ist eine Entscheidung, die Konsequenzen für das ganze Unternehmen hat und dieses sehr stark verändern kann.

EB03\_Fabian Gackstatter

EB03\_Fabian Gackstatter: 23 - 23 (0)

Das ist halt sehr wichtig, dass der Kunde wirklich weiß, worauf er sich denn da einlässt. Wie gesagt, eine No Code/ Low Code-Plattform ist eben nicht nur: Ich führe jetzt ein neues Tool ein ins System, sondern das ist schon eine, das geht schon ziemlich an die Basis von einem Unternehmen und das unterschätzen eben viele Kunden. Und wenn sie sich dann darauf Committed, dass sie wirklich sich darauf fokussieren, wie kriege ich denn jetzt die Technik zum Laufen. Weil es in der Regel von der IT mal vorangetrieben wird, so ein Thema, und dann halt auch so mit Scheuklappen draufguckt wird: Ich muss nur die Technik ans Laufen kriegen. Der Rest passiert dann schon irgendwie. Das ist halt gefährlich, so eine Geschichte. Also da muss ich auch meine Kunden immer wieder drauf hinweisen: Ihr dürft es nicht nur auf die Technik reduzieren. Die Technik macht maximal 25 % aus von der ganzen Geschichte. Der Rest ist, wie eure Unternehmensstruktur dieses Thema absorbieren kann und tatsächlich umsetzen kann. Wenn ihr nur die Technik zum Laufen kriegt, werdet ihr an den anderen Themen scheitern, wenn ihr das so unterschätzt. Das würde ich so mitgeben. Das ist einfach wichtig, so als Best Practice und da ruhig auch drauf rumreiten. So als Mantra: Jungs, es ist nicht nur die Technik! Power Platform betrifft alles. Power Platform rennt Silos ein, weil, ich kann auch nicht mehr alleine für mich alles entscheiden, wenn es zum Beispiel um Thema Policies für Connectoren geht. Mit welchen Datenstrukturen darf ich denn kommunizieren? Betrifft alles, was auf der Plattform stattfindet und eben nicht nur das Team das jetzt Connectoren baut, zum Beispiel und solche Sachen.

EB03\_Fabian Gackstatter: 71 - 71 (0)

1 - Vorbereitung Unternehmen\1.4 - Vorbereitung der Organisation im Unternehmen

Also, dass allgemein ein besseres Verständnis für Daten herrscht in dem Unternehmen, also das wird ja propagiert, seit Jahren schon, in jedem Unternehmen wahrscheinlich, Daten sind das

Innerhalb einer Organisation sollte ein Verständnis für Daten erzeugt werden. Es muss unternehmensweit in jeder Abteilung eine Sensibilisierung entstehen, was Daten sind, wofür sie genutzt

neue Gold. Ich will jetzt gar nicht so sehr in die Richtung blasen und weiß auch nicht, ob das für unser Unternehmen so sehr stimmt, wie das propagiert wird. Aber, dass Prozesse mehr auch von den Daten her gedacht werden und dass da einfach ein Verständnis für herrscht. Also, dass in jeder Abteilung irgendwo das Verständnis für herrscht, was für Daten produzieren wir und wie können wir diese verwenden, welche davon können wir strukturieren, welche lohnt es sich zu strukturieren, um, zum Beispiel, herrscht gar kein Verständnis dafür, dass man mit E-Mail-Strömen aufhören soll. Und vielleicht ist viel besser auf ein Ticketsystem, bei welchem Thema auch immer, umzustellen, weil einfach gar nicht gesehen wird, dass eigentlich immer ein Datenstrom, mit jeder E-Mail enthält die gleichen 10 Informationen, die sich super machen würden in einer in der Datenbank strukturiert, in den Prozess dann auch wieder. Schick ich halt die E-Mail an den nächsten, der muss das bearbeiten, aber die Person, die das geschickt hat, weiß dann gar nicht mehr, wie der Status ist, also das ist so ein Klassiker, den wir an ganz vielen Stellen haben. Weil einfach niemand das Verständnis dafür hat, dass da eigentlich ein Datenstrom abläuft. Dass es eigentlich nichts anderes wäre als der Zeile in der Datenbank und den Prozess für alle Seiten einfach transparenter macht und erst, wenn ich dieses Verständnis habe, dann habe ich ja auch die Möglichkeit, die Low Code Tools einzusetzen. Also das ist so ganz oft auch was, was mir entgegengebracht wird, ich würde das sehr gern verwenden, aber ich weiß nicht wofür. Wenn ich mit Leuten, also egal jeder Abteilung, nach 5 Minuten Gespräch, habe ich in der Regel 10 Ideen, was man damit machen kann, ja und andere Leute aus den Abteilungen rauskommen, die selber auf keine dieser Ideen, sagen dann „Ah ja, das stimmt ja, alles gut“ aber ja, das kommt halt mit einem gewissen Verständnis für Daten/ Datenstrukturen, kommt das dann halt erst mal so was sieht. Das ist für mich auch ein Erfolgsfaktor, dass das funktionieren kann.

werden können, dass viele Informationen mehrfach weitergegeben werden und welche Folgen ein falscher Gebrauch verursachen kann.

Informationen werden immer weitergeschickt, obwohl diese in einer Datenbank, die allen zugänglich ist, wesentlich besser aufgehoben sind. Viele wissen nicht, was sie mit Daten, die sie besitzen, tun sollen und wie sie diese verwenden können, um ihre Arbeit zu erleichtern. Dies führt auch zu dem Problem, das durch dieses fehlende Wissen viele keine Ahnung haben, wie sie NCLC positiv in ihre Arbeitsabläufe integrieren können.

EB02\_Rosengrün, Robin

Dann halt allgemein, das muss von der Führungskraft bis zum letzten Mitarbeiter, muss ich das einfach wahrscheinlich in jedem Unternehmen dieser Welt, außer vielleicht bei Twitter und Facebook, aber muss da einfach die Awareness für Daten geschafft werden

EB02\_Rosengrün, Robin: 63 - 63 (0)

Also Citizen Developer funktionieren eigentlich nur, wenn ich das in einer kontrollierten Offensive beschreibe. Sprich, ich brauche eine solide Grundlage. Ich muss die Plattform sauber definiert haben. Ich muss die richtigen Guidelines und Policies etabliert haben und ich muss aber auch meine Supportorganisation darauf vorbereitet haben und auch meine Changemanagement Organisationen, dass da jetzt auf einmal potenziell 90 % mehr Entwickler im Haus unterwegs sind, mit den entsprechenden Bedürfnissen. Wenn ich das tatsächlich abgedeckt habe, dann kann ich das Risiko durch so einen Citizen Developer auch drastisch reduzieren.

EB03\_Fabian Gackstatter: 13 - 13 (0)

Das ist halt sehr wichtig, dass der Kunde wirklich weiß, worauf er sich denn da einlässt. Wie gesagt, eine No Code/ Low Code-Plattform ist eben nicht nur: Ich führe jetzt ein neues Tool ein ins System, sondern das ist schon eine, das geht schon ziemlich an die Basis von einem Unternehmen und das unterschätzen eben viele Kunden. Und wenn sie sich dann darauf Committed, dass sie wirklich sich darauf fokussieren, wie kriege ich denn jetzt die Technik zum Laufen. Weil es in der Regel von der IT mal vorangetrieben wird, so ein Thema, und dann halt auch so mit Scheuklappen draufguckt wird: Ich muss nur die Technik ans Laufen kriegen. Der Rest passiert dann schon irgendwie. Das ist halt gefährlich, so eine Geschichte. Also da muss ich auch meine Kunden immer wieder drauf hinweisen: Ihr dürft es nicht nur auf die Technik reduzieren. Die Technik macht maximal 25 % aus von der ganzen Geschichte. Der Rest ist, wie eure Unternehmensstruktur dieses Thema

Als wichtiger Aspekt vor der Einführung von NCLC wird genannt, dass ein Bewusstsein dafür herrscht, welche großen organisatorischen Veränderungen durch die Einführung von NCLC entstehen. Dabei sollten Vorbereitungen getroffen werden dafür, dass sich verschiedene innerbetriebliche Guidelines und Policies des Unternehmens verändern werden. Zusätzlich sollte in eine strategische Planung aufgenommen werden, dass neue Strukturen (wie Supportstrukturen für die Citizen Developer) entstehen werden, welche organisatorisch geplant werden sollten. Ein weiterer wichtiger Aspekt ist, dass in eine strategische Planung aufgenommen wird, dass NCLC nicht nur ein System ist, sondern eine Plattformscheidung. Diese muss entsprechend auch als solche bewertet werden und muss mit dem entsprechenden Commitment für eine solche Entscheidung begleitet werden. Für die Planung sollte berücksichtigt werden, dass dies nicht nur technische Veränderungen bedeutet, sondern strukturell weitreichende Änderungen herbeiführen wird. Für diese Änderungen müssen die entsprechenden Vorbereitungen getroffen werden und strategischen Planungen vorgenommen werden.

EB03\_Fabian Gackstatter

absorbieren kann und tatsächlich umsetzen kann. Wenn ihr nur die Technik zum Laufen kriegt, werdet ihr an den anderen Themen scheitern, wenn ihr das so unterschätzt. Das würde ich so mitgeben. Das ist einfach wichtig, so als Best Practices und da ruhig auch drauf rumreiten. So als Mantra: Jungs, es ist nicht nur die Technik! Power Platform betrifft alles. Power Platform rennt Silos ein, weil, ich kann auch nicht mehr alleine für mich alles entscheiden, wenn es zum Beispiel um Thema Policies für Connectoren geht. Mit welchen Datenstrukturen darf ich denn kommunizieren? Betrifft alles, was auf der Plattform stattfindet und eben nicht nur das Team das jetzt Connectoren baut, zum Beispiel und solche Sachen.

EB03\_Fabian Gackstatter: 71 - 71 (0)

Genau. Also es gibt, manche nennen es die IT-Strategie, manche nennen es digitale Transformationsstrategie, und da sind dann halt Punkte drin. Ganz klar, was wir häufig sehen, gibt es eine Cloudstrategie. Möchte man mit Systemen in die Cloud gehen, aber auch möchten wir weiterhin bei einzelnen Häusern, die ich gesehen habe, steht das da explizit drin. Wir verwenden No Code/ Low Code oder man formuliert es anders: "Wir helfen dem Fachbereich, sich selbst zu helfen, ohne Entwickler zu werden." Also das ist schon, dass es dort verankert ist und das ist ja eine Marschroute und da sollte es auf alle Fälle auch verankert sein, dass das Commitment von oben da ist.

EB04\_Göttl, Simone: 25 - 25 (0)

Also ich glaube, was man einfach sich immer überlegen muss, was sind so Vor- und Nachteile von so einer Plattform. Es gibt ja verschiedene Anbieter und die haben verschiedene, ich habe wenig über Anbieter selber gesprochen, da sieht man natürlich auch Präferenzen bei Unternehmen. Der eine sagt, ich finde jetzt Microsoft ganz toll, weil, das kennt ja jeder, weil sich das einbettet in PowerPoint, in Excel. Kommt dann aber der nächste um die Ecke und sagt: "Nee, ich

Eine wichtige organisatorische Entscheidung ist das Thema, in weit langfristige Planungsrahmen mit aufzunehmen sind, wie eine IT-Strategie oder Ähnliches. Um ein entsprechendes Commitment, innerhalb des Unternehmens zu erzeugen, sollte es in so einem strategischen Rahmenwerk fest drin stehen, dass NCLC genutzt wird. Die Formulierung kann umschreibend sein oder explizit sein.

Ein weiterer wichtiger Punkt, der vor der Einführung von NCLC bearbeitet werden sollte, ist die Auswahl der Anbieter. Welcher Anbieter passt für das eigene Unternehmen? Passen mehrere Anbieter für unterschiedliche Abteilungen? Wie funktioniert die Integration mit anderen im Unternehmen vorhanden Systemen? Und wie sieht eine mögliche Exit-Strategie aus, falls die Plattform einmal nicht mehr passen sollte?

EB04\_Göttl, Simone



habe es ein bisschen komplizierter. Ich möchte gerne Pega nutzen." Oder "Nee, ich mag dann doch noch mal Appian, weil das auch mit komplexeren Workflows möglich ist und ich eigentlich ganz andere Sachen damit mache." Also von daher, ich glaube ein wichtiger Punkt ist einfach, wie mache ich so eine Anbieterauswahl und welche Tools, Prozesse, kann ich mir als Unternehmen vorstellen, zukünftig in so einer No Code/ Low Code-Plattform abzubilden. Das ist, glaube ich, noch mal ein wichtiger Punkt, den ich an der Stelle auch noch mal mitgeben würde, weil, die sind schon unterschiedlich.

EB04\_Göttl, Simone: 59 - 59 (0)

Ich kann jetzt nur rein aus der Projektmanagementsicht oder aus meinem täglichen Business reden, da spielt natürlich viel auch eine Rolle von diesen ganzen Hard Facts. Jetzt dieses Kennen von Systemen, das Kennen von Prozessen, Abläufen und der Technik,

EB05\_Rohrbeck, Christian: 77 - 77 (0)

Ansonsten, auch wieder abhängig von der Projektgröße und der Unternehmenskultur, ist es sicherlich hilfreich bei so einem Einführungsprojekt, diesen kulturellen Aspekt auch irgendwie zu begleiten. Ja, also wir gehen ja im Grunde ganz krass, findet ein Wandel statt von: Ich habe als Fachabteilung eine Bittstellung. Die muss ich erst mal irgendwie mühselig an die IT herantragen. Vielleicht muss ich die fachlich erst mal von der Geschäftsführung absegnen lassen, bevor ich damit auf die IT zugehen kann. Versus, ich habe eine Idee, ich log mich eben ins Web ein. Bau mir so ein Ding. 3 Minuten später habe ich das Ding bei mir auf dem Handy und kann damit operativ loslegen. Das ist ja ein unglaublicher kultureller Shift, der hier stattfindet. Ich würde mal behaupten, das ist kein Erfordernis, dass man das begleitet, kulturell.

EB06\_Hesse, Moritz: 71 - 71 (0)

Und ein weiterer Nachteil ist aus meiner Sicht, dass man das, glaub ich,

Es sollte vor der Einführung dafür gesorgt werden, dass die eigenen Prozesse wirklich bekannt sind, wie diese funktionieren und welche Systeme wie im Einsatz sind.

EB05\_Rohrbeck, Christian

Vorbereitung auf kulturelle Veränderungen im Unternehmen und dafür offen sein.

Bereit sein, vorhandene Prozesse zu verändern.

EB06\_Hesse, Moritz

Es wird als wichtig angesehen, dass sich Unternehmen klar machen, dass

organisationsweit nutzen muss, also man muss sich darauf verständigen, in der Organisation, dass es genutzt wird, weil, wenn es nur eine Abteilung gibt, die damit arbeitet, alle anderen arbeiten nicht damit, kann das Thema auch nicht richtig wachsen und skalieren, also dann hab ich einfach diese Vorteile, zum Beispiel einer Community, nicht, die kann sich dann nicht voll entfalten. Das heißt, ich müsste mehr in die Wissensvermittlung investieren, und dann, glaube ich, ist der Nutzen auch nicht so groß, wie wenn man das über die gesamte Organisation macht dieses Thema.

EB07\_Janke, Tino: 15 - 15 (0)

Und das ist, sozusagen, das umfassendere Paket und das ist dann halt eben nicht möglich, indem ich mich vor einen Bildschirm setze und in die Tastatur hake, sondern es geht da darum, dass man, bevor man etwas anfängt, wirklich sich auch schon mal eine Idee davon macht, wer ist eigentlich mein Stakeholder? Wer ist jemand, der partizipiert an dem Programm? Wo sind die Touchpoints, um es ganz neudeutsch wieder auszudrücken? Wie sieht die Customer Journey aus? Oder halt, wenn man es anders ausgedrückt haben möchte, wie sieht die Journey der Fachbereiche aus? Welche Daten werden verarbeitet? Das kann man alles in den No Code/ Low Code-Applikationen abbilden, aber die gedankliche Arbeit, die ist dann vorher zu leisten. Also sich einfach hinzusetzen und zu sagen: "Ich schreibe jetzt mal schnell ein Programm. Ich weiß zwar noch nicht, was ich eigentlich schreiben will." Ist wahrscheinlich genauso erfolgreich wie zu sagen: "Ich lauf jetzt einfach mal durch die Stadt. Weiß jetzt nicht genau, wohin ich eigentliche komme, aber ich habe dann alles gesehen, was wichtig ist." Nein, man muss leider in der Welt oft eben den anderen Weg gehen. Das heißt, man muss eine Primary Research durchführen, Secondary Research und lauter Preperation Research, um dann wirklich sagen zu können: "Das sind die Informationen, die ich brauche. Jetzt habe ich die richtigen Fragen gestellt und

das der bestmögliche Nutzen solcher Plattformen durch das Skalieren in das gesamte Unternehmen entsteht. Es sollte also vor dem Projekt bereits bekannt sein, wie weit die Skalierung gehen soll, um während der Einführung nicht der Problematik gegenüber zu stehen, dass zusätzliche Investitionen notwendig werden, weil doch mehr skaliert werden soll als ursprünglich gedacht.

EB07\_Janke, Tino

Es ist enorm wichtig im Vorfeld der Einführung, sich über das ganze Thema NCLC zu informieren. Es sollten sich, bevor die Einführung stattfindet bereits viele Fragen gestellt werden und nicht erst, während man in der Einführung ist und bestimmte Dinge herausgefunden werden. Learning by Doing wird nicht als akzeptabler Ansatz für die Einführung gesehen.

Es sollten bereits im Vorfeld, während der Recherche zu NCLC, sich Gedanken gemacht werden, wie Informationsmanagement und Cybersecurity-Aspekte von diesen Plattformen beeinflusst werden und wo bestimmte Schranken erstellt werden müssen, damit keine kritischen Fehler passieren, da diese Plattformen ein sehr hohes Maß an Freiheiten gewähren.

EB08\_Fey, Peter

jetzt kann ich zur Lösungsfindung kommen.“ Das ist aber eher Beritt von Citizen Development.

EB08\_Fey, Peter: 11 - 11 (0)

Ja, die Nachteile bestehen halt einfach darin, dass ich Freiheiten habe. Das hört sich jetzt vielleicht ein bisschen seltsam an, aber die Freiheit, die ich habe, die beschränkt natürlich auch die Möglichkeit einer zielgerichteten Kommunikation. Es gibt in der Literatur einige Beispiele dafür, dass gerade die Schnittstellen, die jetzt innerhalb dieser Applikationen aufgebaut werden, im Nachhinein die wirklich komplizierten Fragestellungen sind. Also schnell, sagen wir mal, eine Routine aus einer Perspektive zuschreiben, das ist, glaube ich nicht besonders herausfordernd. Herausfordernd wird es dann, wenn ich eine Information habe und sage, diese Information geht jetzt auch auf diesen Empfänger. Oder jener Empfänger muss mir etwas geben. Das heißt, ich muss ein Datenmodell haben, in dem die Informationen ausgetauscht werden können, und zwar so, Aspekt passive Datensicherheit, dass die Daten korrekt gespeichert werden. Also das da auch keine Datenkorruption stattfindet und man mit den gespeicherten Daten nichts mehr anfangen kann. Das heißt, es muss etwas aus einem Guss sein und das stellt sich nicht heraus, indem ich das praktisch umsetze, sondern indem ich einfach erst mal hingehe und wirklich mir genau überlege, in welchem Umfang muss ich meine Daten aufbereiten. In welchem Umfang möchte ich diese Daten austauschen im Nachhinein und was soll mit den Daten passieren? Was soll aber auch mit der Pragmatik, das heißt, mit der Handlung, passieren, auf die ich die dann aufbaue und sage, in diese und jene Richtung soll dann halt auch in dem Workflow gehandelt werden. Das sind alles Dinge, die muss ich mir vorher überlegen und diese Freiheiten, die ich dann in der Gestaltung habe, die kann ich im Endeffekt nur beherrschen, indem ich wirklich sehr, sehr genau weiß, was ich eigentlich tun will. Also der große Vorteil, ich kann alles tun, ist halt auch im

dialektischen Verhältnis gesehen, auch der große Nachteil. Ich kann halt auch alles falsch tun, ich kann halt auch alles so tun, dass es eben nicht zu einem einzigen Guss wird und darauf kommt es ja an.

EB08\_Fey, Peter: 13 - 13 (0)

1 - Vorbereitung Unternehmen\1.5 - Umgang mit MA als Teil der Unternehmenskultur

und auch das Vertrauen seinen Mitarbeitern gegenüber, weil, ansonsten lasse ich halt viel Potenzial von so einer No Code/ Low Code- Plattform liegen. Wir haben Kunden, die es tatsächlich einführen, aber dann nur mit tatsächlichen Entwicklern bestücken. Das heißt, die ganzen Citizen Developer bleiben nach wie vor außen vor. Was natürlich dann viel Potenzial von so einer Plattform wieder verschenkt. Zum einen dem Misstrauen geschuldet, dass man sagt: "Ich traue das meinen Mitarbeitern einfach nicht zu." Oder: "Wir können die nicht in den entsprechenden Kontrollmechanismus hineinbringen, wie ich es mit meiner zentralen IT machen kann." oder weil man sagt: "Ich habe einfach nicht die Manpower, um das zu betreuen, die Geschichte. Wir haben nicht genügend Leute für Projekte, dann können wir jetzt nicht nebenher noch eine neue Plattform betreiben." Dann lass ich halt viel Potenzial liegen, weil, wie gesagt, es geht nicht darum, ein neues Office Paket einzuführen, was ich mit ein, zwei Mann betreiben kann und tragen, sondern No Code/ Low Code-Plattform heißt halt eben auch, ganz neue Aufgabengebiete, die es so vorher nicht gab

EB03\_Fabian Gackstatter: 23 - 23 (0)

Unternehmen müssen den eigenen Mitarbeitern vertrauen, um das volle Potenzial der Plattform zu entfalten. Es wird nicht nur ein neues Office Paket eingeführt, sondern eine Unternehmensverändernde Plattform.

EB03\_Fabian Gackstatter

aber entscheidend ist tatsächlich und das merke ich immer wieder im täglichen Leben, ist eher schon so die emotionale Seite. Also so diese Softskills, weil die in der Regel das Thema haben, dass die Kollegen für ihre Themen Experten sind. Die wissen genau Bescheid. Es gibt nichts, was ich da irgendjemand sagen kann in der Regel. Das ist auch gar nicht mein Ziel, aber woran das halt scheitert, sind diese rein menschlichen Themen. Da redet A nicht B oder, ein absoluter Klassiker ist: A schreibt B eine E-Mail. B

Es sollte ein gutes Unternehmensklima geschaffen werden, indem die Leute sich miteinander unterhalten, um Problemstellungen zu verhindern, bei denen Informationen oder Prozesse hängen bleiben, aufgrund von mangelndem Feedback.

EB05\_Rohrbeck, Christian

antwortet nicht, weil er denkt: "Ist ja gar nicht mein Thema." Also er gibt einfach kein Feedback oder vergisst es. A wartet, die Zeit verstreicht und so gehen immer wieder ganz viele gute Sachen verschütt und die dann wieder rauszuholen, dass voranzubringen, das ist wirklich schwierig. Weil viele auch ein bisschen auf Zeit spielen, sage ich mal und das ganze Thema kann man, glaube ich, nur dann dem begegnen, wenn man da ein gutes Klima schafft. Ein freundschaftliches Klima, wo die Leute auch aktiv Lust haben, miteinander zu reden, sich auch auszutauschen. Das ist, glaube eine der wichtigsten Aufgaben mit. Dass diese Seite, diese nicht messbare Seite, gut passt und dann könnte es auch ein Selbstläufer werden.

EB05\_Rohrbeck, Christian: 77 - 77 (0)

Also aus meiner Sicht, muss man natürlich immer schauen, dass es richtig in der Organisation verankert ist, aber wenn das einmal der Fall ist, dann kann man schon sagen, dass es zum einen die Innovation innerhalb der Organisation fördert, weil ich natürlich viel mehr Leute in die Lage versetze, dass sie selber einfach mal Ideen ausprobieren können. Und dass sie auch wesentlich schneller sind, als wenn sie sich jetzt mit Programmiersprachen oder so beschäftigen müssten, dass sie dann auch Ergebnisse zeigen können. Das heißt, ich glaube, ich rege die Mitarbeiter dazu an, auch selber mal Produkte zu verbessern, neue Produkte zu kreieren, das ist, glaube ich, für mich ein wesentlicher Faktor, den können wir auch jetzt schon mit Low Code/ No Code-Plattformen erreichen.

EB07\_Janke, Tino: 17 - 17 (0)

einer Psychological safety, die dann gegeben sein muss. So dass jeder seinen Vorschlag machen kann, ohne dass Gefahr zu laufen, als Nichtfachmann dann diskreditiert zu werden, sondern genau diese Ideen bringen die Abteilungen weiter. Die sozusagen sich in neues Terrain vorwagen und altbekannte Möglichkeiten verlassen, ist halt auch nicht im Rahmen der Low Code/ No Code-Plattformen ein

Ein Umfeld schaffen, in dem die MA innovativ sein wollen und sie angeregt werden, neue Applikationen zu kreieren.

EB07\_Janke, Tino

In der Unternehmenskultur sollte ein Denken vorhanden sein, das den Citi-zen Developern ein Gefühl der Sicherheit gibt. Diese Sicherheit hat den Zweck, dass diese sich trauen, neue Vorschläge zu unterbreiten, die nicht zwingend einer traditionellen Vorgehensweise aus der IT-Perspektive entsprechen. Dadurch wird ein besserer Feedback-Mechanismus ermöglicht

Bestandteil, aber im Citizen Development, als umfassendere Vorgehensmethode, soll es berücksichtigt werden.

EB08\_Fey, Peter: 9 - 9 (0)

Na, ich will einfach nur mal betonen, dass, ähnlich wie beim agilen Vorgehen, auch bei Citizen Development, der Fachbereichsentwicklung, das Mindset im Unternehmen ein ganz wichtiger Erfolgsfaktor ist. Also man kann, sollte wirklich den Mitarbeitern die Freiheit, aber auch das Vertrauen geben, Lösungen zu suchen, Lösungen auch umzusetzen. Sollte in der Lage sein, eine gute Teamstruktur zu schaffen. Sicherheit für alle Beteiligten. Was nicht heißt, ein Raum, in dem gepampert wird. Auch ein Raum, in dem Sicherheit herrscht, da können schon mal klare Worte gesprochen werden, das sollte auch so sein, aber eben halt nicht, dass die von oben herab gesprochen werden, sondern dass man klar in der sachlichen Ebene bleibt. Das sollte man versuchen alles umzusetzen, und wenn es eben geht, als Geschäftsführung, diese Leistungen, die dann on top erbracht werden, honorieren. Jetzt nicht unbedingt mit Geld, sondern halt mit den Möglichkeiten, die man sonst im Unternehmen hat. Auf jeden Fall Vertrauen und einer ganz klaren Akzeptanz gegenüber denjenigen, die das tun und diese Arbeit auf sich nehmen.

EB08\_Fey, Peter: 59 - 59 (0)

und es kann ein innovatives Lösungsvorgehen etabliert werden. Dabei müssen auch falsche Aussagen honoriert werden und nicht abgeschmettert, um ein Vertrauen bei den Citizen Developern zu etablieren.

EB08\_Fey, Peter

1 - Vorbereitung Unternehmen\1.6 - Vorteile NCLC Plattformen

Vorteile, die Unternehmen wissen sollten, bevor Sie NCLC einführen, sind: Kein Kommunikationsproblem zwischen den Entwicklern und Anfordernern, da der Anforderer und der Entwickler die gleiche Person sind. Es können einfache Prozesse sehr einfach automatisiert werden. Beispielsweise Rechnungseingang, Geschäftsprozesse etc. Dabei ist es realistisch, dass kleine Applikationen auch genutzt werden, um Daten zu pflegen. Ein weiterer Vorteil ist die hohe Geschwindigkeit, durch die man innerhalb kurzer Zeit schnell gute Ergebnisse erzielt.

EB01\_Eckhard Hauenherm

Ein wichtiger Vorteil von NCLC gegenüber klassischen IT-Projekten ist die höhere Geschwindigkeit. Durch den erhöhten initialen Aufwand und die potentiellen Probleme durch die Übersetzung der Fachanforderung in IT-Anforderungen kann es zu erheblichen Verzögerungen kommen. Bei NCLC wird direkt begonnen mit der Entwicklung, es sind keine Anforderungsübersetzungen notwendig, da Entwickler und Anforderer dieselbe Person sind bzw. Anforderungen sind für Entwickler verständlich, da es dieselbe Fachabteilung betrifft. Es kann schnell MVP gezeigt werden und Feedback eingeholt werden. Dies löst das Kommunikationsproblem, Kommunikationsprozesse werden vereinfacht bzw. weitreichend automatisiert. Mit solchen Prozessen sind Prozesse gemeint, wo Dokumente via E-Mail zum Ausfüllen hin und her geschickt werden oder einfach eine E-Mail, die von MA zu MA weitergeleitet wird und jeder schreibt eine Anmerkungen dazu.

NCLC decken vielfältige Anwendungsfälle ab, von persönlichen Verbesserung bis zu Kooperation mit anderer Unternehmenssoftware.

NCLC ist immer besser als Workaround- Lösungen via Excel oder ähnlichem.

EB02\_Rosengrün, Robin

Einfache Prozesse können vereinfacht werden.

Ein weiterer Vorteil entsteht durch eine Kooperation von IT-Entwickler und Citizen Developer. In diesem Fall würde der Citizen Developer die Idee haben und eine erste einfach Version der Applikation mit dem entsprechenden User Interface erstellen und der IT-Entwickler würde die Businesslogiken und Vernetzung mit den Datenquellen erstellen. Weitere Unterstützung würde dann dadurch entstehen, dass der IT-Entwickler den Citizen Developer beim Application Lifecyclemanagement unterstützt.

Ein weiterer Vorteil ist, dass Citizen Developer wesentlich schneller

Applikationen entwickeln können, da ein langwieriger initialer Planungsprozess, der bei einer IT-Entwicklung entstehen würde, entfällt. Ein Citizen Developer sieht ein Fachproblem und kann direkt versuchen, dafür eine Lösung zu entwickeln. Dadurch ist er wesentlich schneller und flexibler. Durch die Umgehung der IT-Abteilung als Bottleneck erschließt sich die Möglichkeit, dass eine wesentlich höhere Zahl von Anwendungen tatsächlich entwickelt wird und auch umgesetzt wird. Die verschiedenen problematischen Prozesse im Unternehmen können so behoben werden, durch die Fachabteilungen selber und das in einer wesentlich kürzeren Zeit.

EB03\_Fabian Gackstatter

Als ein wichtiger Vorteil wird empfunden, dass im Zuge der Beauftragung einer klassischen IT-Lösung viele Anforderungen aufgeschrieben werden, diese aber meist nie so umgesetzt werden, wie es gewollt wurde. Dieses Kommunikationsproblem würde sich lösen lassen, wenn Entwickler und Anforderer ein und dieselbe Person sind. Ein weiter positiver Aspekt, der sich ergibt, ist die persönliche Verbesserung der Produktivität der MA, in dem diese eigene Prozesse automatisieren können und dadurch schneller und effektiver ihre Aufgaben erledigen können. Ein positiver Beitrag zu Entwicklungsgeschwindigkeit ist es, dass die Plattformen im Kern sehr einfach gestaltet sind, so dass Nicht-Entwickler diese auch nutzen können. Das wirkt sich natürlich auch positiv auf die IT-Entwicklungsgeschwindigkeit aus, da sie einerseits die kleinen Prozesse, die der Citizen Developer realisiert, nicht mehr in ihrem Backlog haben und sich dadurch auf die andere Entwicklungen fokussieren können, welche nicht durch den Citizen Developer realisiert werden können.

EB04\_Göttl, Simone

Einer der wichtigen Vorteile ist, dass fachliches Know-How ohne



Reibungsverluste in die IT-Lösung umgesetzt werden kann. Dieses Problem entsteht durch die verschiedenen Ansprechpartner bei den klassischen IT-Lösungen, was im NCLC-Bereich entfällt.

Im Bereich der Hilfsprozesse kann NCLC dabei verschiedene Prozesse zumindest teilweise automatisieren und dadurch die persönliche Produktivität der MA erhöhen.

Das Citizen Development ermöglicht in diesem Zusammenhang das Hinzufügen von Wissen der Fachabteilung direkt in die Lösung, welche entwickelt wird.

EB05\_Rohrbeck, Christian

Ein wichtiger Vorteil von NCLC ist, dass der Aspekt der Übersetzung zwischen den Abteilungen und der IT-Abteilung entfällt und dadurch enorm viel Zeit gespart werden kann. Der Entwickler sieht direkt, was er benötigt und weiß, da er oftmals auch der Anforderer ist bzw. aus der Abteilung des Anforderers kommt, was er tun muss und versteht die Anforderungen genau.

Dadurch lässt sich eine Erhöhung der Produktivität erreichen, wenn der eigene Prozess beschleunigt wird. Die Lösung ist im Weiteren auch wesentlich effektiver, weil die Anforderungen komplett umgesetzt werden, ohne dabei Kommunikationsverluste zu haben. Aufgrund des agilen Charakters von NCLC-Lösungen ist es möglich, für jeden individuell seine eigenen Bedarfe zu decken und dadurch bestmögliche persönliche Lösungen zu entwickeln.

EB06\_Hesse, Moritz

Der Experte ist der Meinung, dass durch NCLC es zu schnelleren Entwicklungsergebnissen kommt, da die Entwicklung von einem selbst übernommen werden kann und kein umständlicher Weg über die IT-Abteilungen mehr genommen werden muss. Anforderungen müssen nicht mehr übersetzt und es müssen keine freien Kapazitäten der IT-Abteilung mehr berücksichtigt werden. Damit korreliert der positive Effekt,

dass der Aufgabenbacklog der IT-Abteilungen kleiner wird, da viele Aufgaben dort gar nicht mehr landen und die Fachabteilung ihre eigenen Lösungen entwickeln kann. Dies führt zu einer teilweisen Auflösung des Bottlenecks IT-Abteilung.

Für Unternehmen ergibt sich dadurch auch die positive Situation, dass eine schnelle Anpassung von Prozessen und Anwendungen an die geänderten Gegebenheiten am Markt oder dem Unternehmensumfeld möglich ist. Dies kann zu dem positiven Effekt führen, dass Bottlenecks innerhalb des Unternehmens aufgelöst werden, welche aufgrund der fehlenden Umsetzung einer Anwendung entstanden sind und durch die Automatisierung von einem Prozess zum Beispiel verschwunden sind.

EB07\_Janke, Tino

Der Experte vertritt die Meinung, dass durch eine im Verhältnis zur normalen IT-Entwicklung kostengünstige Plattformgebühr es möglich ist, Anwendungen zu entwickeln, die aus Sicht der Fachabteilung wichtig sind. Dabei schwingt der positive Aspekt mit, dass es zu keinen Reibungsverlusten mehr kommt in der Kommunikation zwischen Fachabteilung und IT-Abteilung.

Es ist auch möglich, die NCLC-Programme mit anderen im Unternehmen verwendeten Programmen zusammen zu verwenden und zu verknüpfen.

Durch diese Kombination es möglich, teilweise schneller und standardisierter dies durchzuführen, da die Plattformen hier Rahmenbedingungen automatisch vorgeben. Es wird der Vergleich zu Excel gezogen, wo die meisten Menschen innerhalb der Rahmenbedingungen von Exceltabellen arbeiten und die wenigsten die letzten 10% gehen, mit VBA beginnen Makros in der Tiefe zu programmieren.

Ein Vorteil, der oft nicht gesehen wird, ist das Fachabteilungsentwickler aufgrund ihres fehlenden IT-Wissens unter anderem auf unterorthodoxe Lösungsvorschläge kommen, weil die

Fachanwender mit einem anderem Blickwinkel auf das Problem schauen. Durch NCLC können IT-Workflows abgebildet werden auf Basis der Anforderungen des Fachbereichs. Es ist auch möglich, große komplexe Anwendungen, die normalerweise als komplexe Lösung durch die IT realisiert werden würden, wesentlich effektiver und effizienter durch NCLC zu realisieren. Das kann von einer Buchhaltungssoftware bis hin zu einem Endkundenprodukt reichen.

EB08\_Fey, Peter

1 - Vorbereitung Unternehmen\1.6 - Vorteile NCLC Plattformen\1.6.1 - Lösung Kommunikationsproblem

Die Vorteile sehe ich natürlich insbesondere darin, dass ich kein Kommunikationsproblem mit irgendwelchen Entwicklern habe, denen ich erst mal deutlich machen muss, was ich eigentlich will. Ich kann sozusagen selber meinen eigenen Prozess, so wie er mir vorschwebt, ohne Code-Kenntnisse mit den relativ einfachen Werkzeugen als logischen Prozess besser umsetzen. Also, das ist ein großer Vorteil.

EB01\_Eckhard Hauenherm: 13 - 13 (0)

Also vielleicht die, die Geschwindigkeit. Ich zeige immer mal so ein Bild bei uns, wie ein klassisches IT-Projekt aufgesetzt wird, hat man einen sehr hohen Initialaufwand, ohne dass eine Zeile Code umgesetzt ist. Also klassischerweise haben wir da einen Fachbereich mit der Anforderung, dann haben wir einen IT-Projektleiter, der das irgendwie noch mal in IT-Anforderungen übersetzt und haben dann Entwickler, intern oder extern, die dann das umsetzen. Und bis diese Struktur aufgebaut ist, die Entwicklungen gefunden sind, die Anforderungen eingegossen sind, dauert es, hat man erst einen gewissen Initialaufwand, egal jetzt, ob das Projekt agil stattfindet oder klassisch mit einem langen Lasten/ Pflichtenheft. Irgendwie man hat einen langen Initialaufwand und bei Low Code kann ich einfach direkt anfangen, meine Ideen in eine Anwendung zu übersetzen und hab sehr schnell eine gemeinsame Sicht darauf. Wir haben uns kurz vor der Aufnahme darüber unterhalten, dass jemand

eine falsche Sicht auf dem Thema hatte, das finde ich bei der Softwareentwicklung auch immer ein ganz großes Thema. Dass der Fachbereich eine Vorstellung davon hat, was er will. Aber eine Sprachblockade quasi zum IT-Bereich besteht und dann in den Köpfen verschiedene Vorstellungen davon entstehen, was man eigentlich braucht und das fliegt dann erst auf, oder merkt man dann erst, wenn es dann den ersten Prototyp gibt. Und ja, dann hat man halt bei der Low Code-Entwicklung sehr schnell einen ersten Prototyp, den man dann immer weiter verfeinert und hat sehr schnell ein gemeinsames Bild davon, was eigentlich entwickelt wird und kann dann halt an vielen Stellen noch mal anpassen, was eigentlich der Bedarf ist.  
EB02\_Rosengrün, Robin: 13 - 13 (0)

Das heißt, dafür eignet sich es definitiv und ein Vorteil ist natürlich dort dann auch, also jeder, der schon mal ein IT-Projekt gemacht hat, ein ganz klassisches, man schreibt ganz viele Anforderungen, gibt die an die IT und bekommt aber nicht das zurück, was man sich eigentlich gewünscht hat. Wenn ich aber jetzt ein Fachbereich oder selber die Person bin, die die Anforderung definiert und es umsetze, dann geht es auch schneller und es ist am Ende auch besser, weil es zu meinen Anforderungen passt.  
EB04\_Göttl, Simone: 15 - 15 (0)

ich denke mal, der absolute Vorteil, der Hauptvorteil, ist einfach, dass hier das fachliche Know-how ohne Reibungsverluste in eine IT-Lösung umgesetzt werden kann. Das muss man sagen, ist in unserem Berufsfeld das Hauptproblem, dass wir verschiedene Ansprechpartner haben, die so unterschiedliche Sprachen sprechen, dass da enorm hohe Kommunikationsreibungsverluste da sind, das entfällt. Also wir haben, derjenige der den Prozess verstanden hat, ist auch in der Lage, das abzubilden in ein System.

EB05\_Rohrbeck, Christian: 15 - 15 (0)

Ich selber bin Wirtschaftsinformatiker von der Historie und ich weiß, das ist quasi so ein wesentliches Charakteristikum von Wirtschaftsinformatikern, dass sie gerne als Dolmetscher zwischen Fachabteilung, also IT fremden Abteilungen und der IT wiederum fungieren, weil sie in der Lage sind, einerseits betriebliche-fachliche Prozesse zu verstehen, andererseits aber auch IT-Zusammenhänge, technische Zusammenhänge und insofern sind Wirtschaftsinformatiker immer gut geeignet, um ebendiese fachlichen Vorgänge technisch zu beschreiben. Und das ist aber immer ein schwieriger Prozess und diesen Vorgang kann man sich dann ja sparen, weil der Fachmitarbeiter muss dem IT'ler nicht erklären, was seine Anforderungen sind, sondern er geht einfach auf die Lösung zu und die Low Code/ No Code-Lösung zu und sieht, was er benötigt und richtet sich das halt alles selber ein und das ist ein absoluter Vorteil, dadurch entsteht unglaublich hohe Geschwindigkeit. Und daraus natürlich weitergeleitete Effekte, weitergeleitete betriebliche Effekte.

EB06\_Hesse, Moritz: 17 - 17 (0)

Diese Themen selber begleiten zu können, also sie würden dann in der Low Code/ No Code-Welt keine Anforderungen mehr an ihre IT-Abteilung oder an externe IT-Berater stellen, sondern würden sich anhand ihrer Anforderungen selber mit der Entwicklung beschäftigen, was zu schnelleren Entwicklungsergebnissen führt, weil die Anforderungen eben nicht erst umgesetzt und übersetzt werden müssen und dann auf freie Kapazitäten warten, sondern man als Fachabteilung im Prinzip seine Anforderungen selber steuern kann, direkt selber in die Umsetzung gehen kann. Zum anderen würde es auch die IT-Abteilung natürlich entlasten, wenn Anforderungen bei denen nicht mehr aufschlagen, weil sich die Fachabteilung selber damit beschäftigen kann.

EB07\_Janke, Tino: 13 - 13 (0)

Ja also die Low Code/ No Code-Plattformen, die habe ich ja eben schon

beschrieben. Im Prinzip kann man also hingehen und gegen den Einwurf kleiner Münzen bei den herstellenden Unternehmen, die natürlich programmieren müssen, diese Plattformen sich aneignen und die, ich sage mal, die Gebrauchsbücher, Handbücher lesen und dann dort die großen Applikationen schreiben, die das Unternehmen tatsächlich aus Sicht der Fachabteilung braucht. Es ist keine Abstimmung notwendig zwischen der IT-Abteilung und den Fachabteilungen. Es ist, der Informationsverlust kann geschmälert werden dadurch, bevor ich dann so eine Applikation in die Tat umsetze.

EB08\_Fey, Peter: 11 - 11 (0)

1 - Vorbereitung Unternehmen\1.6 - Vorteile NCLC Plattformen\1.6.2 - Prozessverbesserung

Realistische Erwartungen aus meiner Sicht, was man heutzutage sehr gut schafft, ist einfache Prozesse zu automatisieren. So klassische Geschäftsprozesse, wie Rechnungseingang und solche Dinge, Prozesse automatisieren

EB01\_Eckhard Hauenherm: 19 - 19 (0)

Also ich sehe die realistische Erwartung in der Umsetzung interner Prozesse, interner kleiner Applikationen, die ich selber nutze, um Daten zu pflegen, um Prozesse zu automatisieren.

EB01\_Eckhard Hauenherm: 19 - 19 (0)

Große Geschichte, die ich darüber hinaus auch noch sehe, ist Kommunikation zwischen 2 oder mehreren Abteilungen. Also es gibt ja ganz klassische Prozesse, die von Abteilungen A nach B nach C und dann manchmal nach D manchmal nach E laufen oder so was, die ganz klassisch über E-Mails abgebildet werden oder, was ich noch schlimmer finde, Word Dokumente, die ausgefüllt werden müssen oder so was und dann per E-Mail weitergeschickt werden. Also, das ist auch noch ein großer, großer Einsatzbereich.

EB02\_Rosengrün, Robin: 17 - 17 (0)

Also wenn es jetzt wirklich nur darum geht, ich habe hier einen Papierprozess. Ich habe hier ein Formular, was jeden Tag ausgefüllt werden muss. Da sind fünf Informationen drauf und wenn ich

die ausgefüllt habe, scanne ich das Ding ein und schicke es an eine E-Mail-Adresse.

EB03\_Fabian Gackstatter: 15 - 15 (0)

Wenn es aber darum geht, sich einerseits Helferlein im Alltag zu bauen, vielleicht auch, um Sachen zu automatisieren, die man tagtäglich macht oder um fachbereichsspezifische Prozesse geht, dann ist das, glaube ich schon etwas, wo die Endentwickler auch Vorteile erhalten können.

EB04\_Göttl, Simone: 15 - 15 (0)

Ja also, realistisch ist auf jeden Fall erstmal, sich vom Scope her, dessen was diese Anwendung kann, ein bisschen, ich sage mal, eine einfache Erwartungshaltung zu haben. Ich glaube nicht, dass solche Plattformen geeignet sind, hochkomplexe Prozesse abzubilden, sondern eher für einfache Hilfs-, na ja, für Hilfsaufgaben ist jetzt wieder untertrieben, für einfachere Aufgaben gedacht ist und das halt, tja, was sie realistisch erfüllen können. Sie können realistisch helfen, so eine Art Halbautomatisierung durchzuführen. Dass man im Prinzip einen Assistenten sich baut, aber trotzdem die ausführende Gewalt, sage ich mal, beim Mitarbeiter verbleibt.

EB05\_Rohrbeck, Christian: 21 - 21 (0)

Nun ja, daraus lässt sich eine Erhöhung, viel höhere Produktivität zum Beispiel, ableiten. Also wenn ich jetzt sage, eine Applikation ist in der Lage, meinen Prozess zu beschleunigen. Ja, dann habe ich ja die höhere Effektivität durch die Lösung, wenn ich jetzt aber sage, die Lösung braucht einen unglaublich hohen Aufwand, um hergestellt zu werden, dann ist quasi mein Produktivitätseffekt, den ich daraus gewinne, mit einem unglaublichen hohen komplexen Aufwand verbunden, den ich für die Lösung quasi bereitstellen muss. Indem ich aber einen sehr geringen Aufwand nur habe, um die Lösung mir zu entwickeln, habe ich diesen Produktivitätseffekt einfach viel schneller erzeugt, das heißt, die

Effektivität kann ich in beiden Fällen gewährleisten, die Effizienz aber, um zu dieser Lösung zu kommen, ist bei Low Code / No Code viel, viel höher.

EB06\_Hesse, Moritz: 19 - 19 (0)

Ich würde mal grundsätzlich behaupten, mit No Code/ Low Code-Lösung kann man mit Sicherheit heutzutage schon einen enorm großen Anteil von betrieblichen Prozessen abbilden. Ich würde aber behaupten, wenngleich ich das als eine unbestätigte oder unüberprüfte Hypothese kennzeichnen wollte, dass No Code und Low Code-Lösungen durchaus dort besser geeignet sind, einsetzbar sind und zu einer erfolgreichen Einführung führen, wo es nicht um hochoperative Prozesse geht. Also um die wesentlichen wertschöpfenden Prozesse, sondern eher um die Auxiliar-Prozesse, also bestimmte Hilfsprozesse, kleine Prozesse, die vielleicht auch nicht ständig und in dem standardisierten Wertschöpfungsmodell stattfinden, sondern die vielleicht eher mal eben schnell stattfinden müssen, also wo man mal schnell eine Datenerhebung benötigt. Wo man mal schnell irgendwie eine kleine Zwischenlösung benötigt, meine Erfahrung würde eher dagegensprechen, das wirklich für hochstandardisierte Industrieprozesse anzuwenden, da würde ich sagen, sind die Skaleneffekte für ein hochprofessionell entwickelten und auch überwachten Softwareartefakt, was man da einsetzt, durchaus geeigneter als so Low Code-Lösungen. Ich sehe den Vorteil eher bei diesen Hilfsprozessen, bei diesen Sekundärprozessen, bei diesen Auxiliar-Vorgängen.

EB06\_Hesse, Moritz: 27 - 27 (0)

Man muss sich ja vorstellen, bisher obliegt die Umsetzung aller digitalen Produkte, aller Prozesse, obliegt momentan der IT-Abteilung, wenn man sich da mal so umschaute, ist es eigentlich so, dass die IT-Abteilungen einfach ein riesiges Backlog mit sich her schleppen. Das heißt, große Anzahl an unbearbeiteten Aufgaben, an unbearbeiteten Anforderungen. Das ist so eine Art Flaschenhals



geworden, weil es auch einfach an Fachkräften mangelt. Genau. Durch den Einsatz von Low Code/ No Code möchte man erreichen, dass man diesen Flaschenhals so ein bisschen auflöst, indem man Kollegen, die eigentlich nicht programmieren können, in die Lage versetzt, ihre, ihren Digitalisierungsbedarf selber vornehmen zu können.

EB07\_Janke, Tino: 13 - 13 (0)

Auf der anderen Seite habe ich natürlich schnellere Entwicklungszeiten für meine Anwendungen und Prozesse, damit kann ich mich schneller auf Marktsituationen einstellen. Das heißt, ich habe nicht die Situation, wie es beispielsweise bei Corona war. Das ist jetzt ein Beispiel, ich erhalte jetzt die gesetzliche Auflage, dass ich den Impfstatus meiner Mitarbeiter kontrollieren muss, beziehungsweise die Testergebnisse meiner Mitarbeiter nachhalten muss. Ich würde das dann ohne Low Code/ No Code als Anforderungen an die IT geben. Ich bau dafür eine Anwendung. Die IT ist aber ausgelastet und das Ganze liegt erst mal, bis man in die Umsetzung starten kann. Mit Low Code/ No Code war es eben so, dass man aus ganz vielen Organisationen gehört hat, dass diese gesetzlichen Anforderungen, die ja neu waren, überraschend waren, nicht vorhersehbar waren, dass die innerhalb von Tagen bis wirklich wenigen Wochen so umgesetzt werden konnten, dass man eben auch ein digitales Produkt dafür hat, wo man diese Anforderungen abbilden kann und das ist natürlich ein Thema, das gäbe es mit einem traditionellen entwickeln nicht und ich kann natürlich auch, das vielleicht auch, als dritten Punkt, so die Flaschenhälse auflösen, die ich in meiner Organisation habe, wenn es darum geht, neue Produkte zu entwickeln, durch das Verteilen von Aufgaben auf mehrere Schultern.

EB07\_Janke, Tino: 17 - 17 (0)

Schnittstellen zu anderen Programmen, die werden im Nachhinein im Unternehmen wirksam, im Sinne der Finanzbuchhaltung. Die sind dann halt auch möglich.

Das heißt, im Prinzip kann ich große, komplexe Anwendungen, die im Moment auch schon im IT-Bereich programmiert worden sind, die kann ich in diesen No Code/ Low Code-Bereichen abbilden und zum Teil, das muss man ganz deutlich sagen, nicht nur schneller, sondern auch wegen der Standardisierung, also ich habe zwar die Freiheiten, ganz viel zu machen, aber ich habe nur ganz wenige, wenige ist vielleicht übertrieben gesagt, aber nur einige Möglichkeiten das zu realisieren, indem ich mich ja auf die Programmierung der No Code/ Low Code-Plattform verlassen muss und nur diese mir bereitgestellten Funktionen nutzen kann.

EB08\_Fey, Peter: 15 - 15 (0)

wenn ich ein Fachbereich-Mensch bin und schaue immer nur aus diesen fachlichen Aspekten auf die IT, dann stoße ich natürlich an Grenzen mit dem Verständnis für die IT. Und komm dann vielleicht mit Lösungsvorschlägen, die auf den ersten Augenblick, aus IT-Sicht, deppert sich anfühlen.

EB08\_Fey, Peter: 19 - 19 (0)

1 - Vorbereitung Unternehmen\1.6 - Vorteile NCLC Plattformen\1.6.3 - Verbesserung der Arbeitsabläufe MA

Und decken weitreichend ab, einfach zwischen der, also auch für die persönliche Produktivität, aber ich setze das irgendwie immer an zwischen der persönlichen Excel-Datei und dem SAP ERP System.

EB02\_Rosengrün, Robin: 17 - 17 (0)

Also es gibt ja nicht nur den Citizen Developer am einen Ende und den Pro Developer am anderen. Ich habe ja auch den Gesamtbereich in der Mitte. Den sogenannten Hybrid Approach oder Fusion Approach, wo ich tatsächlich sagen kann, hier arbeiten jetzt Citizen Developer und Pro Developer zusammen. Das heißt, der Citizen Developer hat vielleicht eine gute Idee für eine Applikation. Er hat auch schon mal das Design erstellt. Er hat mit dem No Code/ Low Code-Tool das UI erstellt, tut sich jetzt aber schwer, die Businesslogik zu implementieren, für die dann No Code/ Low Code irgendwann an ihre Grenzen stößt. Oder eben

auch, wo dann Datentöpfe angezapft werden sollen, die dem klassischen Citizen Developer nicht zur Verfügung stehen und da kommt dann das Zusammenspiel mit dem Pro Entwickler dazu, der da tatsächlich unterstützt, diese Businesslogik zu bauen, der den Citizen Developer auch unterstützt bei einem Application Lifecycle Management. Das heißt, die Applikation, die der Citizen Developer auf seinem eigenen Environment entwickelt hat, die dann transportiert wird auf ein Testsystem, die dann transportiert wird auf ein Produktivsystem, zum Beispiel. Und das sind, zum Beispiel, zehn Szenarien, die ich mit der Power Platform komplett abdecken kann. Von dem Citizen Developer, der sich mit Canvas Apps auf SharePoint Basis, mit denen fängt er mal an, bis zum Pro Developer, der den kompletten Stack von Dynamics Power Platform, Azure zur Verfügung hat und alles, was dazwischen passiert.

EB03\_Fabian Gackstatter: 15 - 15 (0)

Wenn es aber darum geht, sich einerseits Helferlein im Alltag zu bauen, vielleicht auch, um Sachen zu automatisieren, die man tagtäglich macht oder um fachbereichsspezifische Prozesse geht, dann ist das, glaube ich schon etwas, wo die Endentwickler auch Vorteile erhalten können.

EB04\_Göttl, Simone: 15 - 15 (0)

Und meine Vorstellung von dem Citizen Development ist diejenige, dass man einem Kollegen, der in einer Fachabteilung ist und an einem speziellen Problem arbeitet, eine solche Software an die Hand gibt und dass dieser sich, ohne jetzt Fachkenntnisse von Programmierung zu haben, in der Lage sieht, einfache Algorithmen aufzusetzen, die dann Funktionen ausführen innerhalb dieses speziellen Geschäftszweiges, in dem die Kollegen da dann unterwegs sind.

EB05\_Rohrbeck, Christian: 13 - 13 (0)

Nun ja, ich komme selber aus der Softwareentwicklung, also bin selber auch langjährig in der Softwareentwicklung

tätig gewesen und auch zu Zeiten, wo es noch nicht agile Software-Entwicklungsverfahren gab. Das heißt, ich kenne noch die sehr klassischen Verfahren, wie zum Beispiel das Wasserfallmodell. Und derartige Modelle sind ja geprägt, das zeigen die Vorteile von so agilen Methoden auf. Dass sie sehr zäh sind, sehr langwierig sind. Und insofern sind Vorteile von solchen Low Code-Lösungen oder auch von solchen Initiativen, wie Citizen Development, dass man die Notwendigkeit, einfach von langfristigen IT-Projekten einerseits, aber auch andererseits überhaupt die erforderliche Abstimmung mit der IT, dass die außen vor bleibt und dass letztendlich jeder seinen eigenen unmittelbaren Bedarf mit einer No Code/ Low Code-Lösung decken kann.

EB06\_Hesse, Moritz: 17 - 17 (0)

Diese Themen selber begleiten zu können, also sie würden dann in der Low Code/ No Code-Welt keine Anforderungen mehr an ihre IT-Abteilung oder an externe IT-Berater stellen, sondern würden sich anhand ihrer Anforderungen selber mit der Entwicklung beschäftigen, was zu schnelleren Entwicklungsergebnissen führt, weil die Anforderungen eben nicht erst umgesetzt und übersetzt werden müssen und dann auf freie Kapazitäten warten, sondern man als Fachabteilung im Prinzip seine Anforderungen selber steuern kann, direkt selber in die Umsetzung gehen kann. Zum anderen würde es auch die IT-Abteilung natürlich entlasten, wenn Anforderungen bei denen nicht mehr aufschlagen, weil sich die Fachabteilung selber damit beschäftigen kann.

EB07\_Janke, Tino: 13 - 13 (0)

Ja also die Low Code/ No Code-Plattformen, die habe ich ja eben schon beschrieben. Im Prinzip kann man also hingehen und gegen den Einwurf kleiner Münzen bei den herstellenden Unternehmen, die natürlich programmieren müssen, diese Plattformen sich aneignen und die, ich sage mal, die Gebrauchsbücher, Handbücher lesen und dann dort die großen Applikationen schreiben, die

das Unternehmen tatsächlich aus Sicht der Fachabteilung braucht. Es ist keine Abstimmung notwendig zwischen der IT-Abteilung und den Fachabteilungen. Es ist, der Informationsverlust kann geschmälert werden dadurch, bevor ich dann so eine Applikation in die Tat umsetze.

EB08\_Fey, Peter: 11 - 11 (0)

Also die realistischen Erwartungen, die habe ich ja eben ganz kurz schon mal aufgezeichnet, dazu fällt auf jeden Fall die Abbildung eines Workflows im IT-Bereich auf der Basis und auf der Handlungsmöglichkeit eines Fachbereichs. Also die Mitarbeiter, die Mitarbeiterinnen eines Fachbereichs, können sich hinsetzen, können darüber nachdenken, was wollen wir eigentlich für ein Programm haben? Was soll das leisten? Wie wollen wir uns mit anderen vernetzen? Was wollen wir dann mit diesen Informationen machen? Welche Handlungen sollen ausgelöst werden? Das ist aktuell möglich, und zwar von dem Beginn der Produktion, über die Auslieferung, über den Kundenkontakt, bis hin zum Einzug der Zahlungen durch die Geldinstitute.

EB08\_Fey, Peter: 15 - 15 (0)

Die Fachabteilung findet dieses oder jenes nur nicht so ganz gelungen und sie hätte gerne folgendes. Oder folgender Fehler, das kann durchaus auch sein, ist in diesem IT-Programm aufgetreten, das sollte korrigiert werden. Das ging ja jetzt bisher in der Regel, wenn man es in den alten Strukturen sagt. Es ging über die entsprechenden Hefte, die man angelegt. Also man hat gesagt: "Das und das hätte ich jetzt gerne in der Umsetzung." Dann wurde ein Pflichtenheft daraus abgebildet und das Pflichtenheft wurde dann umgesetzt und das Pflichtenheft, das haben wir dann häufig bei der T-Systems Fachkonzept genannt und diese Umsetzung der Kundenwünsche in das Fachkonzept, oder in das Pflichtenheft, die haben häufig zu Informationsdefiziten geführt. Und da genau setzt jetzt No Code/ Low Code ein. Diese

Informationsdefizite sollen jetzt vermieden werden. Derjenige, der an der Tastatur sitzt und dort aus einer großen Vielzahl von Angeboten, Bibliotheken und Funktionen grafisch dann sein Programm zusammenklicken kann. Wie gesagt, das ist keine Sache, die man innerhalb von 15 Minuten lernt, sondern da braucht man schon einige Wochen für, bis man das gut kann und in der entsprechenden Qualität auch machen kann. Und mit Aspekten des abstrakten Niveaus. Derjenige, der ist ja aus dem Fachbereich, der versteht eigentlich sehr genau, was da getan werden muss. Und in diesem Zusammenhang ist der Informationsverlust vom Fachbereich bis zu demjenigen, der es umsetzt, eigentlich möglichst kurz.

EB08\_Fey, Peter: 37 - 37 (0)

1 - Vorbereitung Unternehmen\1.6 - Vorteile NCLC Plattformen\1.6.4 - Steigerung der Qualität von Lösungen

Aber jetzt gerade, weil dein Thema ja auch das Projektmanagement ist, da Sachen, die jetzt bei uns nicht im ERP System abgebildet werden, wäre ich immer sofort dabei, mach das bitte, bitte, bevor dort anfängt, es über Excel-Dateien etc. zu machen, nutzt da ein Low Code-Tool, eine Low Code-Plattform dafür. Man hat immer ein besseres Ergebnis, als wenn man das über Kalkulationstabellen macht oder so was.

EB02\_Rosengrün, Robin: 17 - 17 (0)

Also es gibt ja nicht nur den Citizen Developer am einen Ende und den Pro Developer am anderen. Ich habe ja auch den Gesamtbereich in der Mitte. Den sogenannten Hybrid Approach oder Fusion Approach, wo ich tatsächlich sagen kann, hier arbeiten jetzt Citizen Developer und Pro Developer zusammen. Das heißt, der Citizen Developer hat vielleicht eine gute Idee für eine Applikation. Er hat auch schon mal das Design erstellt. Er hat mit dem No Code/ Low Code-Tool das UI erstellt, tut sich jetzt aber schwer, die Businesslogik zu implementieren, für die dann No Code/ Low Code irgendwann an ihre Grenzen stößt. Oder eben auch, wo dann Datentöpfe angezapft werden sollen, die dem klassischen Citizen Developer nicht zur Verfügung stehen und da kommt dann das

Zusammenspiel mit dem Pro Entwickler dazu, der da tatsächlich unterstützt, diese Businesslogik zu bauen, der den Citizen Developer auch unterstützt bei einem Applikation Lifecycle Management. Das heißt, die Applikation, die der Citizen Developer auf seinem eigenen Environment entwickelt hat, die dann transportiert wird auf ein Testsystem, die dann transportiert wird auf ein Produktivsystem, zum Beispiel. Und das sind, zum Beispiel, zehn Szenarien, die ich mit der Power Platform komplett abdecken kann. Von dem Citizen Developer, der sich mit Canvas Apps auf SharePoint Basis, mit denen fängt er mal an, bis zum Pro Developer, der den kompletten Stack von Dynamics Power Platform, Azure zur Verfügung hat und alles, was dazwischen passiert.

EB03\_Fabian Gackstatter: 15 - 15 (0)

Die sind dann halt auch möglich. Das heißt, im Prinzip kann ich große, komplexe Anwendungen, die im Moment auch schon im IT-Bereich programmiert worden sind, die kann ich in diesen No Code/ Low Code-Bereichen abbilden und zum Teil, das muss man ganz deutlich sagen, nicht nur schneller, sondern auch wegen der Standardisierung, also ich habe zwar die Freiheiten, ganz viel zu machen, aber ich habe nur ganz wenige, wenige ist vielleicht übertrieben gesagt, aber nur einige Möglichkeiten das zu realisieren, indem ich mich ja auf die Programmierung der No Code/ Low Code-Plattform verlassen muss und nur diese mir bereitgestellten Funktionen nutzen kann. Ähnlich kann man das ja bei Excel auch beobachten. Also ich kann nicht beliebig viele Funktionen haben, sondern die Excel-Programmierer bieten mir Funktionen an und die kann ich nutzen. Das ist reichlich viel. Die meisten Menschen werden von diesen Funktionsmöglichkeiten höchsten 10 bis 25 % nutzen und so ist es wahrscheinlich auch bei den No Code/ Low Code- Plattformen, dass viel mehr angeboten wird, als tatsächlich nachher genutzt wird. Aber deshalb können diese Funktionen, die genutzt werden, noch besser sein, weil

eben keine individuellen Programmierfehler mehr auftreten, sondern alles das, was da erarbeitet worden ist, zum Beispiel die Anbindung an Geld- oder Kreditinstitute, die sind dann geprüft und nur dann, wenn die Qualität tatsächlich sich eingestellt hat, nach vielen, vielen Prüfungen in dieser Plattform, werden sie dann freigegeben. Und wenn Fehler auftreten innerhalb der Plattform, werden die dann auch schnell durch die Kunden gemeldet und können dann bei den Plattformherstellern korrigiert werden. Also es kann zum Beispiel sein, gerade in der Anbindung an Kreditinstitute, dass da die No Code/ Low Code-Funktionen sicherer, besser und praktikabler sind als eine Einzelprogrammierung und neben der Zahl der Spalten sollen sie eben auch eine höhere Datensicherheit bieten.

EB08\_Fey, Peter: 15 - 15 (0)

1 - Vorbereitung Unternehmen\1.6 - Vorteile NCLC Plattformen\1.6.5 - Höhere Entwicklungsgeschwindigkeit

Das hat auch was mit Geschwindigkeit zu tun. Ich bin in den meisten Fällen dann schon relativ schnell fertig. Die Plattformen sind heute wirklich so mächtig und so gut, dass man wirklich in ein paar Stunden schon sehr gute Dinge zustande bringt. Ja, das sind die Vorteile. Sie haben ja nach den Vorteilen gefragt, die Nachteile kommen wahrscheinlich später.

EB01\_Eckhard Hauenherm: 13 - 13 (0)

Also vielleicht die, die Geschwindigkeit. Ich zeige immer mal so ein Bild bei uns, wie ein klassisches IT-Projekt aufgesetzt wird, hat man einen sehr hohen Initialaufwand, ohne dass eine Zeile Code umgesetzt ist. Also klassischerweise haben wir da einen Fachbereich mit der Anforderung, dann haben wir einen IT-Projektleiter, der das irgendwie noch mal in IT-Anforderungen übersetzt und haben dann Entwickler, intern oder extern, die dann das umsetzen. Und bis diese Struktur aufgebaut ist, die Entwicklungen gefunden sind, die Anforderungen eingegossen sind, dauert es, hat man erst einen gewissen Initialaufwand, egal jetzt, ob das Projekt agil stattfindet oder klassisch mit einem langen Lasten/ Pflichtenheft. Irgendwie man hat einen langen



Initialaufwand und bei Low Code kann ich einfach direkt anfangen, meine Ideen in eine Anwendung zu übersetzen und hab sehr schnell eine gemeinsame Sicht darauf.

EB02\_Rosengrün, Robin: 13 - 13 (0)

Der Vorteil des Citizen Developers generell ist einfach das Potenzial, das ich innerhalb eines Unternehmens habe, weil wie gesagt, die IT ist in der Regel ein eher kleiner Bestandteil eines jeden Unternehmens und kann dadurch auch sehr schnell zum Bottleneck werden, wenn es darum geht, neue Applikationen einzuführen, Regress aus dem Business zu bedienen, weil der klassische Entwicklungsprozess eben etwas komplexer gestrickt ist, mit den verschiedenen Qualitätsstufen, die durchlaufen werden müssen. Mit den verschiedenen Anforderungen. Ich muss ein Design erstellen. Ich muss einen technischen Blueprint erstellen, das muss genehmigt werden. Der Citizen Developer muss sich um solche Sachen in der Regel nicht kümmern. Der Citizen Developer kommt. Er hat eine Idee, dass er sagt: "Ich habe hier einen papierbasierten Prozess, den würde ich gerne digitalisieren." Oder: "Wir haben hier eine uralte Applikation, die eigentlich schon nicht mehr funktioniert. Die macht nicht viel, das könnte ich mit einem modernen Tool eigentlich selber ersetzen, diese Geschichte" und mit den entsprechenden Tools kann ein Citizen Developer dann innerhalb von ein paar Stunden sich seine eigenen Applikationen zusammenbauen, ohne den kompletten Entwicklungszyklus durchlaufen zu müssen, den klassische Enterprise-Applikationen einfach haben. Und dadurch, dass prinzipiell jeder zu einem Citizen Developer werden kann, habe ich entsprechend auf einmal viel mehr Potenzial, technische Lösungen zu implementieren, ohne meine IT als Bottleneck im Hintergrund zu haben.

EB03\_Fabian Gackstatter: 11 - 11 (0)

Also schon mal, was ich realistisch tatsächlich einmal erreichen kann ist, dass wesentlich mehr Ideen umgesetzt

werden, die aus sämtlichen Businessströmen kommen können. Die im normalen, im klassischen Unternehmen, sage ich mal, irgendwo in der IT im Zweifelsfall hängen bleiben und niemals das Tageslicht erblicken. Der Citizen Developer hat eben selber die Möglichkeit, gerade kleinere Sachen umzusetzen, mit der sich eine normale IT überhaupt nicht beschäftigen würde. Also wenn es jetzt wirklich nur darum geht, ich habe hier einen Papierprozess. Ich habe hier ein Formular, was jeden Tag ausgefüllt werden muss. Da sind fünf Informationen drauf und wenn ich die ausgefüllt habe, scanne ich das Ding ein und schicke es an eine E-Mail-Adresse. Da wird dir niemals eine interne IT eine Applikation dafür bauen, weil du dann immer so weit unten im Stapel des Notwendigen landen wirst, dass es wahrscheinlich niemals so weit kommt. Die interne IT kümmert sich um die SAPs dieser Welt, um die Secret Server dieser Welt, um die Dynamics Solutions dieser Welt, aber gerade die kleinen Prozesse, die bleiben relativ häufig liegen, wenn ich dann nicht einen dedizierten Entwickler habe, der sich eben um diese kleinen Themen kümmert. Der Citizen Developer, der kann tatsächlich selber diese Sachen bauen. Die können dann im Zweifelsfall von der IT noch einmal gereviewt werden, bevor sie dann zum produktiven Einsatz kommen, nehmen aber den Großteil der Arbeit der IT einfach ab.

EB03\_Fabian Gackstatter: 15 - 15 (0)

Also, wie gesagt die Einfachheit, also das es auch Nicht-Programmierer nutzen können und das ist natürlich dann ein großer Vorteil für eine IT, also gerade bei großen Konzernen oder bei Banken und Versicherungen. Dass IT-Entwicklungen häufig langwierig dauern, zu lang für einen Endnutzer, dass es auch sehr kostspielig ist, plus, das sehen wir ja auch alle, dass gerade Entwickler oder Cloud-Entwickler auch nicht immer verfügbar sind und der Markt ziemlich leergefegt ist von diesen Fachkräften. Das heißt, es besteht auch häufig einfach gar keine Möglichkeit mehr, so schnell auf

Anwendungen, Apps als IT reagieren zu können und deshalb hat sich dieser Markt so entwickelt. Das ist definitiv dort auch ein großer Vorteil für No Code/ Low Code, um das lösen zu können.

EB04\_Göttl, Simone: 13 - 13 (0)

1 - Vorbereitung  
Unternehmen\1.7 -  
Nachteile NCLC  
Plattformen

Als eine Problemstellung wird empfunden, dass die von Citizen Developern entwickelten Anwendungen qualitativ nicht mit den Anwendungen und klassisch entwickelten Anwendungen übereinstimmen, welche von professionellen Entwicklern programmiert werden. Dabei wird auch bezweifelt, dass zum jetzigen Zeitpunkt der Entwicklung solche Anwendungen geeignet sind, um außerhalb des Unternehmen in der Interaktion mit dem Kunden eingesetzt zu werden.

Ein weiterer Aspekt, der bei NCLC beachtet werden muss, ist die Situation der Entlastung des Fachkräftemangels. Einerseits gibt es ja nicht nur Fachkräftemangel in der IT-Abteilung, sondern auch in anderen Berufsfeldern, zum anderen ist es fraglich, dass die Fachabteilungen so viel mehr Kapazitäten haben als die IT-Abteilung.

EB01\_Eckhard Hauenherm

Es wird durch den Experten darauf hingewiesen, dass aufgrund von mangelnden IT-Wissen, die Applikationen eine schlechtere Qualität haben. Das kann sich negativ auf die Performance auswirken oder in nicht standardisierten Datenstrukturen. Die Auswirkungen machen sich bemerkbar, wenn die zugehörigen Projekte erweitert und die Applikationen verändert oder in ihrem Funktionsumfang erweitert werden sollen.

Als weiterer Aspekt, auf den Unternehmen achten sollten, ist, dass nicht die Erwartungshaltung entstehen darf, dass NCLC Standardanwendungen ersetzen kann. Im Falle, dass eine Verarbeitung mit einer großen Anzahl von komplexen Kundendatensätzen oder ähnlichem gearbeitet wird, können diese Anwendungen nicht an die Lösungen von großen Unternehmen

heranreichen, welche sich schon jahrelang mit diesem Thema beschäftigen.

EB02\_Rosengrün, Robin

Eine falsche Erwartungshaltung ist, dass durch NCLC die interne IT-Abteilung ersetzt werden könnte oder Standard-Software-Lösungen obsolet werden. Diese werden weiter benötigt.

EB03\_Fabian Gackstatter

Eine falsche Erwartungshaltung ist, dass Standard-Software-Lösungen komplett ersetzt werden können. Das spielt in den Aspekt Erwartungsmanagement ein. Viele Hersteller stellen ihre Lösung als sehr schnell und einfach anzuwenden dar. Es gibt zwar Use-Cases, die sich leicht und schnell umsetzen lassen, das betrifft aber nicht alle. Es muss klar sein das auch bei der Nutzung von NCLC-Plattformen es einen Anwendungsentwicklungsprozess gibt und dieser eben nicht immer schnell umgesetzt werden kann, sondern dass auch dort ein tiefgreifendes Verständnis der Plattform erarbeitet werden muss. Damit verbunden ist der Fakt, dass es nötig wird, auch dort Proof of Concept-Phasen durchzuführen, bei denen Prototypen erstellt werden, auch mehrere, bevor eine finale Version released werden kann.

EB04\_Görtl, Simone

Es wird durch den Experten darauf hingewiesen, dass in vielen Non-IT-Abteilungen grundlegende IT-Kenntnisse fehlen. Das führt dazu, dass eine falsche Erwartungshaltung an solche Plattformen gestellt wird oder bei einem negativen Ergebnis nicht die richtigen Schlussfolgerungen gezogen werden. Es wird nicht die eigentliche Umsetzung als Problem gesehen, sondern andere Aspekte, wie, dass die Plattform nicht passt. An dieser Stelle können zusätzliche Aufwände entstehen, die in der traditionellen IT-Entwicklung nicht entstanden wären. Das zeigt gut, dass eben nicht die IT-Abteilung und IT-Software-Lösungen komplett ersetzt werden können.

EB05\_Rohrbeck, Christian

1 - Vorbereitung Unternehmen\1.7 - Nachteile NCLC Plattformen\1.7.1 - schlechte Qualität der Anwendung

Also es gibt natürlich erst mal aus Programmsicht so ein bisschen, sagen wir mal, die Qualität dessen, was da rauskommt, hängt natürlich sehr stark davon ab, wie gut auch das technische und das logische Verständnis desjenigen ist, der programmiert. Es ist wirklich unterschiedlicher Qualität oder, sagen wir mal so, die Qualitätsschwankungen im Vergleich zu einer, sagen wir mal, wirklich von Entwicklern getriebenen Entwicklung ist sicherlich da.

EB01\_Eckhard Hauenherm: 15 - 15 (0)

natürlich, dadurch dass, also ich habe ja gesagt, das Citizen Development heißt nicht, dass die Leute keine IT-Kenntnis haben, aber oft ist es halt so, dass sie nicht von einem klassischen IT-Background kommen, sich Sachen selber beigebracht haben und dann halt Sachen auch nicht optimal umgesetzt werden. Was sich dann in Performance äußern oder im Datenstrukturen, die alles andere als optimal sind, gerade das, was man dann merkt, wenn sich Projekte erweitern. Ja, das sehe ich als die Hauptnachteile, also einmal Governance Aspekt und zweitens schlechte, schlechte Programmierung und Design. Also Design im Sinne von Design, aber auch im Sinne von Softwarearchitektur.

EB02\_Rosengrün, Robin: 15 - 15 (0)

1 - Vorbereitung Unternehmen\1.7 - Nachteile NCLC Plattformen\1.7.2 - Falsche Erwartungen

Wo ich sage, ist die Erwartungshaltung noch ein bisschen übertrieben, ist wirklich das Applikationen erstellen. Also das ich so Applikationen im Fachbereich erstelle, die extern zum Beispiel angewendet wird, in der Kundenkommunikation und so was. Da, glaube ich, sind wir noch nicht so weit und da weiß ich auch gar nicht, ob das wirklich so eine Zielsetzung sein kann.

EB01\_Eckhard Hauenherm: 19 - 19 (0)

Ich weiß aber, dass das in der Perspektive immer damit verbunden wird. Ich brauche keine Entwickler mehr, weil die Fachleute können das ja selber machen. Aber es ist ja nicht so, dass das Business nichts anderes zu tun hätte. Also das ist ja eine Milchmädchen-Rechnung

zu sagen, ich leiste die IT-Arbeit, die ich jetzt nicht leisten kann, die gebe ich in die Fachabteilung und die haben ja Zeit dafür, die können das ja machen. Das würde ja nicht hinhauen. Ich habe vielleicht den Vorteil, dass ich mit anderen Qualifikationen arbeiten kann. Also es geht weniger von der rein technischen Qualifikation IT-Wissen mehr in die Logik, Businesslogik, die ich auskenne. Also ich kann Betriebswirtschaftler mit einem guten technischen Verständnis sicherlich dann auch an so was dransetzen. Insofern kann das tatsächlich ein Nutzen sein, die Frage ist, haben wir nur einen Fachkräftemangel bei der IT oder haben wir nicht auch einen Fachkräftemangel in anderen Bereichen. Weil wie gesagt, ich würde auch nicht sagen, dass No Code/ Low Code so was ist, was jeder mit Kindergartenwissen soeben mal machen kann, sondern bisschen Verstand muss man da schon hereinbringen.

EB01\_Eckhard Hauenherm: 21 - 21 (0)

Und dann muss man nur wieder die Grenze ziehen, zu, bei uns wäre es jetzt klassisch Kundendatenverarbeitung plus Abrechnung oder so was. Das gehört dann, wenn ich 1000000 Kunden habe, auf jeden Fall nicht mehr in eine Low Code-Plattform, sondern da gibt es ja zu Recht dann halt große Anbieter, die da seit Jahrzehnten dran entwickeln und daneben ausgereifte Systeme haben, die das alles schon können.

EB02\_Rosengrün, Robin: 17 - 17 (0)

Man darf jetzt natürlich nicht erwarten, dass ich dadurch meine komplette interne IT einsparen kann, weil, ich habe ja jetzt alles Citizen Developer. Also der klassische Citizen Developer fängt jetzt nicht an, das nächste SAP ERP-System zu bauen. Da muss man auch ganz klar trennen, was ist denn wirklich No Code/ Low Code? Was sind die Einsatzgebiete dafür? Was sind auch die realistischen Ergebnisse gegenüber der klassischen IT? Und dann habe ich halt den ganzen Zwischenraum dazwischen.

EB03\_Fabian Gackstatter: 15 - 15 (0)

Also, ich glaube man muss ein bisschen differenzieren, was man mit No Code/ Low Code nicht machen kann, ich sage mal, ein SAP-System oder andere im Banken-, Kernbankensystem abzulösen.  
EB04\_Göttl, Simone: 15 - 15 (0)

Eine Erwartungshaltung ist, also wenn ich aus der Erfahrung spreche, die Hersteller, ich will nicht sagen, dass sie immer mehr versprechen, als am Ende möglich ist, aber es ist ein Werbeversprechen, was teilweise einfacher klingt, als es wirklich ist. Was ich viel in der Praxis sehe ist, dass es dann doch nicht so einfach ist. Und es gibt auch Hersteller, die bieten App in a Day an. Ja, ich kann eine App in einem Tag bauen, das heißt aber nicht, dass ich jeden Use Case in einem Tag bauen kann. Und das heißt, es ist ganz klar, was ich erwarten muss, es ist ein Erwartungsmanagement und das, was mir vielleicht versprochen wird, sollte ich gründlich hinterfragen, im Sinne, ist es wirklich so einfach und so leicht und sollte ich mich tiefergehend damit auseinandersetzen. Das heißt, auch wirklich Prototypen bauen oder Proof of Concept Phase machen und dort verschiedene Fachbereiche mit einbeziehen. Und nicht nur auf so App in a Day, was so ein vorgefertigter Tag ist, alles total einfach klingt. Also ich bleibe jetzt mal bei einem großen Möbelhersteller, wo wir das alle kennen. Das klingt immer so schön und so toll und so schön und dann sitzt man dann doch zu Hause und stellt fest: "Wo ist denn diese verdammte eine Schraube hin?"  
EB04\_Göttl, Simone: 21 - 21 (0)

Absolut. Ich merke halt immer wieder, wenn ich mit den Kollegen vom Fachbereich unterwegs bin, dass da, ich sage mal, grundsätzliches IT-Verständnis überhaupt nicht vorhanden ist. Weder wird verstanden, warum bestimmte Sachen gewartet werden müssen. Also absolute Basics, mit denen ich halt aufgewachsen bin, fehlen und das führt dann auch dazu, dass eine falsche Erwartungshaltung, zum Beispiel, an solche

Software gestellt wird oder im Falle des Misslingens, sage ich mal, nicht die richtigen Schlüsse gezogen werden. Sprich, dass dann die eigentliche Umsetzung das Problem ist. Also die Transformation der fachlichen Information in die IT-Technik. Das hier an genau der Stelle dann Zeitaufwände entstehen, die bei einem ITler nicht entstehen würden. Nur der ITler hat halt das fachliche Know-how nicht. Also es wird verschoben so ein bisschen, die Problematik.

EB05\_Rohrbeck, Christian: 17 - 17 (0)

Sie sind, was den Hype angeht, ist vielleicht der Hype dahin gehend, dass man sich vorstellt, dass man halt komplette IT-Prozesse oder IT-Software-Lösungen ablösen kann damit und daran glaube ich persönlich nicht.

EB05\_Rohrbeck, Christian: 21 - 21 (0)

1 - Vorbereitung Unternehmen\1.8 - Governance Einfluss

Als ein sehr wichtiger Punkt wird hervorgehoben, dass eine Governance vorhanden sein sollte für NCLC-Plattformen und die NCLC-Anwendungen. Es müssen Fragen beantwortet werden zu wer darf was? Wie wird die Sicherheit gewährleistet? Was geschieht mit den Daten? Wie wird eine Application Life Cycle gemanaged?

Die Plattformen sind trotz ihrer Standardisierung relativ offen und haben viele Möglichkeiten, mit Daten des Unternehmens zu arbeiten. Diese Fragen sollten geklärt werden und im Vorfeld bereits zumindest angesprochen werden, da ein hohes Risikopotential entstehen kann.

EB01\_Eckhard Hauenherm

Ein Risiko, das durch den Experten gesehen wird, ist das Entstehen einer Schatten-IT 2.0 und bezieht sich auf die fehlende Governance. Es sollte darauf geachtet werden, viel zu dokumentieren, um eine Nachvollziehbarkeit zu erhalten. Es sollte auch zu diesem Zwecke bei den Citizen Developern ein



Verständnis für IT-Wissen aufgebaut werden, wenn noch nicht vorhanden.

EB02\_Rosengrün, Robin

Es wird durch den Experten als Nachteil gesehen, wenn eine Governance fehlt. Dabei sollten Fragen geklärt sein, wer auf welche Daten wie zugreifen darf? Eine Fragestellung, die auch geklärt werden muss, ist, wie dürfen die Daten nach der Verarbeitung weiterverteilt werden?

Diese Governance-Regulierungen sollten sauber definiert sein und sich an die Wünsche und Bedürfnisse des Unternehmens anpassen sowie berücksichtigen, was wirklich möglich ist. Beschränke ich Citizen Developer auf die Arbeit mit Canvas-Apps und Sharepointlisten oder beschließt das Unternehmen, dass die User so viel technisches Verständnis haben, dass diese mit tiefergehende Systeme arbeiten können, welche wesentlich mehr Berechtigungen und Risiken mit sich bringen. In diesem Zusammenhang sollte sich auch Gedanken gemacht werden über die Administration einer solchen Plattform und wer diese Administration realisiert und durchführt. Es sollte sich ebenfalls Gedanken gemacht werden über das Thema Monitoring der Plattform, um einen Überblick darüber zu behalten, wie und ob die Plattform genutzt wird. Erste Gedanken sollten auch in die Richtung gehen, falls Tools nicht so genutzt werden, wie es erwünscht ist. Wie könnte ein solches Vorgehen in vorhandene unternehmensinterne Prozesse implementiert werden?

Ein weiterer Aspekt, über den sich im Vorfeld Gedanken gemacht werden sollte, ist, wie mit sehr guten Anwendungen umgegangen wird, die eventuell einen Mehrwert für das ganze Unternehmen bieten und diese dann anderen Unternehmensteilen zur Verfügung gestellt werden können.

Die negative Seite, die sich hier entwickeln kann, ist, was passiert, wenn ein MA so stark durch Citizen Development eingebunden wird, das er seine

primären Fachabteilungsaufgaben nicht mehr erfüllt.

Was nicht passieren sollte, dass ein Einführungsprojekt gestartet wird und sich diese Fragen erst während dieses Projektes gestellt werden. Dies kann zu erheblichen Verzögerungen im Projektverlauf führen.

EB03\_Fabian Gackstatter

Es wird auf die Schwierigkeit hingewiesen, innerhalb des Unternehmen das entsprechende Rahmenwerk zu etablieren. Dabei wird explizit auf das Thema Datenschutz und Zugriffsberechtigungen eingegangen sowie technische Schnittstellen zu anderer Unternehmenssoftware, aber auch, welche Rahmenbedingungen in Bezug auf Governance geschaffen werden müssen.

Es wird die Gefahr gesehen, dass bei fehlender Governance eine Schatten-IT entsteht, wenn die IT-Abteilung dort nicht ordentlich eingebunden wurde.

Dabei ist die Frage wichtig:

Welche Apps werden entwickelt und wie ist der App-Lebenszyklus?

Der Experte weist darauf hin, dass zur Lösung verschiedene Wege möglich sind. Einerseits ein Center of Excellence, wo Prozesse, Berechtigungen, welche Apps wer erstellen darf und Rahmenbedingungen im Allgemeinen festgehalten werden.

Eine zweite Möglichkeit ist es, den Mitarbeitern die Governance-Regularien mitzugeben, in dem sie geschult werden. Dabei wird hervorgehoben, dass dies ein laufender Prozess und keine einmalige Schulung ist.

EB04\_Göttl, Simone

Es wird die Problemstellung gesehen, dass aufgrund der fehlenden IT-Kenntnisse der Citizen Developer es zu verschiedenen Problemen kommen kann. Dabei ist es auch explizit nicht gewollt, dass die Citizen Developer sich mit IT-Modellen auseinandersetzen müssen, da dies dem Zweck von NCLC in gewissen Maße entgegenstehen würde. Um zu verhindern, dass allerdings

aufgrund dieses fehlenden Wissens ein Wildwuchs entsteht an Lösungen, die keiner IT-Struktur folgen und wo am Ende niemand mehr so richtig weiß, was und wie diese Apps etwas tun, sollte es eine IT-Governance-Strategie geben. Diese fehlende IT-Struktur kann sich auch nachteilig auswirken, wenn fachbereichsübergreifend eine App verwendet oder weiterentwickelt werden soll.

Zur Vermittlung der entsprechenden Governance-Richtlinien wird eine Lösung über Schulung gesehen. Wünschenswert ist, wenn die verwendete Plattform gewisse Einschränkungen schon als Basisfunktionalität besitzt.

EB06\_Hesse, Moritz

Governance wird als sehr wichtig angesehen, da die Befürchtung geäußert wird, dass durch eine fehlende Governance die Vorteile, die durch NCLC und Citizen Development entstehen, wieder aufgehoben werden und im schlimmsten Fall sich ins Negative verkehren. Dabei ist es wichtig, die Nutzer entsprechend mit einzubinden und die Governance ordentlich zu vermitteln.

Negative Aspekte, die geäußert werden, sind die Gefahr von Unübersichtlichkeit durch den Umstand, dass ohne Governance jeder entwickeln kann ohne Regeln und Bestimmungen.

Das Thema Governance sollte kein Thema sein, was Abteilungen intern regeln, sondern es sollte unternehmensweite Regularien geben.

EB07\_Janke, Tino

Es wird durch den Experten herausgestellt, dass Governance eine Grundlage für das erfolgreiche Anwenden von Citizen Development ist, wie die Rahmenbedingungen für erstellte Anwendungen sind und wie innerhalb und zwischen verschiedenen Abteilungen gearbeitet und Daten ausgetauscht werden. In dieses Thema fällt auch der Bereich der Datensicherheit, des Zugriffsmanagements und der Datenkonsistenz.

EB08\_Fey, Peter

1 - Vorbereitung Unternehmen\1.8 - Governance Einfluss\1.8.1- Fehlende Governance

Das Zweite ist natürlich aus Unternehmenssicht sehr stark, zu Fragen wie ein Application Life Cycle. Also wer betreut eigentlich die Anwendung weiter? Wer kümmert sich darum, dass die angewendet werden? Wer kümmert sich um die Sicherheit? Wo gehen die Daten eigentlich hin? Also all das, was so Governance, Application Life Cycle und so weiter ist. Das ist natürlich ein Punkt, über den man sehr stark nachdenken muss und das ist natürlich gerade bei der Einführung in Unternehmen dann eine der Kernfragen, mit denen ich auch gerade relativ häufig zu kämpfen habe. Oder zumindest zu reden habe, zu kämpfen habe ich damit weniger, aber das ist so ein Punkt, über den man explizit nachdenken muss, wie man damit umgeht. Das sind so die Nachteile, Anwendungsqualität, sage ich mal, und Einhaltung Governance.

EB01\_Eckhard Hauenherm: 15 - 15 (0)

Ja, wie schon eben erwähnt, die Governance. Das ist ein ganz wichtiger Punkt. Das hängt ein bisschen davon ab, mit welcher Plattform ich arbeite, aber die meisten Plattformen sind am Anfang immer relativ offen und bieten relativ viele Möglichkeiten und wenn ich bei der Einführung nicht darauf achte und erst mal mit offenem Scheunentor alle Anwender sozusagen da rauf lasse und die Daten von links und rechts verschieben lasse, auswerten lasse, irgendwelche Formulare dem Kunden zur Verfügung stelle, ohne an eine Datenschutzerklärung zu denken und solche Sachen. Da sehe ich tatsächlich eine Hürde, das kann gefährlich werden für ein Unternehmen. Also das ist jetzt keine Einführungshürde, aber es ist ein hohes Risiko, an das ein Unternehmen denken sollte.

EB01\_Eckhard Hauenherm: 35 - 35 (0)

Klar, also wenn es aus dem Citizen Development raus betrieben wird, dann ist natürlich die große Gefahr, dass wir Schatten-IT 2.0, nenne ich das immer, haben, also dass wir keine Governance aufgesetzt haben und das eigentlich, ja, sowas wie Excel Script ist, keiner weiß

Bescheid, es gibt keine Doku zu gar nichts. Das ist ein großes Thema  
EB02\_Rosengrün, Robin: 15 - 15 (0)

Die Nachteile hast du natürlich auch, wenn ich das Ganze nicht kontrolliert mache. Ich gebe ja dem Citizen Developer eine gewisse Fähigkeit an die Hand, eigene Applikationen zu bauen, auf die Daten des Unternehmens zuzugreifen. Wenn ich da nicht die richtigen Richtlinien platziere, sprich, auf welche Daten kann denn ein Citizen Developer zugreifen? Wie kann er die verteilen? Wie kann er die verarbeiten? Habe ich natürlich das Risiko, dass diese Daten irgendwo hinkommen, wo sie nicht hingehören, zum Beispiel.

EB03\_Fabian Gackstatter: 13 - 13 (0)

Was ein bisschen Hürden sind, ist natürlich, dort die Leitplanken innerhalb eines Unternehmens zu setzen. Das heißt, einerseits sicherzustellen, wir hatten das vorhin auch, Datenschutz, dass das erfüllt ist. Zugriffsberechtigungen sicherzustellen, wer darf welche Daten sehen? Wie kann ich aber auch diese No Code/ Low Code-Plattform an die bestehenden Bestandssysteme abbilden? Wo brauche ich APIs? Und natürlich auch die ganze Hürde, was ist eigentlich mein Rahmenkonstrukt? Meine Governance? Was erlaube ich überhaupt, um das machen zu können?

EB04\_Göttl, Simone: 17 - 17 (0)

Was aber auch ein Nachteil sein kann und deshalb, wenn so eine Plattform nicht gemanagt ist, No Code/ Low Code-Plattform ist, dass man immer mehr Schatten-IT bekommt. Das heißt, wenn ich nicht als IT zum Beispiel schaue, welche Apps werden entwickelt und die auch einem Lebenszyklus unterlasse, habe ich irgendwann ein Riesenwildwuchs und das ist definitiv ein Nachteil und ich laufe natürlich auch Gefahr, das ist auch so, dass ich an der einen oder anderen Stelle, wenn der derjenige, der das als Fachbereich entwickelt, nicht richtig geschult ist oder nicht richtig

ausgebildet ist, dass es am Ende doch länger dauert.

EB04\_Görtl, Simone: 17 - 17 (0)

Absolut, sehe ich auch im täglichen Geschäft in Anwendungen der Kunden unserer Lösung und zwar ist es so, wenn man quasi von der grünen Wiese, nein, das stimmt auch nicht unbedingt, aber man muss ein gewisses Maß an Struktur mitbringen, wenn man eine Software erstellt. Auch diese Aussage ist pauschal sehr schwierig zu bewerten man kann auch sehr dreckige Scripte schreiben, aber nichtsdestotrotz man muss sich anders mit Struktur und Modellierung und fachlichen Zusammenhänge zwischen den einzelnen Entitäten auseinandersetzen. Was bei Low Code und No Code-Lösungen möglicherweise auch gefordert wäre. Die Anwender bringen das aber häufig gar nicht mit und das liegt auch in der Charakteristik, quasi in der DNA von den Produkten selber, dass man ja eben genau diese Abstraktion haben möchte. Also man möchte den Nutzer ja gar nicht mit all diesen Dingen belasten, wie keine Ahnung, Entity Relationship-Modell. Was sind denn da für Zusammenhänge zwischen irgendwelchen Entitäten? Genau das möchte man dem Nutzer ja erleichtern. Aber das führt dann auch häufig zu, oder ich sehe hier die Gefahr einfach, dass sich diese Strukturlosigkeit, auch in den entwickelten Lösungen letztendlich wiederfindet und dass natürlich hier ein gewisser Wildwuchs entsteht, den es durchaus, ja ich sage mal, betrieblich würde man sagen, den es zu managen gilt. Also zu beobachten, zu planen, gegebenenfalls Richtlinien zu erstellen, Konventionen vorzugeben. Das ist durchaus eine Gefahr oder Nachteil, die ich hier sehe. Und insbesondere, wenn die Lösungen dann doch irgendwie wieder an die IT gereicht werden, also der User kann sich zwar selber seine App irgendwie zusammenbauen, aber jetzt soll die nicht als Insellösung irgendwo stehen, sondern wenn er da irgendwie die Daten eingegeben hat, sollen die dann bitte doch an umliegende Systeme weitergeleitet werden, die Daten. Und

spätestens dann trifft man ja doch wieder auf die IT und spätestens dann gilt es doch wieder die Strukturen zu finden, die entsprechenden Datentypen zu wrappen, alles so was und die quasi Schnelligkeit und Einfachheit und Flexibilität, die der Anwender in der Lösung hat, schlägt sich dann später bei der Integration, kann die dann wieder auf die Füße fallen, weil dann eben dann genau das, ja, dass was letztendlich der Vorteil der Lösung für den Anwender ist, kann sich dann als Nachteil wiederum ausgestalten bei der technischen Implementierung und insbesondere Integration.

**I: RS** Also den Nachteil, den Sie gerade beschrieben haben, würden Sie den unter dem großen Feld Governance quasi zusammenfassen?

**B: MH** Durchaus ja, ja das ist durchaus eine Governance-Aufgabe, oder eine gute IT Governance kann helfen, diese Probleme auf jeden Fall, ja, ich möchte nicht sagen zu verhindern, aber es abzuschwächen. Ja, dass man die Mitarbeiter entsprechend schult oder vielleicht auch die Technik, dass die Lösung, die Low Code-Lösung vielleicht sogar auch die Möglichkeit bietet, irgendwelche Best Practice zu implementieren oder Constraints oder irgendwelche Restriktionen zu hinterlegen. Ja, weiß ich nicht, dass eine bestimmte Komplexität nicht überschritten wird oder so. Aber ja, würde ich sagen, Governance ist hier ein Stichwort, was passen würde.

EB06\_Hesse, Moritz: 21 - 23 (0)

Ja, also auf jeden Fall gibt es da Nachteile, man muss halt, wenn man sich vorstellt, dass jetzt die Fachabteilung so eine Einkaufsabteilung, jetzt selber sich Anwendungen bauen kann und Prozesse digitalisieren kann, muss man halt schon sagen, dass es zu dem umgangssprachlichen Wildwuchs führen kann, wenn jetzt jede Abteilung für sich losmacht und für sich die besten Anwendungen baut. Das heißt, man braucht im Hintergrund ganz klar das Thema Governance, damit muss man sich auseinandersetzen, wie man

sozusagen das so ein bisschen orchestrieren kann, dass nicht jeder drauflos entwickelt, sondern dass das Ganze eben den gewissen Leitplanken folgt, weil es auch einfach schlussendlich IT-Security- Thema ist, wenn jeder was entwickelt. Niemand hält sich aber an gängige Vorgaben. Genau.

EB07\_Janke, Tino: 15 - 15 (0)

1 - Vorbereitung Unternehmen\1.8 - Governance Einfluss\1.8.2 - Governancestrategie

Die dritte Wissensbasis, die ich brauche, ist tatsächlich so ein Gefühl für Risiken und Governance, was den Umgang mit Daten angeht. Also die meisten Plattform sind extrem mächtig, was den Austausch von Daten zwischen unterschiedlichen Datenplattformen, Hostingplattformen und so weiter angeht. Und dass es vielleicht keine gute Idee ist, meinen aktuellen Geschäftsbericht in Drafte mal eben von, was weiß ich, von One Drive in die Dropbox zu kopieren, nur weil ich das automatisch machen kann und ich von da aus einen leichteren Zugriff habe. Das muss natürlich bei den Anwendern auch ankommen und das ist, was ich eben meinte, mit der Governance im Unternehmen. Da muss ich entweder eine Governance drüberlegen oder ich muss ein zumindest eine Awareness bei den Nutzern erzeugen dafür, dass nicht alles, was machbar ist, auch wirklich gut ist.

EB01\_Eckhard Hauenherm: 23 - 23 (0)

Also das was wir besprochen haben, das kann ich nur wieder als Anregung mitgeben, weil gerade im Cloudumfeld ist es natürlich so, ich setze einfach Lizenzen: "Ach ja, Power Platform ja, Power Automate schalte ich einfach mal an für alle Benutzer." Passiert da irgendwas, passiert da nichts, welche Konnektoren dürfen verwendet werden, welche dürfen nicht verwendet werden? Wer macht da was mit. Davon würde ich abraten, das bringt einfach nichts.

EB01\_Eckhard Hauenherm: 79 - 79 (0)

Klar, also wenn es aus dem Citizen Development raus betrieben wird, dann ist natürlich die große Gefahr, dass wir Schatten-IT 2.0, nenne ich das immer, haben, also dass wir keine Governance



aufgesetzt haben und das eigentlich, ja, sowas wie Excel Script ist, keiner weiß Bescheid, es gibt keine Doku zu gar nichts. Das ist ein großes Thema und natürlich, dadurch dass, also ich habe ja gesagt, das Citizen Development heißt nicht, dass die Leute keine IT-Kenntnis haben, aber oft ist es halt so, dass sie nicht von einem klassischen IT-Background kommen, sich Sachen selber beigebracht haben und dann halt Sachen auch nicht optimal umgesetzt werden. Was sich dann in Performance äußern oder im Datenstrukturen, die alles andere als optimal sind, gerade das, was man dann merkt, wenn sich Projekte erweitern. Ja, das sehe ich als die Hauptnachteile, also einmal Governance Aspekt und zweitens schlechte, schlechte Programmierung und Design. Also Design im Sinne von Design, aber auch im Sinne von Softwarearchitektur.

EB02\_Rosengrün, Robin: 15 - 15 (0)

Und das muss ich halt als Unternehmen auch sauber definieren. Wo möchte ich denn, dass meine Leute ihre Zeit, ihre Ressourcen verbringen? Sind das wirklich die klassischen Citizen Developer, wo ich sage, ihr habt SharePoint und Canvas-Apps? Das ist eure Spielwiese oder erweitere ich dieses Spektrum, weil ich sage: "Ich habe durchaus technisch affine User, auch in meinem Business, denen gebe ich auch mehr Teile zur Verfügung und stelle ihnen auch mehr Datentöpfe zur Verfügung, um wirklich eigene Applikationen zu bauen, die für ein ganzes Team oder eine ganze Abteilung dann noch zum Einsatz kommen können."

EB03\_Fabian Gackstatter: 15 - 15 (0)

Und dann muss ich natürlich auch die technische Grundlage sauber definiert haben. Also da kommen dann die Nicht-Entwickler aus meiner zentralen IT wieder zum Tragen, sprich die Power Platform-Administratoren, ein Active Directory-Administrator, der jetzt auch die richtigen Rechte tatsächlich freischalten kann und der auch das ganze Thema monitoring dann überwacht. Ich will ja als

Unternehmen wissen, was machen denn meine Citizen Developer überhaupt im System. Bauen die Applikationen? Bauen die Automationen? Nutzen die tatsächlich auch nur die Tools, die eigentlich dafür vorgesehen sind oder haben die aus Versehen irgendwelche Sachen offengelassen? Dass ich zum einen das monitoren kann, zum anderen aber auch darauf reagieren kann. Was passiert denn, wenn der Rocco jetzt eine Applikation gebaut hat und teilt die mit hunderttausend seiner Kollegen. Ist das so vorgesehen? Wollen wir das überhaupt? Und wie unterstützen wir Rocco dabei, wenn seine Applikation jetzt wirklich gut ist? Wie verknüpfe ich Rocco mit unserem zentralen Support? Weil, wenn ich jetzt eine Applikation baue und teile die mit meinem Team, mit fünf anderen Leuten, die kriege ich selber nach wie vor unterstützt. Wenn ich jetzt eine Applikation baue, die morgen von hunderttausend Leuten bedient wird, dann werde ich wahrscheinlich unter der Last der Issues und feature Request und Anfragen zusammenbrechen und kann meinen eigenen Tagesjob eigentlich nicht mehr machen. Also, über genau solche Themen muss ich mir Gedanken machen. Was sind denn die Citizen Developer-Applikationen? Wie, in welchem Rahmen sollen die sich bewegen? Und wann wird aus einer Citizen Developer-Applikation tatsächlich eine businesskritische, die im ganzen Unternehmen eingesetzt wird? Und das sind dann alles solche Gedanken, die mache ich mir idealer Weise im Vorfeld. Ich muss leider dazu sagen, die Hälfte unserer Kunden fängt erstmal damit an, dass sie sagt: "Wir haben jetzt die Lizenzen gekauft. Wir bauen jetzt mal eine Applikation." Und nach einem halben Jahr ist dann: entweder brennt die Hütte, weil die interne IT dann keinerlei Kontrolle mehr darüber hat, die weiß nicht, wo sie gucken muss. Sie weiß nicht, wie sie darauf reagieren soll und stellt im Zweifelsfall die Plattform wieder ein. Oder es passiert, dass eben die User nicht abgeholt werden. Es wird kein Training zur Verfügung gestellt. Es gibt keinen Feedback-Mechanismus, es

gibt keine Trainings und die Plattform hebt eben nicht ab und es werden keine Applikationen gebaut und die Lizenzen gammeln einfach vor sich hin.

EB03\_Fabian Gackstatter: 21 - 21 (0)

Center of Excellence für uns ist unsere Organisationsstruktur, die wir bei unseren Kunden etablieren möchten. Ein Center of Excellence für die Power Plattform zum Beispiel. Die entsprechend die Governance orchestriert. Die entsprechend die Tools innehat, die zur Verfügung gestellt werden. Die aber auch die Themen Changemanagement und Service Support zum Beispiel betreuen. Personell, was muss ich da an Prozessen auf die Beine stellen? Was muss ich an Tools zur Verfügung stellen? Was sind die Trainings? Was sind die Dokumente, die ich zur Verfügung stellen muss? Und wie könnten jetzt solche Lebenszyklen von Entwicklern und von Applikationen aussehen? All diese Themen fassen wir normalerweise zusammen unter einem Center of Excellence.

EB03\_Fabian Gackstatter: 27 - 27 (0)

Also es gibt verschiedene Formate. Es gibt dann so Center of Excellence, wo man so was verankern kann. Das heißt, einerseits die Governance verankert, also Prozesse, was darf ich, was darf ich nicht? Was ist erlaubt? Wie erfolgen die Prozesse für Freigaben? Welche Apps darf ich überhaupt programmieren? Da findet man häufig eine gewisse Expertise, dass man sagt: "Man hat so Leuchtturmmitarbeiter, die da natürlich auch helfen können." Nichtsdestotrotz würde ich bei einem Rollout immer die Mitarbeiter mitnehmen. Das heißt, deshalb sage ich auch bewusst mitnehmen, für mich ist das nicht eine einmalige Schulung, sondern für mich ist das eine fortwährende Enablement, Befähigung, was durch Trainings passieren kann, durch einzelne Learningnuggets, YouTube-Videos, aber auch vielleicht durch Communitys. Und deshalb ist es nicht nur Aufgabe des Center of Excellence, sondern muss man einfach auch schauen, wer am Ende auch der Product Owner ist und dass der

sein Produkt natürlich durch so einen Konzern und Haus trägt.

EB04\_Göttl, Simone: 19 - 19 (0)

Und ich muss auch zwingend mir Gedanken machen über das Thema Governance. Ansonsten ist es toll, dass die Menschen damit arbeiten können, aber ich glaube, dann geht mit der Zeit auch der Nutzen verloren, weil es einfach zu unübersichtlich wird, das Ganze zu verwalten und es geht nur miteinander.

EB07\_Janke, Tino: 47 - 47 (0)

Also mir bringt die beste Governance nichts, wenn ich die Menschen nicht mit an Bord hole, in dem Projekt, wenn es also niemand nutzt. Und wie gesagt, mir bringen die begeisterten Nutzer der Plattform nichts, wenn ich das nicht schaffe, in sinnvolle Leitplanken zu geben, diese Projekte.

EB07\_Janke, Tino: 47 - 47 (0)

Also Citizen Development, halt im Endeffekt, wenn man auf das Deutsche überträgt, den Begriff Fachbereichsentwicklung und es geht einfach darum, wenn ich jetzt einen Fachbereich ermächtige, große Workflows für die Abläufe auf IT-Basis für Unternehmen zu erstellen, dann muss ich halt, unabhängig davon, ob die Leute Code schreiben wollen oder nicht, muss ich die ganzen Umgebungsbedingungen, die muss ich halt auch bedenken. Das heißt, es geht um Datensicherheit bei der Richtung, dass einerseits die Sicherheit der Daten, aber andererseits die Verweigerung von unberechtigten Zugriffen auf die Daten, also das ist die Datenkonsistenz zum Beispiel, das gehört auch noch zur Datensicherheit. Das ist ein Aspekt, dann die Governance

EB08\_Fey, Peter: 9 - 9 (0)

Und der Unterschied zu Citizen Development, wenn ich die Frage jetzt richtig interpretiert habe, besteht halt darin, dass das Citizen Development, die Fachbereichsentwicklung, diese No Code/ Low Code-Plattformen als Basis der Realisierung voraussetzt, aber gleichzeitig den

Rahmen auch schafft, dass in einem Unternehmen die Richtung dessen, wie gearbeitet werden muss, einheitlich ist. Dass vorgegeben ist, wie Abteilungen untereinander zum Beispiel ihre Applikationen aufbauen, damit ein leichter Datenaustausch möglich ist.

EB08\_Fey, Peter: 11 - 11 (0)

1 - Vorbereitung Unternehmen\1.9 - IT Abteilung und NCLC

Der Experte ist der Meinung, dass IT-Abteilungen nicht durch NCLC und Citizen Development ersetzt werden können. Er vertritt die Meinung, dass diese eher als starke Unterstützungen wirken können und dadurch einen entlastenden Faktor bilden. Vorteilhaft ist dabei, dass mit anderem Wissen gearbeitet werden kann, als es in der klassischen IT vorhanden ist. Es entsteht der Zugriff auf fachbereichsspezifisches Wissen und Expertise. Dies ist aber unter der Voraussetzung, dass der Fachkräftemangel nicht ebenso drängend ist in anderen Abteilungen.

Ein weiterer Vorteil, den NCLC für IT-Abteilungen mitbringt, ist, dass die IT-Abteilung selbst die Plattformen verwenden kann und sich dadurch die Arbeit erleichtern kann oder für andere Abteilungen Anwendungen schnell erstellen kann.

EB01\_Eckhard Hauenherm

Der Experte trifft die Einschätzung, dass der IT-Fachkräftemangel nicht allein durch NCLC behoben werden kann, selbst wenn es möglich wäre, alle Entwicklungsaufgaben in diese Plattform auszulagern, da das Aufgabenfeld von IT-Abteilungen wesentlich größer ist. Dabei wurden Aufgabenfelder hervorgehoben wie Datenbankadministrierung, SAP-Administrierung und ähnliche Aufgaben.

Allerdings kann Citizen Development und NCLC die IT-Abteilung darin unterstützen, dass verschiedene kleine Entwicklungsaufgaben realisiert werden. Diese kleinen Entwicklungsaufgaben würden im Normalfall selten realisiert werden durch die IT-Abteilung, aufgrund des großen Aufgabenbacklogs der IT-Abteilung und der niedrigen

Priorisierung der Aufgabe. Die fertige Entwicklung des Citizen Developer könnte am Ende gereviewt werden durch die IT-Abteilung und dann eingesetzt.

Ein weiteres Feld, wo Citizen Developer die interne IT-Abteilung entlasten können, ist, wenn es darum geht, innovative Ideen auf der Plattform umzusetzen, wo normalerweise in erst Linie der klassische IT-Entwickler die Aufgabe übernehmen würde. Dabei geht es um kleine Anwendungen, die Fachabteilungen entlasten und durch einfache Mittel realisierbar sind. Die Bedingung dafür ist, dass die Komplexität nicht das Wissenslevel der Citizen Developer übersteigt.

Ein Punkt, der angemerkt wurde, wo in diesem Zusammenhang ein Mehraufwand entstehen kann, ist, wenn die IT-Abteilung und die damit zusammenhängende Supportorganisation nicht darauf vorbereitet ist, dass die Anzahl der Entwickler im Unternehmen stark steigt. Sollten keine entsprechenden Maßnahmen ergriffen werden, kann die Situation entstehen, dass IT-Abteilungen mit Supportanfragen zu stark belastet werden.

EB03\_Fabian Gackstatter

Der Experte äußert seine Zweifel daran, dass es zu einer Entlastung der IT-Abteilung kommt durch Citizen Development und NCLC. Diese Befürchtung stützt er darauf, dass es wahrscheinlich nicht weniger Aufgaben werden, sondern einfach nur die Aufgabenfelder sich verschieben durch zusätzliche Supportanfragen der Citizen Developer. Ein weiteres Aufgabenfeld, was dazu kommen würde, ist, dass das Monitoring der vielen neuen Lösungen die IT-Abteilung ebenfalls belastet. Es findet eine Verschiebung von Entwicklung einer Anwendung zum Support für die Anwendungsentwickler statt.

EB05\_Rohrbeck, Christian

Der Experte stellt heraus, dass durch die Fähigkeit der Citizen Developer, eigenen Lösungen zu entwickeln, die IT-

Abteilungen entlastet würden, da der Aufgabenbacklog sich verringert. Das könnte dazu beitragen, dass der Flaschenhals, der sich durch den IT-Fachkräftemangel gebildet hat, verringert wird.

EB07\_Janke, Tino

Der Experte stellt zwei Aspekte heraus, wie Citizen Development und NCLC die IT-Abteilungen entlasten können. Ein Citizen Developer, der gründlich ausgebildet wurde und als Power User der Plattform fungiert, kann die IT-Abteilung entlasten durch eine Verringerung des Aufgabenbacklogs. Dabei legt der Experte ausführlich dar, dass eine entsprechende vorherige Ausbildung auf diese Plattform wichtig ist. Ein zweiter Aspekt, der zur Entlastung der IT-Abteilung beitragen kann, ist, dass durch die Plattformen relativ schnell, ohne großen Programmieraufwand, Lösungen entwickelt werden können durch die IT-Abteilung. Es entfällt die Notwendigkeit von viel Programmieraufwand im Kleinen und eine IT-Abteilungen kann Lösungen schneller erstellen und sich danach wieder auf Probleme konzentrieren, die nicht durch NCLC abgebildet werden können.

EB08\_Fey, Peter

1 - Vorbereitung Unternehmen\1.9 - IT Abteilung und NCLC\1.9.1 - Negative Auswirkungen für IT

Oder falls meine Supportorganisation nicht darauf vorbereitet ist, dass da jetzt auf einmal statt zehn Entwicklern hundert Entwickler in meinem System unterwegs sind, die dann im Zweifelsfall geflutet werden von Serviceanfragen, von Supportanfragen, auf die sie nicht vorbereitet sind, habe ich natürlich auch ein Problem.

EB03\_Fabian Gackstatter: 13 - 13 (0)

Also ich denke, er wird ihn nicht komplett beheben können, denn die Anforderungen an einen IT-Mitarbeiter sind ja in der Regel mehr als nur eine reine Entwicklungsaufgabe. Also wenn ich mir angucke, was sind denn meine Fachkräfte in der IT? Das sind die Entwickler auf der einen Seite, das sind aber auch die

Leute wie Quality Assurance Administrator, das sind Leute wie Datenbank-Administratoren, SAP-Administratoren. Das sind ja jetzt nicht die Rollen, die ein Citizen Developer übernehmen wird in meiner internen IT.

EB03\_Fabian Gackstatter: 17 - 17 (0)

Also ich hole mal anders aus an der Stelle. Mal angenommen, man hat ein kleines Dorf und eine Durchgangsstraße, dann staut sich da der Verkehr, viele Leute sind verärgert und man beschließt, eine Umgehungsstraße zu bauen. Jetzt hätte man die Erwartung, dass die Umgehungsstraße dann den Verkehr aufnimmt und die Durchgangsstraße leer ist. Aber es führt dazu, dass insgesamt mehr Verkehr unterwegs ist, weil der ja flüssiger läuft jetzt. Was wieder zu Staus und Verstopfungen führt. Ich kann mir vorstellen, dass das dieses grundsätzliche Problem des Fachkräftemangels nicht oder nur teilweise tangiert oder entlastet. Also ich glaube nicht, dass das hier dazu geeignet ist, denn an dem einen oder anderen Punkt wird derjenige, der aus dem Fachbereich heraus eine solche Anwendung bedient, trotz allem auch Rückfragen haben. Es würde also eher dazu führen, dass mehr IT-Lösungen unterwegs sind und die mehr Human Resources an sich brauchen und am Ende keine Entlastung mit sich bringt, was das Thema angeht.

**I: RS** Also du würdest eher sagen, dass durch die Einführung, dass die Fachabteilungen selbst für sich aus ihrem Fachwissen heraus, kleine Anwendungen bauen können, dass dadurch ein gesteigertes Anfrageaufkommen für die IT kommt und quasi keine Entlastung, dass halt weniger Anfragen oder große Anfragen an die IT gestellt werden?

**B: CR** Ich bin davon überzeugt, dass die ein oder andere Systemanfrage wegfällt, aber ich glaube eher, dass der Support, den die IT zu leisten hat, also dass im Prinzip sich dadurch die Aufwände von Projektdurchführung eher in Richtung



Support verschieben, weil diese Kollegen dann gecoacht werden müssen in gewisser Weise, um einfach dann auch fertig zu werden. Also ich würde jetzt nicht sagen, dass es von der absoluten Gesamtheit mehr wird, aber es wird auch nicht spürbar zu einer Entlastung beiführen.

EB05\_Rohrbeck, Christian: 23 - 25 (0)

1 - Vorbereitung Unternehmen\1.9 - IT Abteilung und NCLC\1.9.2 - Entlastung IT-Situation

Ich weiß aber, dass das in der Perspektive immer damit verbunden wird. Ich brauche keine Entwickler mehr, weil die Fachleute können das ja selber machen. Aber es ist ja nicht so, dass das Business nichts anderes zu tun hätte. Also das ist ja eine Milchmädchen-Rechnung zu sagen, ich leiste die IT-Arbeit, die ich jetzt nicht leisten kann, die gebe ich in die Fachabteilung und die haben ja Zeit dafür, die können das ja machen. Das würde ja nicht hinhalten. Ich habe vielleicht den Vorteil, dass ich mit anderen Qualifikationen arbeiten kann. Also es geht weniger von der rein technischen Qualifikation IT-Wissen mehr in die Logik, Businesslogik, die ich auskenne. Also ich kann Betriebswirtschaftler mit einem guten technischen Verständnis sicherlich dann auch an so was dransetzen. Insofern kann das tatsächlich ein Nutzen sein, die Frage ist, haben wir nur einen Fachkräftemangel bei der IT oder haben wir nicht auch einen Fachkräftemangel in anderen Bereichen. Weil wie gesagt, ich würde auch nicht sagen, dass No Code/ Low Code so was ist, was jeder mit Kindergartenwissen soeben mal machen kann, sondern bisschen Verstand muss man da schon hereinbringen.

EB01\_Eckhard Hauenherm: 21 - 21 (0)

Da sind fünf Informationen drauf und wenn ich die ausgefüllt habe, scanne ich das Ding ein und schicke es an eine E-Mail-Adresse. Da wird dir niemals eine interne IT eine Applikation dafür bauen, weil du dann immer so weit unten im Stapel des Notwendigen landen wirst, dass es wahrscheinlich niemals so weit kommt. Die interne IT kümmert sich um die SAPs dieser Welt, um die Secret Server dieser Welt, um die Dynamics

Solutions dieser Welt, aber gerade die kleinen Prozesse, die bleiben relativ häufig liegen, wenn ich dann nicht einen dedizierten Entwickler habe, der sich eben um diese kleinen Themen kümmert. Der Citizen Developer, der kann tatsächlich selber diese Sachen bauen. Die können dann im Zweifelsfall von der IT noch einmal gereviewt werden, bevor sie dann zum produktiven Einsatz kommen, nehmen aber den Großteil der Arbeit der IT einfach ab.

EB03\_Fabian Gackstatter: 15 - 15 (0)

Wo der Citizen Developer die interne IT definitiv entlassen wird, ist es, wenn es wirklich darum geht, Innovationen umzusetzen, Ideen umzusetzen, die von außen kommen, die von innen kommen. Die tatsächlich in Applikationen übersetzt werden müssen, wo in der Regel erst mal der klassische IT-Entwickler in der Pflicht steht. Den kann ich auf alle Fälle durch die Citizen Developer entlasten, weil auf einmal sein Stack, was er abarbeiten muss, sind wirklich die Dynamics, die SAP-Lösungen dieser Welt, aber eben nicht mehr den Speiseplan für die Kantine zu programmieren, den kann auch ein Citizen Developer zusammensetzen. Oder das Weihnachtswichtelspiel oder den Approval-Prozess für einen Urlaubsantrag. Alles Themen, die heute erst mal in der IT landen, bei irgendeinem Entwickler auf dem Tisch. Einen Großteil davon kann ich tatsächlich an die Citizen Developer geben, weil sie eben von der Komplexität her relativ klein gehalten sind und kann damit zumindest den Fachkräftemangel in der Entwicklung entgegenwirken.

EB03\_Fabian Gackstatter: 17 - 17 (0)

Und natürlich könnte man jetzt auch sagen, die überlasteten IT-Leute werden da entlastet. Aber das sind alles Nebenvorteile, sage ich mal. Hauptsächlich geht es darum, dass keine Reibungsverluste entstehen

EB05\_Rohrbeck, Christian: 15 - 15 (0)

Man muss sich ja vorstellen, bisher obliegt die Umsetzung aller digitalen Produkte, aller Prozesse, obliegt momentan der IT-Abteilung, wenn man sich da mal so umschaute, ist es eigentlich so, dass die IT-Abteilungen einfach ein riesiges Backlog mit sich her schleppen. Das heißt, große Anzahl an unbearbeiteten Aufgaben, an unbearbeiteten Anforderungen. Das ist so eine Art Flaschenhals geworden, weil es auch einfach an Fachkräften mangelt. Genau. Durch den Einsatz von Low Code/ No Code möchte man erreichen, dass man diesen Flaschenhals so ein bisschen auflöst, indem man Kollegen, die eigentlich nicht programmieren können, in die Lage versetzt, ihre, ihren Digitalisierungsbedarf selber vornehmen zu können.

EB07\_Janke, Tino: 13 - 13 (0)

1 - Vorbereitung Unternehmen\1.9 - IT Abteilung und NCLC\1.9.3 - Fachkräftemangel IT

Also ich glaube nicht, dass man die IT damit ersetzen kann. Ich glaube, dass man die IT damit sehr stark unterstützen kann. Tatsächlich ist meine Erfahrung aus den Unternehmen, dass Low Code/ No Code schwerpunktmäßig in der IT eingesetzt wird und weniger in den Fachbereichen tatsächlich, also das, was man eigentlich will, da sind wir, glaube ich, noch nicht.

EB01\_Eckhard Hauenherm: 17 - 17 (0)

Also zwei Richtungen muss man hier bedenken. Einerseits aus dem Citizen Development, wäre ja die Fachbereichsentwicklung. Das heißt, wenn ich jetzt jemanden habe im Fachbereich, der sagt: "Mensch ich finde das super. Ich würde mich gerne in so eine Low Code/ No Code-Plattform einfuchsen." Wobei dieses einfuchsen eben nicht zwischen Suppe und Kartoffeln passieren kann, sondern da muss man tatsächlich einige Wochen, sozusagen Schulungen, vielleicht neben der Arbeit, aber halt doch Schulungen einplanen. Das ist ähnlich, wie wenn man Excel sehr, sehr intensiv nutzen will und zum Power-User werden möchte, ist das dann keine Sache, die man innerhalb von einem Tag lernt. Man kann sich innerhalb eines Tages annähern. Man kann auch schon die ersten

Schritte gehen. Das auf jeden Fall, aber wenn man sagt, ich strebe an, eigentlich einen größeren Workflow abzubilden, dann braucht man schon einige Wochen Schulung und Erfahrung auch mit diesem Tool, auch wenn es dann schon eher in die Programmierung geht. Insofern, wenn ich dann jemanden aus dem Fachbereich habe, dann ist es ganz klar, dann habe ich oder benötige ich niemanden mehr aus der IT-Abteilung, der das programmiert. Insofern geht ja die Rechnung relativ leicht auf. Auch wenn dann die Möglichkeiten, den Fachkräftemangel in der IT zu beseitigen, dann nicht so ganz umfangreich ist. So, das ist eine. Man kann es, wenn auch jetzt nicht so zu hunderten, dann durch einen Fachbereich, die Leute dann ersetzt werden können. Aber man kann es. Auf der anderen Seite und das ist, glaube ich, der charmantere Teil der Sache. Ich sagte es ja gerade, durch das Anbieten von Funktionen, das Anbieten von in Alt-IT-Sprache ausgedrückt, von vorgefertigten Bibliotheken, also man muss sich dann auch die No Code/ Low Code-Plattformen als eine riesige Anreihung von Bibliotheken vorstellen und man zieht die über grafische Module, zieht man die dann in den aktiven Code hinein. Dadurch erspare ich natürlich halt auch den Programmierern viel Einzelarbeit. Also wenn ich in der Lage bin zu sagen: "Ich hätte gerne eine Anbindung an ein Kreditinstitut und ich wähle durch das Anklicken der entsprechenden Kreditinstitute oder der Karten, die ich nutzen möchte, wähle ich alles aus und diese Funktion kann ich dann per Drag-and-drop in meinen Code hineinziehen oder in meinen Ablauf hineinziehen", dann ist das natürlich auch eine enorme Erleichterung für die Programmierer und dadurch kann man natürlich auch sagen, der Arbeitseinsatz in der IT kann dann darauf fokussiert werden, was wirklich durch diese Plattform nicht abgedeckt werden kann. Der kann darauf fokussiert werden, was tatsächlich an neuen Dingen im Unternehmen zu regeln und zu lösen ist. Und durch die Entlastung auf der einen Seite habe ich dann mehr Zeit, mich auf die anderen

Dinge zu konzentrieren. Das könnte halt auch zu einer großen Erleichterung in diesem IT-Bereich führen. Und ja, das sind so die, die beiden Aspekte. Also No Code/ Low Code, auch wenn Citizen Development Fachbereichsentwicklung heißt, wird halt eben nicht nur in den Fachbereichen eingesetzt, sondern halt auch in den IT-Abteilungen und das ist meines Erachtens auch eine gute Sache.  
EB08\_Fey, Peter: 17 - 17 (0)

2 - EP: Eigenschaften Unternehmen\2.1 - Mindset

Mindset würde ich schon sagen, weil Low Code/ No Code ist natürlich sozusagen schon eine sehr agile Form des Arbeitens. Ich mache halt Sachen selber, die ich vorher vielleicht ausgelagert habe. Also ich muss mit flexiblem Rollenverständnis rangehen. Also Mindset, im Sinne von: so stark Hierarchie-abgegrenzte Rollen und so weiter, da funktioniert das meistens nicht, sondern ich muss sowohl aus, sagen wir mal, aus IT oder aus Operationssicht ein Verständnis dafür haben. "OK wir machen das jetzt selber und wir vertrauen den Anwendern auch so weit." Und gleichzeitig müssen die Anwender sagen: "Ok, ich übernehme auch Verantwortung für das, was ich hier tue." Also da würde ich schon sagen, dass das ein gewisser Mindset ist, der da existieren muss oder der herausgearbeitet werden muss. Das kann sich auch mit der Zeit entwickeln, das muss vielleicht nicht von Anfang an da sein, aber ich würde eben aus genau dieser Berücksichtigung so was nicht einfach flächendeckend im Unternehmen verteilen, sondern schon eine Einführungsstrategie mir überlegen, in welchen Ecken ich anfangen damit, um auch zu testen und zu probieren. Das würde ich schon sagen  
EB01\_Eckhard Hauenherm: 27 - 27 (0)

Mindset wird durch den Experten als wichtig eingestuft. Dabei wird hervorgehoben, dass schon aufgrund der Agilität der Plattform ein agiles Mindset wichtig ist. Starres Hierarchiedenken und festgelegte Rollenverständnisse werden als starke Hürde wahrgenommen. Wichtig ist dabei, dass aus Unternehmens- und IT-Sicht Vertrauen in die Anwender selbst gesetzt wird und aus Anwendersicht sollte ein Verantwortungsbewusstsein vorhanden sein. Es ist keine zwingende Voraussetzungen, dass dieses Mindset bereits im Unternehmen vorhanden ist, es kann sich auch während des Einführungsprojekts entwickeln, sollte aber Teil der Einführungsstrategie sein.

EB01\_Eckhard Hauenherm

Mindset wird als relevant betrachtet. Dabei stellt der Experte heraus, dass ein agiles Mindset wünschenswert ist. Dies bezieht er besonders auf die Denkweisen, die nach der Einführung angewendet werden sollten in der Zusammenarbeit mit den Citizen Developern. Es darf nicht so sein, dass von

Anfang perfekte Programme erwartet werden, die im Idealfall eine Unternehmenssoftware ersetzen. Es muss sich darauf eingestellt werden, dass Fortschritte langsam und stetig passieren und dabei auch eine gute Fehlerkultur entwickeln, wenn etwas Negatives passiert.

EB02\_Rosengrün, Robin

Ok, also als Grundlage, das Mindset muss natürlich passen. Also das sich ein Unternehmen wirklich darauf einstellt: "Wir möchten No Code/ Low Code und das Thema Citizen Development einsetzen." Das ist halt ein drastisches Umdenken, im Sinne von: Ich habe eine interne IT-Abteilung, die sich um alle IT-Themen kümmert. Bei denen, das breche ich auf. Alles was jetzt wirklich an IT-Anfragen reinkommt, muss ich eben unter dem Gesichtspunkt betrachten: Ist das nach wie vor ein Thema für meine klassische IT? Oder sind das tatsächlich Themen, die auch Citizen Developer erledigen können? In der Regel kommt tatsächlich der meiste Input für diesen Stream von den Citizen Developern selber, in dem sie sagen: "Wir haben hier Ideen, wir haben hier Probleme im Unternehmen, die relativ einfach zu adressieren wären, mit denen sich die interne IT aber nicht beschäftigt." Also, ich sage mal, das Mindset muss auf alle Fälle passen. Das Vertrauen gegenüber den Mitarbeitern muss auch da sein, dass man sie tatsächlich jetzt an Themen ranlässt, die klassischerweise jetzt nicht bei ihnen gelegen haben. Das setzt natürlich auch voraus, das ich intern die richtigen Weichen stelle. Also ein Betriebsrat, zum Beispiel, muss auch involviert sein. Wir haben Kunden, wo der Betriebsrat eben so weit weg davon ist, dass der sagt: "Alles was als Applikation im Unternehmen ausgerollt oder entwickelt wird, muss durch einen Betriebsrat genehmigt werden." Was für einen klassischen IT-Bereich natürlich ok ist. Was aber das Citizen Development krass ausbremsen würde, weil, wenn jetzt jeder anfängt, Applikationen zu bauen und jede einzelne Applikation vom Betriebsrat genehmigt

Der Experte sieht das Mindset als wichtig an. Dabei stellt er Aspekte heraus wie Commitment zur Plattform und dem Willen, diese einzusetzen. Er erwartet, dass bei Unternehmen ein Umdenken wichtig ist, in Bezug auf ein Entfernen von der Denkweise, dass IT-Themen nur durch die IT-Abteilung erledigt werden können. Es muss neu evaluiert werden im Falle, dass ein IT-Thema bearbeitet werden muss, ob die IT-Abteilung oder die Citizen Developer das erledigen. Ein weiterer Aspekt ist eine Offenheit für Vorschläge von Citizen Developern.

Es wird hervorgehoben, dass zu einem guten Mindset auch Vertrauen in die Citizen Developer zugehörig ist. Es wird darauf hingewiesen, dass nicht nur technische Aspekte eine Voraussetzung für die Einführung sind, sondern dass ein Veränderung im Denken stattfindet, das sich auf die Aspekte Kommunikation im Unternehmen, Bereitschaft zur Veränderung von Richtlinien und Anwenderbefähigung erweitert.

EB03\_Fabian Gackstatter

werden muss, ist dein Betriebsrat auf einmal der totale Flaschenhals, weil, dafür sind die ja nicht ausgelegt, auf einmal von hunderttausend Leuten hunderttausend Applikationen zu sichten und zu genehmigen, zum Beispiel. Es reicht nicht nur, einfach die technische Basis zu schaffen und zu sagen: "Wir haben jetzt die Plattform gekauft. Wir haben die scharf geschaltet. Hier sind die Lizenzen, fangt an zu bauen." Sondern ich muss auch intern vom Mindset her und auch vom Changemanagement mich als Unternehmen darauf einstellen, dass die richtigen Richtlinien gesetzt sind, dass ein Betriebsrat involviert ist, dass auch die richtige Kommunikation und das richtige Training für meine User zur Verfügung steht. Weil, es nutzt nichts, einfach nur die Plattform hinzustellen und dann sagen: "Seht zu, wie ihr klarkommt. Also hier sind von mir aus noch 2 Microsoft-Links, da könnt ihr euch mal Trainings anschauen." Sondern ich muss mir wirklich Gedanken machen, wie fördere ich denn auch meine Citizen Developer, im Sinne von Communities, im Sinne von Champions, die diese Communities unterstützen und auch im Sinne von: wie kann denn die interne IT dabei unterstützen. Also, wie können die, zum Beispiel, Trainings organisieren. Wie können die Community Meetings übernehmen. Wie können die Fragen beantworten, die die Citizen Developer an sie stellen. Also diese Einsatzplanung muss ich entsprechend auch anpassen.

EB03\_Fabian Gackstatter: 21 - 21 (0)

Was es aber bedarf ist, wie Sie es gesagt haben, auch der Mindset, aber das Mindset, glaube ich, kommt von 2 Seiten. Einerseits von der Belegschaft selber, dass man dort auch wirklich eine Akzeptanz fördert und dort auch deutlich macht, was die Vorteile sind oder in gewisser Weise, man wird nicht alle überzeugen, gerade bei großen Unternehmen, sondern ein paar Leuchttürme hat, die dafür stehen und die anderen mitziehen. Das Zweite ist aber, was ganz wichtig ist, das Commitment von oben. Was meine ich damit? Es muss natürlich der

Die Expertin sieht den Punkt Mindset als sehr wichtig an. Dabei bezieht sie zwei Seiten im Unternehmen ein. Auf der einen, dass die Mitarbeiter die Plattform akzeptieren und die Vorteile der Plattform auch wahrnehmen. Dabei wird anerkannt, dass nicht jeder überzeugt werden kann, aber es begünstigend wirkt, wenn ein paar Leuchttürme im Unternehmen dafür einstehen und dort mitziehen.

Die andere Seite ist das Unternehmen selbst. Es sollte in jedem Fall ein Commitment der Unternehmensleitung oder

Vorstand bzw. ein IT-Vorstand auch für diese Transformation stehen und auch sagen, das ist etwas, was wir jetzt machen und es darf ihm nicht aufdoktriniert sein. Das heißt, am besten, was heute gut funktioniert und was ich schon gesehen habe, ist das einzubetten in eine digitale Agenda oder in eine grundsätzliche Digitalisierungsstrategie. Und wenn es da mit drin ist und verankert ist, dann funktioniert das auch gut.

EB04\_Göttl, Simone: 23 - 23 (0)

des IT-Vorstandes vorhanden sein, dass intrinsisch ist und nicht vorgegeben.

EB04\_Göttl, Simone

Also ich glaube, was ich halt dadurch, dass ich auch freiberuflich unterwegs bin und schon die eine oder andere Firma gesehen habe, ist erst mal aus meiner Sicht die Firmenkultur. Hat man in der Firma eine Willkommenskultur, ist man offen für neue Dinge, das färbt spürbar ab auf die Mitarbeiter, die da vor Ort unterwegs sind. Und mal, gesetzt den Fall, es handelt sich um eine Firma, die wirklich gut mit ihren Mitarbeitern umgeht, sehr offen ist und diese Form des Know-how-Aufbaus auch unterstützt, dann wären erst mal da die wichtigsten Einstiegsgrößen getan. Ich muss auch sagen, in meiner Erfahrung, dass Kollegen ab einem gewissen Alter, ich möchte jetzt gar nicht so sehr auf einzelne, aber so rein vom Durchschnitt her, kommt man ab einem gewissen Alter so in eine Idee rein, dass man sagt: "Ich hab das schon immer so gemacht" bzw. ich erlebe das sehr oft, dass ich diese Antwort bekomme, dass man das schon immer so gemacht hat und dass dadurch das Verständnis für neue Voraussetzungen fehlt. Also, Alter spielt durchaus hier eine Rolle. Aber wie gesagt, das möchte ich gar nicht zu sehr pauschalisieren. Es gibt natürlich immer Ausnahmen. Ja, es gibt viele Details, die für ein erfolgreiches Projektteam sprechen. Das kann man auch verallgemeinern, egal, um welche Art von Projekt es sich handelt. Persönlich bin ich auch der Meinung, dass, zum Beispiel, eine Geschlechtermischung gut ist. Ich merke das immer wieder, wenn wir irgendwie Frauen im Team haben, dass dann doch noch mal eine andere

Der Experte stellt heraus, dass eine Willkommenskultur wichtig ist. Wird diese Kultur für neues wirklich gelebt, färbt dieses auch auf die Mitarbeiter ab und erleichtert die Einführung und die Akzeptanz der Plattform. Der Experte sieht aber auch auf Seiten der Mitarbeiter den Bedarf, offen für neues zu sein und nicht an alten Vorgehensweisen festzuhalten. Es sollte auch von Seiten der einzelnen Abteilungen ein entsprechendes Commitment zum Projekt herrschen, da diese zumindest teilweise durch das Projekt beeinflusst werden, wenn Mitarbeiter der Abteilung Teil des Projektteams sind. Es wird hier explizit auf die Gefahr hingewiesen, dass Projektmitglieder bei fehlendem Commitment der Abteilungsleiter sich in Entscheidungssituation immer für den Linienvorgesetzten entscheiden und ihre dortigen Aufgaben in der Abteilung bevorzugt wahrnehmen. Dadurch kann das Projekt behindert werden.

EB05\_Rohrbeck, Christian



Stimmung da ist, die auch mehr auch auf dem Fokus ist. Das tut gut. Dass halt einfach da der Projektansatz der Richtige ist, sei es jetzt agil oder Waterfall. Je nachdem, was da halt dann sinnvoll ist, wie das Teams drauf ist, eignen sich da verschiedenste Methoden. Was auch gut ist und das merke ich halt in größeren Unternehmen, die eine relativ hierarchische Einteilung haben, wenn die Abteilungsleiter voll und ganz dahinterstehen. Das größte Problem, was ich zum Beispiel im Moment habe, ist einfach das, dass wenn wir in unserer Matrixorganisation, in der ich gerade unterwegs bin, vom Projekt her, ich sage mal, quer durch Abteilungen stechen und uns Mitarbeiter zu 50 % ausleihen für ein Projekt, wird dieser Mitarbeiter im Zweifel immer seinem Linienvorgesetzten den Vorrang geben, das heißt, ich als Projektmanager ziehe immer den Kürzeren. Das heißt, ganz wichtig ist hier, ich nenne es mal, das Top-Management Commitment oder das Management Commitment, dass die Abteilungsleiter ihre Leute dafür auch das machen lassen und nicht für ihre originären Aufgaben abziehen. Verständnis und die richtige Umgebung, das richtige Umfeld, sind entscheidend. Jetzt noch mal kurz zurückgefragt, ich habe mich, glaube ich, ein bisschen verloren. Habe ich deine Frage damit beantwortet oder ist da jetzt noch Punkt offen?

**I: RS** Ich würde das jetzt mal unter dem großen Topic Mindset einordnen.

**B: CR** Ok

EB05\_Rohrbeck, Christian: 27 - 29 (0)

aber entscheidend ist tatsächlich und das merke ich immer wieder im täglichen Leben, ist eher schon so die emotionale Seite. Also so diese Softskills, weil die in Regel das Thema haben, dass die Kollegen für ihre Themen Experten sind. Die wissen genau Bescheid. Es gibt nichts, was ich da irgendjemand sagen kann in

der Regel. Das ist auch gar nicht mein Ziel, aber woran das halt scheitert, sind diese rein menschlichen Themen. Da redet A nicht B oder, ein absoluter Klassiker ist: A schreibt B eine E-Mail. B antwortet nicht, weil er denkt: "Ist ja gar nicht mein Thema." Also er gibt einfach kein Feedback oder vergisst es. A wartet, die Zeit verstreicht und so gehen immer wieder ganz viele gute Sachen verschütt und die dann wieder rauszuholen, dass voranzubringen, das ist wirklich schwierig. Weil viele auch ein bisschen auf Zeit spielen, sage ich mal und das ganze Thema kann man, glaube ich, nur dann dem begegnen, wenn man da ein gutes Klima schafft. Ein freundschaftliches Klima, wo die Leute auch aktiv Lust haben, miteinander zu reden, sich auch auszutauschen. Das ist, glaube eine der wichtigsten Aufgaben mit. Dass diese Seite, diese nicht messbare Seite, gut passt und dann könnte es auch ein Selbstläufer werden.

EB05\_Rohrbeck, Christian: 77 - 77 (0)

Sie haben das Mindset angesprochen und deshalb gehe ich da gerne noch mal drauf ein. Ich würde es als sehr begünstigte Komponente beschreiben, die, wenn man die mitbringt, das Richtige in Führungsstrichen, das Richtige, aber ein Mindset, was die Nutzung von No Code/ Low Code begünstigt, wenn man das mitbringt, ist das durchaus vorteilhaft zur Nutzung. Ich würde es aber nicht als Voraussetzungskriterium definieren.

EB06\_Hesse, Moritz: 33 - 33 (0)

Also, ich glaube, 2 wichtige Faktoren sind da, dass einfach ein gewisses Mindset da ist in der Organisation, dass man also sagt, das ist ja ein vergleichsweise neues Thema. Low Code/ No Code, man muss ein bisschen neue Wege gehen bei dem Thema, weil man Zuständigkeiten umverteilt. Man muss mit der IT-Abteilung ins Rennen gehen und dieses Umdenken, was es eben benötigt. Dafür braucht man ein gewisses Mindset. Wenn man zu traditionell in seiner Denkweise ist und da auch nicht alle Leute motivieren kann, ist das, glaube

Das richtige Mindset wird durch den Experten als begünstigend wahrgenommen.

EB06\_Hesse, Moritz

Der Experte stellt heraus, dass ein bestimmtes Mindset als positiv zu bewerten ist. Dieses Mindset liegt auf der einen Seite darin, die Bereitschaft als Unternehmen zu haben, Neuerungen zu akzeptieren und umzusetzen. Ist man zu starr in seiner Denkweise, kann es passieren, dass die Mitarbeiter nicht motiviert werden und das wirkt kontraproduktiv für die Einführungsprojekte. Eine entsprechende Fehlerkultur wird ebenfalls als positiver Aspekt gesehen und die Bereitschaft, dass NCLC sich erst entwickeln muss.

ich, ein Thema, was einen schwierigen Stand haben wird. Demgegenüber braucht man einfach so eine gewisse Offenheit für Neues und man muss dem nicht nur offen gegenüberstehen, sondern man muss dem Thema halt auch mal Raum geben, sich zu beweisen also, dann braucht es ein gewisses Maß an Aufwand, der da initial investiert werden muss, damit man Ergebnisse sehen kann und über die Ergebnisse kann man dann auch schneller einschätzen, das jetzt gutes Thema ist für mich, ob es mir helfen kann. Oh, also dieses Mindset-Thema sehe ich als sehr wichtig an.

EB07\_Janke, Tino: 19 - 19 (0)

Auf der anderen Seite, glaube ich, auch so eine gewisse Herangehensweise, wie ich mit meinen Mitarbeitern umgehe. Also da fallen mir so die Themen kontinuierliches Lernen ein, das man eben sagt: Ich muss meinen Mitarbeitern zutrauen, dass sie dieses mehr an Möglichkeiten auch sinnvoll umsetzen und ich muss eben auch da den Aufwand in Kauf nehmen, dass sich die Mitarbeiter erst mal auf die neue Plattform, also sie müssen lernen, sich darin zu bewegen in dieser Plattform und einfach Wissen aufzubauen in dieser Plattform. Diese beiden Punkte,

EB07\_Janke, Tino: 19 - 19 (0)

Mindset, weil Mindset das Allgemeine ist. Unternehmensgröße, Kapital was man hat, das ist dann halt doch immer relativ individuell und von Unternehmen zu Unternehmen unterschiedlich, aber was eben nicht so unterschiedlich sein kann und sein darf, ist das Mindset. Das Mindset ist ein kooperatives. Um eine Fachbereichsentwicklung anzustoßen, brauche ich natürlich die Möglichkeit, kooperativ mit meinen Kolleginnen und Kollegen im Fachbereich zusammenzuarbeiten. Das heißt, der Mindset muss so gestaltet sein, dass ich in der Lage bin, Probleme, die ich habe, die vielleicht unabhängig von dem fachlichen liegen, die eher im Ablauf im IT-Bereich liegen, zu adaptieren. Dass ich die übertragen kann in mein Arbeitsgebiet. Dass ich in der

Weiterführend geht der Experte auf den Umgang mit den Mitarbeitern ein. Er sieht es als positiv an, wenn ein Unternehmen den Mitarbeitern vertraut, so eine Anwendung auch alleine umzusetzen und ihnen die entsprechenden Ressourcen zur Verfügung stellt zur Befähigung.

EB07\_Janke, Tino

Der Experte sieht Mindset als sehr wichtig. Er hebt besonders hervor, dass eine Bereitschaft zu Lösung gemeinsamer Fragestellungen vorhanden sein sollte. Daraus wird geschlussfolgert, dass die Mitarbeiter gewissen Freiheiten haben sollten. Vorgesetzte sollten Lösungsansätze der Mitarbeiter als positiv wahrnehmen und diese nicht abwerten. Die Mitarbeiter sollten ermutigt werden, sich selbst Lösungen einfallen zu lassen und in diesem Zusammenhang wird auch eine gute Teamführung wichtig. Das sollte so gut implementiert sein, dass Mitarbeiter keine Angst haben, ihre Lösungen zu präsentieren, auch wenn diese originär eigentlich keine IT-Mitarbeiter sind und einen anderen Background haben. Dadurch

Lage bin, mit anderen diese Fragestellung zu lösen. Wenn ich aber in der Lage sein soll, diese Fragestellung zu lösen, als Mitarbeiter innerhalb einer Firma, dann brauche ich auch gewisse Freiheiten, die ich umsetzen kann. Das heißt, die Chefetagen müssen so vorgehen, wie Jobs das einmal in diesem häufig kolportierten Satz gesagt hat: "Wir stellen nicht gute Leute ein, damit wir ihnen sagen können, was sie tun sollen. Sondern wir stellen gute Leute ein, damit die uns sagen, was sie tun sollen." Das heißt, der Chef nimmt sich dann etwas zurück und geht dann in eine ganz starke, aktive Position des Zuhörens hinein und sagt: "Welche Lösung bietest du mir an? Was könnte denn passieren?" Das heißt, ich ermutige und ermächtige die Menschen, da etwas zu tun. Mindset in diesem Sinne fasst dann für mich eigentlich auch eine sehr, sehr gute Teamführung zusammen. Oder beinhaltet die auch. Und gerade da dann eher diese Begrifflichkeiten, die dann manchmal etwas belächelt werden, aber diese Psychological Safty ist eine ganz wichtige Dimension dabei, wenn ich ein Fachbereich-Mensch bin und schaue immer nur aus diesen fachlichen Aspekten auf die IT, dann stoße ich natürlich an Grenzen mit dem Verständnis für die IT. Und komm dann vielleicht mit Lösungsvorschlägen, die auf den ersten Augenblick, aus IT-Sicht, deppert sich anfühlen. Und diese Möglichkeit dann aber auch zu schaffen, in dieser psychologischen Sicherheit, ja man kann auch einen Vorschlag machen, der vielleicht wirklich abstrus ist. Der vielleicht gar nicht so einem erst mal so in den Sinn kommt, aber daraus ergeben sich sehr, sehr häufig Möglichkeiten, zu sagen: "Ok das kann man nicht so und so umsetzen, aber die Grundidee ist gut, dann könnte ich auch das daraus nehmen." Das heißt, man ist in der Lage, ganz vieles in dieser Lösung zu berücksichtigen, was aus IT-Sicht erst mal gar nicht sichtbar war, aus fachlicher Seite sichtbar wird, in Bezug auf die IT und das kann man umsetzen. Und das kann man aber nur dann umsetzen, wenn ich nach der ersten, sozusagen,

kann es ermöglicht werden, dass unorthodoxe Lösungswege gefunden werden, über die so niemand nachgedacht hat. Dieser Aspekt wird als Förderung von Lösungskompetenz zusammengefasst.

Innerhalb des Mindset wird explizit darauf hingewiesen, dass eine fehlende Fehlertoleranz sehr schädlich sein kann. Das kann dazu führen, dass sich Cititzen Developer aus dem Projekt und der späteren Umsetzung der Plattform zurückziehen, wodurch viel Potential verloren geht.

EB08\_Fey, Peter

überraschenden Lösung, die jemand bringt aus dem Fachbereich, das Lachen gar nicht mehr aufhören will und dann zieht der sich zurück und damit hat sich dann auch die Möglichkeit einer kreativen, ich will fast sagen design thinking-Lösung. Also das Vorstoßen ins Unbekannte gehört da ein bisschen mit rein und das muss in einem Unternehmen dann auch möglich sein. Das ist eine ganz wichtige Sache. Das heißt, die Chefetage muss, sollte in der Lage sein, Lösungskompetenz bei den Mitarbeitern aktiv zu fördern, auf diese Mitarbeiter und Mitarbeiterinnen aktiv auch zuzugehen und die um diese Unterstützung zu bitten, sich Lösungen unterbreiten zu lassen. Das Zweite in diesem Mindset ist, klar, wir haben Freiheiten, die können halt auch dann leicht in Gegenden vorstoßen, in denen wir eigentlich gar nicht aktiv sein wollen. Man muss eine gewisse Governance aufbauen. Man muss sagen, so und so wollen wir unsere Programme haben. So und so wollen wir nach außen auftreten. So und so wollen wir aber auch untereinander, miteinander kommunizieren, so sollen die Oberflächen zum Beispiel gestaltet sein, wenn man das so genau machen möchte. Also die Governance muss einhegen, muss diese Freiheit wieder einhegen und muss den Mitarbeitern auch die Sicherheit geben: Ja, wenn ich dieses in diesem eingehegten Bereich tue, bin ich noch auf dem richtigen Pfad und so können wir dann auch im Nachhinein diese so geschafften Applikationen weiterentwickeln.

EB08\_Fey, Peter: 19 - 19 (0)

2 - EP: Eigenschaften Unternehmen\2.1 - Mindset\2.1.1 - Fehlerkultur

Und Mind Set ist natürlich, also man sollte auf der agileren Seite sein, vom Mindset her gerade auch im Citizen Development, sonst stellt man die Leute vor eine unlösbaren Aufgabe, weil man Leute, die sich weder mit Frontend-Entwicklung, noch Backend-Entwicklung, noch normalen IT-Prozessen auskennen, dann also man darf nicht die Erwartung an die stellen, dass sie dann ähnlich arbeiten können, sondern sollte ein agileres Mindset und eine gute Fehlerkultur haben, dass man den Leuten dann auch

erlaubt, da zu wachsen an den Aufgaben und erst kleinere Sachen umzusetzen und dann eben auch größere Sachen umzusetzen. Der Erfolgsfaktor ist für mich auch, dass man mit Miniprojekten anfängt. Leute, die keine Erfahrung damit haben, mit Miniprojekten anfangen und dann an ihren Aufgaben wachsen, das ist eben auch so vom Mindset her dann kommt, ah ja, Low Code-Plattform, es ist alles einfach und ich will jetzt aber ein Tool, was mir ein komplettes CRM-Management, komplettes CRM mit Projektmanagement und so was alles integriert und dann ist halt der Mensch, der das zum ersten Mal macht, überwältigt einfach mit der Aufgabe.

EB02\_Rosengrün, Robin: 21 - 21 (0)

Die Risiken sind die, dass man zu wenig fehlertolerant ist. Das heißt, wenn ich jetzt hingehe und probiere etwas aus und das funktioniert jetzt nicht sofort, dass man dann sagt: "Mensch, das ist jetzt wieder alles schiefgelaufen und das ist alles Mist." Also diese Einstellungssache, Mindset gehört da auch rein und dass man dann sagt: "Ok, wir haben hier etwas entdeckt, was jetzt noch nicht so ganz gut funktioniert. Macht es noch Sinn, in dieser Richtung uns weiterzugeben? Oder müssen wir was ganz anderes als Lösung versuchen?" Also diese Start-up-Mentalität, die muss dann rein.

EB08\_Fey, Peter: 21 - 21 (0)

2 - EP: Eigen-schaften Unternehmen\2.2- Kapital

Also, ich muss die Lizenzen bezahlen können. Das würde ich sagen, das reicht. Also ich würde echt sagen, ich kann mir durchaus vorstellen, dass so ein Einzelunternehmer wie ich, also ich habe mir auch so kleine Prozesse auch automatisiert und da reicht halt meine Lizenz für aus. Ich würde jetzt sagen, nicht unbedingt, dass das Kapital sein muss, um mit Low Code/ No Code zu arbeiten. Das sehe ich einfach unabhängig davon. Es ist keine Riesen-Anfangsinvestition, die ich machen muss, deswegen kann ich auch mit wenig Kapital beginnen.

EB01\_Eckhard Hauenherm: 33 - 33 (0)

Kapital wird durch den Experten als irrelevant eingestuft. Es ist möglich, als Einzelunternehmer sowie als großer Konzern NCLC-Plattformen einzuführen. Es ist lediglich wichtig, genug Kapital zu besitzen, um die nötigen Lizenzkosten tragen zu können.

EB01\_Eckhard Hauenherm

Also da muss man als Unternehmen natürlich auch Geld in die Hand nehmen,

Der Experte sieht finanzielle Mittel als wichtig an, da bei einer Einführung

	<p>weil diese Rollen müssen gestacked werden. EB03_Fabian Gackstatter: 23 - 23 (0)</p>	<p>entsprechende personelle Ressourcen zur Verfügung gestellt werden müssen. EB03_Fabian Gackstatter</p>
	<p>Also, fangen wir vielleicht mal mit dem Kapital an. Also, prinzipiell sollte von der obersten Unternehmensführung, egal welche Größe, da gesagt werden, das ist eine Investition, dafür stellen wir die finanziellen Mittel bereit. Also das ist eine Basisvoraussetzung, damit da einfach kein Druck herrscht, weil, man kann einfach nicht auch entwickeln, wenn man in einem Sparzwang ist. Das ist einfach auch eine Sache, Projekte sind auch so eine Sache, wo Geld verbrannt wird. Das muss klar sein.  EB05_Rohrbeck, Christian: 31 - 31 (0)</p>	<p>Der Experte sieht es als wichtig an, dass die Einführung einer solchen Plattform als Investition wahrgenommen wird und die entsprechenden finanziellen Ressourcen zur Verfügung gestellt werden. Es wird als hinderlich angesehen, wenn ein Sparzwang herrscht. EB05_Rohrbeck, Christian</p>
	<p>Unternehmensgröße oder Kapital, sehe ich da als nicht so wichtig an. Es gibt Start-ups, die mit Low Code/ No Code arbeiten. Es gibt Konzerne, die mit Low Code/ No Code arbeiten. Und auch das Thema Kapital ist, glaube ich, eher zweitrangig, weil da einfach für viele Preisklassen sich eine Low Code/ No Code-Plattform anbieten kann. Es ist jetzt kein K.-o.-Kriterium, wenn man jetzt gerade zum Beispiel als Start-up oder am Anfang noch, ja, das Budget sehr überschaubar ist, das soll einen nicht davon abhalten, auf diese Plattform zurückzugreifen. EB07_Janke, Tino: 19 - 19 (0)</p>	<p>Der Experte sieht Kapital nicht als wichtige Voraussetzung an. Aufgrund der Vielzahl an möglichen Plattformen mit unterschiedlichen Preiskategorien ist die kein wichtiges Kriterium. EB07_Janke, Tino</p>
	<p>Unternehmensgröße, Kapital was man hat, das ist dann halt doch immer relativ individuell und von Unternehmen zu Unternehmen unterschiedlich EB08_Fey, Peter: 19 - 19 (0)</p>	<p>Der Experte sieht Kapital als individuelles Thema und nicht so wichtig. EB08_Fey, Peter</p>
2 - EP: Eigenschaften Unternehmen\2.3 - Unternehmensgröße	<p>Also ich würde sagen, Größe eigentlich nicht. Ich kann das sowohl in kleinen als auch in großen Unternehmen machen. EB01_Eckhard Hauenherm: 27 - 27 (0)</p>	<p>Der Experte sieht die Unternehmensgröße als irrelevant an. EB01_Eckhard Hauenherm</p>
	<p>Also Unternehmensgröße sehe ich überhaupt gar keine Einschränkungen. Also vom Citizen Development spielt sich dann meistens eh alles entweder innerhalb von Abteilungen ab, die dann ja meistens irgendwie eine Größe vielleicht</p>	<p>Der Experte sieht die Unternehmensgröße nicht als relevant an. Das basiert darauf, dass Citizen Development in den Abteilungen durchgeführt wird und diese meist nicht so groß sind. Auch</p>

bis hundert Leute oder so was, da funktioniert das optimal, aber auch dann darüber hinaus. Aber wir machen jetzt, zum Beispiel auch, wir bilden Prozesse ab, die alle 16000 Mitarbeiter bei uns in den Kerngesellschaften umfassen, bilden wir auch teilweise in der Power Platform ab. Also da sehe ich von bis gar keine Probleme.

EB02\_Rosengrün, Robin: 21 - 21 (0)

bei der Skalierung ins Gesamtunternehmen wird keine Relevanz gesehen.

EB02\_Rosengrün, Robin

Also, ich sage mal, die Manpower an sich spielt, glaube ich, die geringste Rolle an der Stelle. Also es gibt keine spezifische Unternehmensgröße, wo ich sagen würde: "Wenn ihr weniger als X-hundert Leute seid, dann lohnt es sich eigentlich nicht." Weil Citizen Developer, wie gesagt, kann jeder sein. Das kann ich in einem Unternehmen mit zehn Mann anfangen. Das machen wir aber auch bei Konzernen mit fünfhunderttausend Leuten. Also die ganze Spanne ist da.

EB03\_Fabian Gackstatter: 23 - 23 (0)

Der Experte sieht Unternehmensgröße und Mitarbeiteranzahl insgesamt als irrelevant an. Er sieht lediglich den Bedarf dabei, wenn eine NCLC-Plattform eingeführt und umgesetzt wird, sollte diese mit den entsprechenden personellen Ressourcen ausgestattet werden, um einen problemlosen Betrieb zu ermöglichen.

EB03\_Fabian Gackstatter

Die betrifft im Zweifelsfall das ganze Unternehmen und genau so muss ich die auch aufhängen. Dass es eben nicht mit den drei Admins getan ist, die das jetzt noch nebenhermachen, sondern es ist ja ein organisationsweiter Change, den ich da einführe und den muss ich auch mit der entsprechenden Manpower unterfüttern.

EB03\_Fabian Gackstatter: 23 - 23 (0)

Iso ich habe das schon bei Unternehmen oder Banken unterschiedlichster Art gesehen. Von: Bank hat 800 Mitarbeiter oder hat 10000 Mitarbeiter, weil, man kann es für beide einführen, das funktioniert. Ich habe es auch schon gesehen von der Zielgruppe her, dass es Vereine genutzt haben, um ihre Mitgliedsanträge zu verwalten und das effizienter zu gestalten

EB04\_Göttl, Simone: 23 - 23 (0)

Unternehmensgröße wird nicht als relevanter Faktor gesehen.

EB04\_Göttl, Simone

Größe des Unternehmens spielt aus meiner Sicht gar nicht so sehr eine Rolle. Auch große Unternehmen sind in der Lage, sich abteilungstechnisch so klein

Der Experte sieht das Thema Unternehmensgröße als nicht relevant an. Auch in großen Unternehmen kann die Plattform auf Abteilungsebene mit wenigen Mitarbeitern funktionieren.



	<p>herunterzumachen, dass man nur noch den Abteilungsfokus hat und das drum rum verschwindet. Es kommt rein auf den Willen für die Innovation an.</p>	EB05_Rohrbeck, Christian
	EB05_Rohrbeck, Christian: 31 - 31 (0)	
	<p>Also nein, Firmengröße spielt für unser Produkt keine Rolle. Im Gegenteil, das hilft den großen Konzernen hier, ja, die große Rollen spielen, wie auch die ganz Kleinen, mit wenig Mitarbeitern und kleinen Kundenkontakten.</p>	<p>Firmengröße durch den Experten als irrelevant angesehen. EB06_Hesse, Moritz</p>
	EB06_Hesse, Moritz: 33 - 33 (0)	
	<p>Unternehmensgröße oder Kapital, sehe ich da als nicht so wichtig an. Es gibt Start-ups, die mit Low Code/ No Code arbeiten. Es gibt Konzerne, die mit Low Code/ No Code arbeiten. Und auch das Thema Kapital ist, glaube ich, eher zweitrangig, weil da einfach für viele Preisklassen sich eine Low Code/ No Code-Plattform anbieten kann. Es ist jetzt kein K.-o.-Kriterium, wenn man jetzt gerade zum Beispiel als Start-up oder am Anfang noch, ja, das Budget sehr überschaubar ist, das soll einen nicht davon abhalten, auf diese Plattform zurückzugreifen.</p>	<p>Unternehmensgröße durch den Experten als irrelevant eingestuft. EB07_Janke, Tino</p>
	EB07_Janke, Tino: 19 - 19 (0)	
	<p>Unternehmensgröße, Kapital was man hat, das ist dann halt doch immer relativ individuell und von Unternehmen zu Unternehmen unterschiedlich</p>	<p>Der Experte sieht Unternehmensgröße als individuelles Thema und nicht so wichtig. EB08_Fey, Peter</p>
	EB08_Fey, Peter: 19 - 19 (0)	
<p>2 - EP: Eigenschaften Unternehmen\2.4 - Technische Planung</p>	<p>Was natürlich sehr stark eine Rolle spielt, wenn man so ein bisschen darüber nachdenkt, die meisten Unternehmen, die ich kenne, die so was einführen, sind schon sehr stark in der Cloud vertreten. Also bilden viele Funktionalitäten im Cloudgeschäft ab oder haben halt Cloudplattformen im Einsatz. Also bei der reinen ON Prem Variante kommt No Code/ Low Code deutlich seltener vor.</p>	<p>Als technische Voraussetzung wird eine Art Cloud Dienst gesehen. On Prem kommt aus der Erfahrung selten vor. EB01_Eckhard Hauenherm</p>
	EB01_Eckhard Hauenherm: 27 - 27 (0)	
	<p>Und dann muss ich natürlich auch die technische Grundlage sauber definiert haben. Also da kommen dann die Nicht-Entwickler aus meiner zentralen IT wieder zum Tragen, sprich die Power</p>	<p>Der Experte hebt hervor, dass eine technische Planung als sinnvoll betrachtet wird. Dafür sollten die Nicht-Entwickler aus der IT-Abteilung mit einbezogen werden. Diese müssen die</p>

Platform-Administratoren, ein Active Directory-Administrator, der jetzt auch die richtigen Rechte tatsächlich freischalten kann und der auch das ganze Thema monitoring dann überwacht. Ich will ja als Unternehmen wissen, was machen denn meine Citizen Developer überhaupt im System. Bauen die Applikationen? Bauen die Automationen? Nutzen die tatsächlich auch nur die Tools, die eigentlich dafür vorgesehen sind oder haben die aus Versehen irgendwelche Sachen offengelassen? Dass ich zum einen das monitoren kann, zum anderen aber auch darauf reagieren kann. Was passiert denn, wenn der Rocco jetzt eine Applikation gebaut hat und teilt die mit hunderttausend seiner Kollegen. Ist das so vorgesehen? Wollen wir das überhaupt? Und wie unterstützen wir Rocco dabei, wenn seine Applikation jetzt wirklich gut ist? Wie verknüpfe ich Rocco mit unserem zentralen Support? Weil, wenn ich jetzt eine Applikation baue und teile die mit meinem Team, mit fünf anderen Leuten, die kriege ich selber nach wie vor unterstützt. Wenn ich jetzt eine Applikation baue, die morgen von hunderttausend Leuten bedient wird, dann werde ich wahrscheinlich unter der Last der Issues und feature Request und Anfragen zusammenbrechen und kann meinen eigenen Tagesjob eigentlich nicht mehr machen. Also, über genau solche Themen muss ich mir Gedanken machen. Was sind denn die Citizen Developer-Applikationen? Wie, in welchem Rahmen sollen die sich bewegen? Und wann wird aus einer Citizen Developer-Applikation tatsächlich eine businesskritische, die im ganzen Unternehmen eingesetzt wird? Und das sind dann alles solche Gedanken, die mache ich mir idealer Weise im Vorfeld. Ich muss leider dazu sagen, die Hälfte unserer Kunden fängt erstmal damit an, dass sie sagt: "Wir haben jetzt die Lizenzen gekauft. Wir bauen jetzt mal eine Applikation." Und nach einem halben Jahr ist dann: entweder brennt die Hütte, weil die interne IT dann keinerlei Kontrolle mehr darüber hat, die weiß nicht, wo sie gucken muss. Sie weiß nicht, wie sie darauf reagieren

entsprechenden Governance-Richtlinien, Monitoring und technischen Vorgaben umsetzen. Diesen obliegt auch die Überwachung den Application Life Cycles und der Erfassung von Messdaten zur Kontrolle über den Erfolg der Plattform. Ein weiterer Aspekt, der in diesem Zusammenhang zu berücksichtigen ist, ist, dass Applikationen, die eventuell unternehmensweit ausgerollt werden, ebenfalls in die Support- und Monitoring-Struktur integriert und im weiteren Verlauf überwacht werden müssen. Es sollte vermieden werden, dass die IT-Abteilung unter der Last von Anfragen so weit eingeschränkt wird, dass diese ihr originäres Aufgabenfeld nicht mehr erfüllt. Es wird als wünschenswert hervorgehoben, dass diese Gedanken bereits vor dem Beginn des Projektes erfolgen, um nicht in die Problemstellungen zu kommen, dass entweder die IT-Abteilung überfordert ist oder die Plattform abgelehnt wird durch die Mitarbeiter.

Als wichtige Fragen, die in einer technischen Richtlinie besprochen werden sollten, werden folgende aufgeführt: Welche Szenarien habe ich denn? Was sind klassische Entwickler-Applikationen? Was sind klassische Citizen Developer? Wie sieht die Zone dazwischen aus? Wie sieht auch eine Progression aus von einer Applikation, die ursprünglich mal ein User für sich gebaut hat? Die dann aber so gut war, dass sie irgendwann zur Enterprise-Applikation geworden ist. Wie sieht dieser Lebenszyklus aus?

EB03\_Fabian Gackstatter

soll und stellt im Zweifelsfall die Plattform wieder ein. Oder es passiert, dass eben die User nicht abgeholt werden. Es wird kein Training zur Verfügung gestellt. Es gibt keinen Feedback-Mechanismus, es gibt keine Trainings und die Plattform hebt eben nicht ab und es werden keine Applikationen gebaut und die Lizenzen gammeln einfach vor sich hin.

EB03\_Fabian Gackstatter: 21 - 21 (0)

Auf der anderen Seite, mit kontrollierter Offensive, kriege ich genau diese Richtlinien platziert, so dass ich definieren kann: Welche Szenarien habe ich denn? Was sind klassische Entwickler-Applikationen? Was sind klassische Citizen Developer? Wie sieht die Zone dazwischen aus? Wie sieht auch eine Progression aus von einer Applikation, die ursprünglich mal ein User für sich gebaut hat? Die dann aber so gut war, dass sie irgendwann zur Enterprise-Applikation geworden ist. Wie sieht dieser Lebenszyklus aus? Und wenn ich die Sachen alle sauber definiert habe, dann funktioniert so eine Plattform auch.

EB03\_Fabian Gackstatter: 21 - 21 (0)

Ja, also ich sage jetzt mal, tatsächlich harte Voraussetzungen sind eigentlich nur technische Systemvoraussetzung um die Plattform, also den Designer zu bedienen, diese App zu bauen und dann die entsprechenden Betriebssysteme und Endgeräte, um die Plattform zu betreiben. Ich würde sagen, ansonsten gibt es A priori keine Voraussetzungen.

EB06\_Hesse, Moritz: 31 - 31 (0)

Als einzige wichtige Voraussetzung, die erfüllt sein muss, sieht der Experte nur technische Voraussetzungen.

EB06\_Hesse, Moritz

2 - EP: Eigenschaftlichen Unternehmen\2.5 - Erfahrenes Projektmitglied

Na gut, das kommt darauf an. Also wenn eine Unternehmung schon mehrere agile Projekte gefahren hat und weiß, wie es geht, was meine ich damit, die Mitarbeiter haben schon Erfahrung damit, dann sehe ich da keine Hürden.

EB04\_Göttl, Simone: 39 - 39 (0)

Der Experte stellt heraus das er es als positiv ansieht, wenn bereits Projektmitglieder im Einführungsteam sind, welches bereits Erfahrung mit Plattform und Projektvorgehen hat.

EB04\_Göttl, Simone

Ich glaube, es ist auf jeden Fall gut, wenn man jemanden, der einen solches No Code/ Low Code-System kennt, mit zu den Kollegen setzt, die das am Ende realisieren sollen.

Der Experte sieht einen erfahrenen Kollegen im Projektteam als positiv an.

EB05\_Rohrbeck, Christian

EB05\_Rohrbeck, Christian: 31 - 31 (0)

3 - EP: Risiken und Hürden vor der Einführung\3.1 - Schatten IT

Ja, wie schon eben erwähnt, die Governance. Das ist ein ganz wichtiger Punkt. Das hängt ein bisschen davon ab, mit welcher Plattform ich arbeite, aber die meisten Plattformen sind am Anfang immer relativ offen und bieten relativ viele Möglichkeiten und wenn ich bei der Einführung nicht darauf achte und erst mal mit offenem Scheunentor alle Anwender sozusagen da rauf lasse und die Daten von links und rechts verschieben lasse, auswerten lasse, irgendwelche Formulare dem Kunden zur Verfügung stelle, ohne an eine Datenschutzerklärung zu denken und solche Sachen. Da sehe ich tatsächlich eine Hürde, das kann gefährlich werden für ein Unternehmen. Also das ist jetzt keine Einführungshürde, aber es ist ein hohes Risiko, an das ein Unternehmen denken sollte.

Der Experte vertritt die Auffassung, dass ein unkontrolliertes Arbeiten mit der Plattform nachteilig ist für den Einsatz der NCLC-Plattform. Er umschreibt dabei eine Schatten- IT.

EB01\_Eckhard Hauenherm

EB01\_Eckhard Hauenherm: 35 - 35 (0)

Also, dass diese Schatten-IT 2.0 eben nicht passiert, muss man ein gewisses Level an Governance aufbauen. Das kann ja auch ein bisschen wachsen mit der Low Code- Plattform. Aber man sollte sich von Anfang an bewusst sein, dass man so was braucht und dass jemand dafür abgestellt ist, der sich damit beschäftigt von: Kritische Konnektoren blocken oder überwachen, Neuerungen in der der Plattform dann entsprechend den Policies umsetzen/ einbinden, also da braucht man einfach jemanden, sollte man jemanden haben, der das macht. Ist das Erste, und das Zweite ist, woran es immer scheitert, es muss ein Bewusstsein für Daten geben, finde ich. Also es muss Leute geben, die Datenstrukturen in Datenbanken denken können, sonst funktioniert es nicht, weil, damit kann ich dann eben auch den Vorteil realisieren, dass ich halt Sachen nicht in einem Excel Spreadsheet abbilde, sondern in einer relationalen Datenbank.

Der Experte stellt heraus, dass in Folge einer Governance Strategie es nötig ist, eine personelle Ressource bereitzustellen, um eine Schatten-IT zu verhindern. Die Aufgabe dieser Ressource ist es, Neuerungen der Plattform den entsprechenden Richtlinien anzupassen, kritische Konnektoren zu überwachen und zu verwalten. Ebenso sollte bei den Beteiligten ein Bewusstsein für Daten, Datenstrukturen und Datenbanken aufgebaut werden. Dies beugt dem Entstehen der Schatten-IT vor.

EB02\_Rosengrün, Robin

EB02\_Rosengrün, Robin: 19 - 19 (0)

Die hatte ich schon bei den Nachteilen angesprochen, und die würde ich an der Stelle auch tatsächlich noch mal mit

Der Experte stellt zur Vermeidung von Wildwuchs die Notwendigkeit einer guten Dokumentation dar. Die

reinbringen. Das eine ist dieser Wildwuchs. Einfach, weil man ja eben nicht an diese klassischen Verfahren gebunden ist. Also egal, ob das jetzt ein Wasserfallmodell ist oder ob das, sag ich jetzt, ein agiler Software-Entwicklungsprozess nach Scrum ist, wo man ja doch eine gute Dokumentation und eine Übersicht trotzdem behält über das, was entwickelt wird und man kann natürlich auch beide Verfahren auf Low Code-Lösungen anwenden. Also man kann sagen, das, was wir jetzt bauen, wollen wir nach Wasserfall bauen oder das, was wir jetzt bauen, wollen wir trotzdem nach Scrum oder nach Kanban oder nach agiler Software-Entwicklungsmethodik bauen. Ich würde aber mal behaupten, das passiert meistens nicht, weil das ja diese Leichtigkeit und Schnelligkeit der Lösung mitbringt, dass man auf diese ganzen Verfahren vielleicht geneigt ist, verzichten zu wollen. Das ist natürlich ein Risiko, weil, dann hat man irgendwann einen Wildwuchs. Man kann vielleicht nicht mehr gut nachvollziehen, wann ist also dieser Change, Change-Management-Prozess, ist vielleicht nicht so gut nachvollziehbar, das muss nicht einhergehen mit einer Low Code-Lösung. Wie gesagt, man kann sich ja trotzdem Governance mäßig quasi vorschreiben, dass bestimmte Prozesse auch entsprechend dokumentiert werden müssen oder so was, aber das sind natürlich, ich sage mal, Schmerzen, die bei solchen Lösungen wahrscheinlich eher auftreten, als wenn das ein strukturierter Software-Entwicklungsprozess ist. Und welche Probleme und Risiken treten auf? Insbesondere würde ich dann halt sagen, das, was ich auch schon vorher berücksichtigt hatte. Die Lösung wird geschafft, die funktioniert schön, aber jetzt geht es darum, diese Insellösung irgendwie wieder in die Hauptarchitektur zu integrieren. Und spätestens dann wird jede Low Code-Lösung, ich kriege das mit als technischer Integrator, als technischer Berater hier. Es ist ja technisch gesehen keine Low Code-Lösung, die darunter liegt, weil, das müssen ja Leute programmieren und wenn es angebunden wird, da fallen ganz normale JSON-Objekte

Problemstellung ist, dass bei mangelhafter Dokumentation, unabhängig vom Vorgehensmodell, im weiteren Verlauf es zu Problemen der Nachvollziehbarkeit kommt, spätestens wenn einige Änderungen im Projekt erfolgt sind.

EB06\_Hesse, Moritz

oder XML-Objekte oder so was raus. Oder irgendwelche Bytestreams, und die müssen ja entsprechend verarbeitet werden und deshalb spätestens dann verlässt das ja den gesamten Low Code-Bereich und insbesondere Firmen, die dann Low Code-Lösungen tief integrieren wollen in ihre Unternehmensarchitektur, die sind dann doch angewiesen auf entsprechende IT-Kompetenzen und Kapazitäten, weil die Integration dann doch häufig halt noch eine entsprechende Expertise erfordert.

EB06\_Hesse, Moritz: 35 - 35 (0)

Governance ist noch ein ganz großer Punkt, das hatte ich vorhin schon mal kurz angesprochen, dass man versucht, durch interne Leitplanken diesem Wildwuchs vorzubeugen, dass einfach klar ist, wie ist die Aufgabenteilung. Wer darf was machen, welche Fachabteilung darf was bauen, wo sind aber gegebenenfalls Aufgaben, die bei der IT-Abteilung liegen bleiben und es muss natürlich auch klar sein, dass es ein Thema ist, dem man sich immer wieder widmen muss. Also man hat nicht einmal Low Code/ No Code-Plattform eingeführt und ist dann damit fertig, sondern es ist natürlich ein Thema, was kontinuierlich verbessert werden muss.

EB07\_Janke, Tino: 21 - 21 (0)

Eine Hürde wird beim Experten gesehen, wenn keine Governance-Strategie erstellt wird oder wurde. Ohne Governance-Strategie besteht die Gefahr, dass der sogenannte Wildwuchs entsteht. Es muss klar sein, wie die Aufgabenteilung ist und welche Abteilung, welche Nutzungsrechte besitzt. Im Weiteren sollte geklärt sein, was schlussendlich weiter durch die IT-Abteilung realisiert werden soll. Es wird auch darauf hingewiesen, dass dieses Thema nicht mit dem Ende des Projekts fertiggestellt ist, sondern eine fortlaufende Aufgabe ist.

EB07\_Janke, Tino

3 - EP: Risiken und Hürden vor der Einführung\3.2 - Cybersecurity und Datenschutz

Also häufiges Thema ist natürlich, was darf ich in dieser Plattform? Das heißt eigentlich auch, welche Funktionen schalte ich frei an dem Punkt und wie löse ich diesen Sicherheitsaspekt. Was meine ich, zum Beispiel, damit. Also bei Teams, wo wir jetzt drüber reden, da gibt es ja zum Beispiel so eine Funktion, die heißt Transkript, und wenn man die anstellt, dann können Sie meinen Wortlaut, wie so ein bisschen im Fernsehen, mitlesen. Das ist ein total cooles Feature, ist aber aus Datenschutzgesichtspunkten total fragwürdig, weil die Daten, zumindest ist so mein letzter Stand, dort häufig halt immer nach Amerika gehen. Das heißt, das wäre aus EU-Sicht für einen Datenschutzkunden gar nicht zulässig. Das heißt, es gibt aber auch in diesen No

Die Expertin weist auf die Relevanz einer Sicherheitsleitlinie hin. Dies betrifft auch das Thema Datenschutz und sollte im Vorfeld geklärt werden. Was passiert mit den Daten? Was darf mit den Daten passieren? Was sind Regularien, nach denen sich das Unternehmen in Bezug auf Datenschutz richten muss? Welche Funktionen können freigeschaltet werden, ohne die unternehmensinterne IT-Sicherheit zu gefährden?

Weitere Überlegungen sind: Lässt ein Unternehmen zu, dass der Hersteller der Plattform Zugriff auf die eigenen Daten erhält? Welche Folgen hat es, wenn kein Zugriff gewährt wird? Welche Daten dürfen durch die Plattform generell verarbeitet werden? Wie

Code/ Low Code- Plattformen Funktionalitäten, die vielleicht, aus welchen Gründen auch immer, nicht in Europa gespeichert werden. Das muss man sich anschauen. Plus, was ein zweiter Punkt ist, man muss natürlich auch die ganzen Security-Aspekte mit betrachten. Hat jetzt ein Hersteller Zugriff auf meine Daten, weil ich im Vertrag stehen habe, dass ich einen Wartungsvertrag habe, dass er Zugriff haben kann und dann ist natürlich die Frage, was sind das für Daten, die ich hier in dieser No Code/ Low Code-Plattform speichere. Sind das öffentlich verfügbare Informationen? Also, ich sage mal so, Informationen, die normalerweise auch auf einer Website wären oder in der Presse, ist das kein Problem. Ist es aber etwas, was gewissen Vertraulichkeitsgesichtspunkten unterliegt, habe ich natürlich schon eine Herausforderung zu schauen, wie schütze ich die und was wende ich für Security-Maßnahmen an, damit die nicht einfach abfließen können, und sei es nur der Hersteller, der sich das anschaut. Und das ist schon ein wichtiger Punkt, wo gerade Banken und Versicherungen natürlich auch viel mit draufschauen. Also für welche Daten gebe ich das überhaupt frei und auch die Informationssicherheit, die da ein großes Auge darauf hat. Und ich habe dann halt die Möglichkeit, das zu lösen mit technischen Maßnahmen, dass ich mehr verschlüssele, vielleicht auch gar nicht ablege und Sachen auch nicht zulasse. Oder ich habe natürlich auch immer die Möglichkeit, dass ich den Mitarbeitern in Form von Arbeitsanweisungen und Hilfestellungen gewisse Vorgaben auferlege. Aber das sind halt Punkte, Menschen sind da natürlich nicht so sicher wie die Technik vielleicht an der einen oder anderen Stelle. Und das sind definitiv gerade, die ich sehe, eine der großen Hürden und wo man sich natürlich dann auch einfach, wenn man in einem regulierten Umfeld unterwegs ist, aber auch andere Konzerne, natürlich relativ schwer damit tut und ein bisschen vorsichtiger ist.

EB04\_Göttl, Simone: 27 - 27 (0)

schütze ich meine Daten? Wie sichere ich die Plattform ab, in Übereinstimmung mit meinen Sicherheitsrichtlinien? Diese Fragestellungen werden als Hürde betrachtet und sollten bedacht werden, vor oder spätestens während der Einführung.

EB04\_Göttl, Simone

wenn ich jetzt einen Fachbereich ermächtigt, große Workflows für die Abläufe auf IT-Basis für Unternehmen zu erstellen, dann muss ich halt, unabhängig davon, ob die Leute Code schreiben wollen oder nicht, muss ich die ganzen Umgebungsbedingungen, die muss ich halt auch bedenken. Das heißt, es geht um Datensicherheit bei der Richtung, dass einerseits die Sicherheit der Daten, aber andererseits die Verweigerung von unberechtigten Zugriffen auf die Daten, also das ist die Datenkonsistenz zum Beispiel, das gehört auch noch zur Datensicherheit.

EB08\_Fey, Peter: 9 - 9 (0)

Dann die nächste Phase der Umsetzung, die ist jetzt aber nicht viel später, sondern die passiert relativ zeitnah zu dem, was man da erarbeitet hat. Man führt diese preparation-Phasen durch, stellt die richtigen Fragen, überlegt sich die richtigen Lösungen und dann geht man mit dieser Lösungsidee auf diejenigen zu, der sich mit diesem Tool, ich sagte es ja gerade, das muss man dann schon ein bisschen üben. Das ist ähnlich wie bei Power User von Excel, ergibt sich das nicht von alleine. Das heißt, derjenige, der setzt das dann um und zeigt es sofort am Bildschirm und dann kann das erste Feedback schon stattfinden: "Ah, das können wir anders noch machen!" So, und dann geht das von diesem Moment an weiter in diesen iterativen Schleifen, bis man dann sagt: "Ok, jetzt haben wir eigentlich diese Lösung, die wir ganz zum Anfang mal angedacht haben. Die haben wir jetzt in der und der Form realisiert." Und dann geht es in das Testing rein, also, das muss ja, so gerade, wenn man nach außen geht, zum Kunden, muss das ja absolut sicher sein, in das Testing rein und dann muss man natürlich, bevor man das überhaupt auf die Welt loslässt, also "Hello World", wie wir alle früher mal gelernt haben, da muss man dann natürlich auch die ITler da und die IT-Governance befragen. Kommen wir mit diesen Programmierungen, die wir da getan haben, kommen wir damit um die Ecke oder stoßen wir da an

Der Experte hebt die Wichtigkeit einer IT-Security-Strategie hervor. Dieses Thema ist eng verbunden mit einer IT-Governance-Strategie. Es müssen Rahmenbedingungen für die Anwendungen geschaffen werden, welche einen Schutz von Daten und die IT-Sicherheit gewährleisten.

Dafür sollte ein Prozess erstellt werden, wie die Einführung von Applikationen durchgeführt wird. Der Vorschlag des Experten teilt sich in verschiedene Phasen auf.

Preparation Phase:

Hier werden Fragen gestellt, welche Lösung sinnvoll ist und wie diese umgesetzt werden soll. Welche Beschränkungen sollen implementiert sein?

Umsetzung:

Es wird ein Prototyp entwickelt und vorgestellt. Hier wird auch ein erstes Feedback eingeholt.

Iterative Wiederholungen:

Der Schritt der Umsetzung wird so lange wiederholt, bis eine optimale Lösung gefunden ist.

Testing:

Im Testing muss die IT-Governance und die IT-Sicherheit das Tool ansehen und bestimmen, ob es den Richtlinien entspricht. Für den Experten ist dies besonders wichtig, da ein normaler Fachbereichsmitarbeiter nicht weiß, welche Sicherheitsrisiken es gibt.

EB08\_Fey, Peter



Sicherheitsgrenzen. Was müssen wir zusätzliche bedenken? Also das fängt schon so eine Low Code/ No code-Applikation ganz gut ab, aber die, wie gesagt, das ist eine allgemeine Vorgabe, eine allgemeine Funktionalität und die Prüfungen, die müssen tatsächlich dann auch durch die IT-Security erfolgen, also Spezialgebiet, und, wie wir jetzt durch die vielen aktuellen Dinge gelernt haben, wird das immer wichtiger, dass man diese IT-Sicherheit betrachtet. Ich habe gestern noch einen Bericht gesehen über diese Fake-Interviews, die die russischen Komödianten da gemacht haben. Also, es ist enorm, was da an Möglichkeiten mittlerweile geschaffen worden ist und guter Programmierer kommt vielleicht drauf. Ein guter Fachbereichsmensch, der kommt nicht drauf, der ahnt nicht, wie viele böse Dinge es in dieser Welt gibt und die sich alle auf sein Programm konzentrieren. Und da muss dann halt, sagen wir mal, ein Fachmann draufgucken und dann sagt der: geht oder geht nicht.

EB08\_Fey, Peter: 31 - 31 (0)

3 - EP: Risiken und Hürden vor der Einführung\3.3 - Thema unterschätzt

Falsche Erwartungen ist ein Problem, ein Risiko, das einfach nicht, also es gibt ja klassischerweise den Anforderer dann an Problemen, ist jetzt vielleicht auch der Chef irgendwie. Der hat dann von der Low Code-Plattform gehört und verdonnert seinen Mitarbeiter, damit was zu machen und hat einfach falsche Erwartungen, was man damit machen kann und was auch ein Mitarbeiter, der neu in so einer Rolle ist oder so etwas zum ersten Mal macht, was er damit bewirken kann.

EB02\_Rosengrün, Robin: 25 - 25 (0)

Also, da habe ich schon mit vielen Leuten darüber gesprochen, weil also unser Ansatz war. Ja, vielleicht, kannst du Transkript kommt in den Anhang? Die Einführung ist, wir waren relativ früh mit der Microsoft 365-Einführung, da ist die Power-Plattform dabei und wir haben uns im ersten Schritt zu wenig Gedanken darüber gemacht, da war ich auch noch nicht in der Struktur und haben da eher

Der Experte warnt davor, wenn sich nicht im Vorfeld ausreichend über das Thema informiert wird und durch die Unternehmensführung falsche Erwartungen erzeugt werden, was die Plattform kann. Dies kann dazu führen, dass die Mitarbeiter nicht bereit sind, die Plattform zu nutzen, da sie die Erwartungen nicht erfüllen können. Es wird auch darauf hingewiesen, dass unter dem Aspekt der falschen Erwartungshaltung der Umfang des Projekts unterschätzt wird. Wird das Projekt nicht richtig durchgeführt, kann es zu erheblichen Verwerfungen kommen und dies kann zu Verzögerungen in der Einführung oder einem erhöhten Ressourcenbedarf führen.

EB02\_Rosengrün, Robin

noch einen Bereich, der eigentlich nichts mit der Power Platform zu tun hatte, wurde das mit aufs Auge gedrückt. Die haben das gut gemacht, aber da kam keine geordnete Einführung. Leute wurden nicht aufgeklärt, was sie damit machen können und dürfen. Und es ging bei uns eher so ein bisschen, naja, es war dann auf einmal da und die Leute konnten es nutzen.

EB02\_Rosengrün, Robin: 35 - 35 (0)

Also, die zwei größten Risiken, die ich eigentlich sehe bei der Einführung für uns, denen wir uns normalerweise gegenübersehen, ist zum einen mangelndes Personal, weil gerade solche Center of Excellence-Implementierungen, die wir am Anfang machen, da ist eben wichtig, dass der Kunde da mitarbeitet. Das ist viel Knowledge-Transfer von unserer Seite, das heißt, da müssen dann aber auch die entsprechenden Kundenressourcen verfügbar sein, präsent sein und mitarbeiten, damit wirklich da was Brauchbares bei rauskommt.

EB03\_Fabian Gackstatter: 25 - 25 (0)

Und es ist eben auch wichtig, dass die Kunden verstehen, dass wir ihnen nicht vorkauen können: "So müsst ihr eine No Code/ Low Code-Plattform aufsetzen." Wir haben zwar entsprechende Best Practices, aber jedes Unternehmen tickt eben anders und unser Job ist es dann, die Plattform so anzupassen und den Kunden durch den richtigen Entscheidungsprozess durchzuführen, dass am Ende die Plattform dabei rauskommt. Also so als Beispiel, wir haben als Kunden auf der einen Seite eine SAP, wo quasi 90 % der Mitarbeiter Entwickler sind und ich habe auf der anderen Seite Kunden, das sind Ministerien, die eine Plattform einführen, wo ein Bruchteil der Leute Entwickler sind. Das heißt, da jetzt zu vermuten, ich platziere genau ein und dieselbe Lösung mit denselben Policies, mit denselben Richtlinien und die funktioniert bei beiden Unternehmen, ist unrealistisch und das müssen die Kunden halt auch verinnerlichen, dass wir ihnen dabei helfen, die Plattform und das ganze

Der Experte berichtet davon, dass eine der größten Hürden bei der Einführung ist, dass das Thema unterschätzt wird. Er berichtet, dass die nötigen personellen Ressourcen nicht zur richtigen Zeit zu Verfügung stehen. Vor allem zu Beginn ist dies problematisch, da dort ein hoher initialer Aufwand zur Befähigung für das Projekt erfolgt. Ein weiterer Punkt, auf den der Experte eingeht, ist, dass die Unternehmen ihre eigenen Anforderungen nicht definiert haben. Dies ist sollte möglichst im Vorfeld erfolgen und die Plattform sollte soweit wie möglich auf die Anforderungen des Unternehmens angepasst werden und nicht das Unternehmen sich an die Plattform anpassen. Dies ist natürlich nicht immer zu 100 % möglich, sollte aber als Maßstab angesehen werden. Hier wird auf Diversität von Fähigkeitsleveln von Mitarbeitern eingegangen, weshalb die es wichtig ist, dass Kunden im Vorfeld bereits die für sie nötigen Aspekte einer Einführung und Befähigung definiert haben.

Der Experte hebt hervor, dass eine weitere Problemstellung bei der Einführung ist, dass Unternehmen den Gesamtumfang des Projektes unterschätzen. Er weist darauf hin, dass Unternehmen durch eine solche Plattform grundlegende Veränderungen erfahren und dies nicht auf die IT-Abteilung als Ansprechpartner reduziert werden kann.

EB03\_Fabian Gackstatter

Tooling drum herum auf ihre Bedürfnisse zuzuschneiden, damit sie ihre Prozesse unterstützen, damit sie ihre Personalstruktur auch unterstützen. Und das kann ich Ihnen halt nicht vorgeben. Also, das ist das Problem, dass die Kunden das personell meistens unterschätzen.

EB03\_Fabian Gackstatter: 25 - 25 (0)

Das ist halt sehr wichtig, dass der Kunde wirklich weiß, worauf er sich denn da einlässt. Wie gesagt, eine No Code/ Low Code-Plattform ist eben nicht nur: Ich führe jetzt ein neues Tool ein ins System, sondern das ist schon eine, das geht schon ziemlich an die Basis von einem Unternehmen und das unterschätzen eben viele Kunden. Und wenn sie sich dann darauf Committed, dass sie wirklich sich darauf fokussieren, wie kriege ich denn jetzt die Technik zum Laufen. Weil es in der Regel von der IT mal vorangetrieben wird, so ein Thema, und dann halt auch so mit Scheuklappen draufguckt wird: Ich muss nur die Technik ans Laufen kriegen. Der Rest passiert dann schon irgendwie. Das ist halt gefährlich, so eine Geschichte. Also da muss ich auch meine Kunden immer wieder drauf hinweisen: Ihr dürft es nicht nur auf die Technik reduzieren. Die Technik macht maximal 25 % aus von der ganzen Geschichte. Der Rest ist, wie eure Unternehmensstruktur dieses Thema absorbieren kann und tatsächlich umsetzen kann. Wenn ihr nur die Technik zum Laufen kriegt, werdet ihr an den anderen Themen scheitern, wenn ihr das so unterschätzt. Das würde ich so mitgeben. Das ist einfach wichtig, so als Best Practice und da ruhig auch drauf rumreiten. So als Mantra: Jungs, es ist nicht nur die Technik! Power Platform betrifft alles. Power Platform rennt Silos ein, weil, ich kann auch nicht mehr alleine für mich alles entscheiden, wenn es zum Beispiel um Thema Policies für Connectoren geht. Mit welchen Datenstrukturen darf ich denn kommunizieren? Betrifft alles, was auf der Plattform stattfindet und eben nicht nur das Team das jetzt Connectoren baut, zum Beispiel und solche Sachen.

EB03\_Fabian Gackstatter: 71 - 71 (0)

das sollte man in Konzernen nie vergessen, das ist der Betriebsrat oder die Personalversammlung. Das heißt, all das beeinflusst ja Mitarbeiter und da hat der Betriebsrat auch ein Mitsprachrecht und auch solche Tools müssen in so eine Konzern- oder einer Betriebsvereinbarung aufgenommen werden. Das heißt, diese Personen oder Personengruppen mit zu involvieren von Tag eins, ist unheimlich wichtig, weil, am Ende können das die Show Stopper sein, dass ein Tool nicht ausgerollt wird. Habe ich auch schon erlebt. Es war bei einer MS Teams- Einführung bei einem Kunden hat sich der Rollout zwei Wochen verzögert, weil es keine Freigabe des Betriebsrats gab. Und das ist, glaube ich, etwas, was außerhalb dessen ist, was man vielleicht im Framework explizit sieht. Es ist vielleicht ein Stakeholder, aber ein Stakeholder, der halt nicht als Stakeholder bezeichnet werden sollte, sondern eine sehr wichtige Rolle einnimmt.

EB04\_Göttl, Simone: 45 - 45 (0)

Probleme, die ich sehe, sind zum einen, dass das Thema unterschätzt wird, dass es als IT-Thema betrachtet wird, was es aber eigentlich nicht ist, sondern dieses Thema muss halt höher aufgehoben werden als, in Führungsstrichen, nur in der IT. Möchte damit nicht die IT-Abteilung abwerten. Aber es ist halt ein Thema, was die ganze Organisation betrifft, weil man ja gerade Verantwortlichkeiten aus der IT herausholen möchte und in andere Abteilungen hineinragen möchte. Und damit kann ich beispielsweise nicht nur den Head of IT damit betrauen, sich dieses Projekt mal anzuschauen, sondern es muss einfach der Sponsor, der muss weiter oben in der Organisation sitzen. Zum anderen muss es in der Organisation einfach ganzheitlich betrachtet werden, das heißt: Es wäre schade für das Projekt, wenn man sagt, ich schaue mir Abteilungen A und B an und die Abteilungen C bis J lass ich bei dem Thema raus, weil ich das bei

Ein Aspekt, den die Expertin herausstellt, ist, darauf zu achten, wirklich alle relevanten Stakeholder von Anfang an mit einzubinden. Sie führt als Beispiel einen Erfahrungsbericht an, in dem der Betriebsrat nicht voll eingebunden wurde und aus diesem Grund eine Verzögerung im Einführungsprojekt erfolgt ist.

EB04\_Göttl, Simone

Der Experte sieht es als Problemstellung an, wenn der Projektumfang unterschätzt wird, wenn es in der Unternehmenshierarchie zu niedrig eingeordnet wird und nur als reines IT-Thema gesehen wird. Die Einführung einer solchen Plattform betrifft oftmals das ganze Unternehmen und sollte mit der entsprechenden Priorität behandelt werden. Das Projekt sollte organisationsweit betrachtet werden und nicht ausschließlich reduziert auf einige Abteilungen, selbst wenn es zu Beginn nicht für das ganze Unternehmen geplant wird, kann eine spätere Skalierung noch erfolgen. Auf einen wichtigen Fehler weist der Experte hin, dass nach Ende des Einführungsprojektes nicht alle Projektmitarbeiter entfernt werden, sondern ein gewisser Anteil für den Weiterbetrieb und die Weiterentwicklung im Projekt belassen werden sollte.

In diesem Zusammenhang wird auf ein Commitment der relevanten Teilnehmer

denen nicht sehe. Sondern ich muss im Prinzip das Thema so weit wie möglich versuchen, in der Organisation zu verankern.

EB07\_Janke, Tino: 21 - 21 (0)

Siehst du da hauptsächlich, also ein strukturelles Vorgehen, wo die Probleme auftauchen können, oder habe ich dich missverstanden?

**B: TJ** Genau. Also zum einen ist es natürlich so, dass das Projekt in seiner Tragweite unterschätzt werden kann. Das kommt ja sehr IT-nahe daher, man muss aber schon sehen, dass man es schafft, das Ganze für die komplette Organisation zu denken, und man muss auch gucken, dass ja, diese Bereitschaft, die man in dem Moment hat, wenn man sich mit der Low Code/ No Code-Plattform beschäftigt, dass die eben auch da bleibt, also nicht im Sinne von wir machen das jetzt ein halbes Jahr. Wir führen das ein und dann können wir alle Projektmitglieder da rausziehen, weil wir das jetzt leben, sondern das muss natürlich weiterbearbeitet werden. Es muss verbessert werden, es muss weiterhin begleitet werden.

EB07\_Janke, Tino: 22 - 23 (0)

Es gibt ja die unterschiedlichsten Organisationsformen. Wobei ich das, glaube ich, losgelöst von den Organisationsformen sehe. Also ich würde sagen, die Organisationsform ist nicht so wichtig. Wichtiger sehe ich, dass das Commitment bei allen relevanten Teilnehmern für dieses Thema da ist, also es muss halt klar sein, dass alle Projektmitglieder und der Sponsor am selben Strang ziehen und auch das gleiche Ziel vor Augen haben. Das sehe ich als wichtiger an, und dann ist mir das egal, ob das jetzt eine sehr hierarchisch geprägte Organisation ist oder ob die Organisation in Projekten arbeitet. Oder wie auch immer. Das sehe ich als entscheidenden Faktor an, dass eben alle relevanten Teilnehmer am selben Strang ziehen.

EB07\_Janke, Tino: 25 - 25 (0)

hingewiesen und das dies nicht als Thema von einzelnen betrachtet wird.

EB07\_Janke, Tino

3 - EP: Risiken und Hürden vor der Einführung\3.4 - Weitere Risiken

Ein anderes Problem, was wir manchmal haben, dass wir uns relativ schnell in einer internen IT-Bürokratie verlieren, weil eben eine Power Platform Silos aufbricht. So eine Power Platform betrifft nicht nur einzelne Teams. Das heißt, ich setze mich nicht nur mit einem Exchange Admin zusammen oder ich setze mich nicht nur mit einem Dynamics Admin zusammen oder mit einem SAP Admin zusammen. Eine Plattform heißt, ich betreffe die alle, das heißt, in vielen Fällen müssen die sich zum ersten Mal an einen Tisch setzen und gemeinsam über Regeln definieren, die sie bisher schön in ihrem Silo für sich verwalten konnten. Das funktioniert halt mit einer Plattform nicht und da haben manche Unternehmen dann ein Problem, zum einen die richtigen Leute zu identifizieren. Wen brauche ich denn da überhaupt? Und die dann auch zur Verfügung zu stellen. Und dann auch gemeinsam mal eine Entscheidung zu treffen, die nicht sechs Monate irgendwo in einem Bürokratienschungel untergeht. Gerade Banken sind da ein schwieriges Thema, da tatsächlich mal alle Kunden an einen Tisch zu bekommen, die richtigen Stakeholder, die dann auch gemeinsam eine Entscheidung treffen können, die eben nicht zwei Jahre dann durch interne Gremien durchlaufen muss. Das sind so meine zwei größten Hürden.

EB03\_Fabian Gackstatter: 25 - 25 (0)

Also die Frage ist auch immer, was ist denn für mich die richtige Plattform. Also was meine ich? Wir haben ja jetzt sehr viel allgemein über No Code/ Low Code-Plattform gesprochen und da gibt es ja verschiedene Hersteller, Microsoft, Mendix, Pega, da gibt es eine Unzahl von und die haben natürlich alle ihre Vor- und Nachteile. Sei es preislich, sei es von den Funktionen, von der Usability und ich bin jetzt bei Kunden auch schon darüber gestolpert. Also ich habe jetzt einen Kunden zum Beispiel gehabt, der auch gesagt hat: "Naja, die Plattform, die ich habe, ist das überhaupt noch die Richtige für mich? Oder bräuchte ich nicht eine andere?" Und das sind

Der Experte weist als Hürde auf eine zu starke IT-Bürokratie hin. Es wird an starren Vorgaben und Prozessen festgehalten, was zu einer Verzögerung des Projektes führt, weil lange Zeiträume vergehen, bis Prozesse entsprechend durchlaufen wurden. Die Einführung einer solchen Plattform bricht solche Strukturen auf und diese sollten dann neu evaluiert und geändert werden, um einen entsprechenden Fortschritt des Projekts zu gewährleisten.

EB03\_Fabian Gackstatter

Die Expertin weist auf die Relevanz einer ausführlichen Evaluationsphase zur Auswahl der Plattform hin. Ein Hürde kann sein, dass die falsche Plattform ausgewählt wurde und dies erst während des Einführungsprojektes bemerkt wird. Dann sollten Prozesse vorhanden sein, wie ein Exit von der Einführung erfolgen kann.

Ein weiterer Aspekt, den die Expertin herausstellt ist, darauf zu achten, wirklich alle relevanten Stakeholder von Anfang an mit einzubinden. Sie führt als Beispiel einen Erfahrungsbericht an, in dem der Betriebsrat nicht voll eingebunden wurde und aus diesem Grund

natürlich auch Entscheidungen, die man berücksichtigen muss. Welche nutze ich jetzt und aus welchen Gründen und wie gehe ich damit um, wenn die Plattformen sich weiterentwickeln und plötzlich die Plattform, die ich auch gewählt habe, nicht mehr so tauglich ist für meine Punkte. Habe ich dann Exit-Optionen? Lass ich es zu, dass es mehr als eine gibt? Kann ich mir das kostenmäßig leisten? Und das sind natürlich die Punkte, wo auch viele, dieser Auswahlprozess an sich, wenn ich nicht sage, ich habe einen Favoriten, aus welchen Gründen auch immer, das sollte auch gut überlegt sein.  
EB04\_Göttl, Simone: 29 - 29 (0)

eine Verzögerung im Einführungsprojekt erfolgt ist.  
EB04\_Göttl, Simone

Ich habe es sehr oft erlebt und fast in jedem Unternehmen. Das halt auch das, was mich sehr überrascht hat, weil ich das von einem theoretischen Ansatz nie gedacht habe. Die meisten Unternehmen kennen Ihre eigenen Prozesse nicht. Die machen was und dann sagen die, ich sage jetzt mal überspitzt und sehr salopp: "Ich krieg was von dem und geb es dem." Aber den gesamten Prozess, mal analysiert oder vielleicht sogar grafisch, wie du das jetzt hier gemacht hast mit deinem Entscheidungsbaum, dass einfach mal die Unternehmensprozesse graphisch aufgezeichnet werden, das existiert fast nie. Das ist gewachsen und wird so gemacht zum Teil, weil es schon immer so gemacht wurde. Sobald man versucht jetzt, mit einer Softwarelösung, egal welcher Art, Standardisierung hinzubringen und zu verstehen, was passiert denn eigentlich in unserem Abteilungsprozess, dann stößt man auf Grenzen. Das Problem ist also gar nicht so sehr die Umsetzung der Softwarelösung, sondern das, was davor gemacht werden muss. Nämlich die Aufbereitung der Prozesse und auch der Datenquellen, sage ich mal, damit man auch versteht, was machen wir hier eigentlich, von wo nach wo wandern unsere Daten und wie verarbeiten wir die. Und diese Vorarbeit wird aus meiner Sicht oft vernachlässigt. Man startet direkt mit der IT und dann steht man auf einmal vor einer Frage: "Soll ich

Der Experte hebt die Problemstellung hervor, dass viele Unternehmen ihre eigenen Prozesse nicht richtig kennen. Hier ist ursächlich, dass die Dokumentation nicht richtig durchgeführt wird oder Prozesse im Verlauf der Zeit sich weiterentwickelt haben, durch Mitarbeiter selbst und ohne das dies durch das Unternehmen gesteuert wurde. Er weist auf die Wichtigkeit einer Dokumentation der tatsächlichen Prozesse hin. Das Aufbrechen solcher veralteten Prozesse durch eine Standardisierung oder während der Implementierung der NCLC-Plattform kann zu weitreichenden negativen Auswirkungen für das Einführungsprojekt führen. Diese wirken sich auf Scope, Kosten oder Zeitplan aus.  
EB05\_Rohrbeck, Christian

jetzt linksrum gehen oder rechtsrum?"  
Und da muss auf jeden Fall, bevor man eine Softwarelösung einführt, sollte man sich diese ganze prozessuale Landschaft anschauen.

EB05\_Rohrbeck, Christian: 33 - 33 (0)

Die Risiken sind die, dass man zu wenig fehlertolerant ist. Das heißt, wenn ich jetzt hingehe und probiere etwas aus und das funktioniert jetzt nicht sofort, dann man dann sagt: "Mensch, das ist jetzt wieder alles schiefgelaufen und das ist alles Mist." Also diese Einstellungssache, Mindset gehört da auch rein und dass man dann sagt: "Ok, wir haben hier etwas entdeckt, was jetzt noch nicht so ganz gut funktioniert. Macht es noch Sinn, in dieser Richtung uns weiterzugeben? Oder müssen wir was ganz anderes als Lösung versuchen?" Also diese Start-up-Mentalität, die muss dann rein. Will ich dieses, wenn man es noch mal in so einem neudeutschen Ausdruck bezeichnen möchte, will ich dieses erste Produkt, was wir erstellt haben, dieses minimal viable product, will man das jetzt fortentwickeln, kommen da gute Feedbacks zu oder sagt man: "Naja, das war der falsche Weg." Aber selbst wenn es der falsche Weg war, dann muss man als Unternehmen sagen: "War eine klasse Idee. Haben wir probiert. War jetzt nicht mit dem Erfolg, den wir uns erwartet haben. Wir schauen jetzt nach was Neuem." Aber man sollte da nicht hingehen und sagen: "Der Meyer hat das wieder falsch gemacht." oder "Das war wieder eine Meyer-Idee." oder was auch immer. Sondern dass man wirklich sagt: "Mensch, von Meyer ist eine klasse Lösungsidee gekommen. Die hat noch nicht so funktioniert, aber vielleicht beim nächsten Mal, dass wir da besser uns drauf vorbereiten können." Also diese Fehlerkultur, die muss ausgeprägt sein und die muss den Mitarbeitern auch die Möglichkeit geben, wenn sie ihre Freiheit genutzt haben und lösungskompetent gehandelt haben, dass sie dann auch wirklich, dass diese Anstrengung auch belohnt worden ist. Sonst frustriert man

Eine Hürde, die durch den Experten geäußert wird, ist ein Mangel an Fehlerkultur und schlechten Feedback-Mechanismen. Diese Hürde kann entstehen, wenn im Vorfeld das falsche Mindset oder eine schlechte Unternehmenskultur gepflegt wurden.

EB08\_Fey, Peter



gerne und maßgeblich und nachhaltig und das ist dann auf jeden Fall zum Schaden für ein Unternehmen. Und ein nicht ganz erfolgreiches Projekt oder Programm haben wir in den Unternehmen auch in der IT-Ägide immer gehabt. Das wird es auch unter der No Code/ Low Code-Zeit geben. Und da muss man einfach als Unternehmen wissen: Ja, von den so und so viel Programmen, die wir machen, kommen vielleicht so und so viel Prozent nur durch. Aber das ist eine Einstellung, die man haben sollte und diese Fehler werden auf jeden Fall passieren.

EB08\_Fey, Peter: 21 - 21 (0)

4 - EP: Projektorganisationsform

**B: EH** Also, ich versuche ein gemischtes Team zusammenzustellen, das tatsächlich aus der IT Leute hat. Weil, ich brauche Leute, die die Plattform pflegen, die die Administration der Plattform machen, wo man darüber reden muss. Ich brauche Leute aus dem Fachbereich, die bereit sind, da mitzuarbeiten. Das muss jetzt kein 100 % Projektteam sein, das da mitarbeitet, weil meistens sind die Einführungen dann eher selber so eine Art agiles/ iteratives Projekt (**unverständlich**), das heißt, ich fange mit einem kleinen Piloten an. Probiere das ein bisschen aus. Guck mal, ob die Leute damit arbeiten können. Begleite das durch Schulungen und dann kann ich eigentlich zum Beispiel von Fachbereich zu Fachbereich gehen, oder mal strategisch zu überlegen, wie wollen wir das jetzt weiter ausrollen. Wo stellen wir das alles zur Verfügung, deshalb ist das schon eine Art Matrix-Organisation, aber typischerweise mit Teilzeit-Teams, die sich regelmäßig treffen, mal zusammensetzen. Das muss jetzt nicht kontinuierlich betreut werden. Also das ist jetzt nicht so, zumindest in dem Umfeld, in dem ich arbeite, ist das nicht so ein Riesenaufwand, diese Plattform einzuführen, wo ich dann wirklich Leute 3 Wochen lang oder ein halbes Jahr nur mit dem Projekt beschäftigen muss, sondern das kann ich schön nach und nach machen und das liebe ich eigentlich, das in kleinen Häppchen zu machen.

Der Experte stellt heraus, dass seine bevorzugte Organisationsform eine Matrixorganisation ist. Vor allem aus der Erfahrung in seinen Projekten hat es sich als vorteilhaft ist gezeigt, dass die Projektmitglieder nur teilweise im Projekt verankert sind. Diese Mitglieder organisiert er in Teams, die sich regelmäßig treffen und dann wieder zurück in ihre Fachbereiche gehen. Bei diesen Treffen spricht das Team über den Fortschritt, es werden Problemstellungen diskutiert und über den weiteren Rollout des Projektes beraten. Dieses Vorgehen begründet er mit dem iterativen Charakter, den er in seinen Projekten durchführt.

EB01\_Eckhard Hauenherm

EB01\_Eckhard Hauenherm: 39 - 39 (0)

Es ist wichtig, dass die, also es ist bei uns so, und es ist denke ich auch wichtig, dass die Administrierung aus einer IT-Einheit raus passiert.

**I: RS** Meinst du damit jetzt den Projektleiter? Mit Administrierung, wen meinst du damit, wenn du jetzt von klassischen Projektmanagementrollen sprechen müsstest?

**B: RR** Also der Teil Projektgovernance Administrierung von der Plattform. Ist für mich wichtig, dass es in der IT-Abteilung stattfindet, einfach um engen Kontakt zur IT zu haben, um Vertrauen zur IT zu schaffen und das nicht in Konkurrenz dazu aufzubauen, sondern aus der IT raus aufzubauen, um einfach Sachen vorzubeugen. Wie: Ihr macht da euer komplett eigenes Ding, das ist nur halb gar. Die Prozesse sind völlig an allen IT-Prozessen vorbei. Wie es jetzt bei uns ist, dass meine Rolle außerhalb der IT in einem extra Digitalisierungsteam angesiedelt ist. Was natürlich ganz wichtig ist, die Leute müssen Hand in Hand zusammenarbeiten. Ich finde es gut, dass ich nicht in der IT bin, einfach dadurch, dass ich, wenn ich das Citizen Development fördern will, einfach dass ich sagen kann: Das sage ich am Anfang von jeder Veranstaltung eigentlich, die ich mache, ich bin nicht in der IT, einer wie ihr, schreibe ich meistens darunter, weil ich eben auch aus dem Fachbereich komme, einfach, dass sie sehen, es muss niemand aus der IT sein, man kann da auch selber Erfolge generieren und von daher finde ich es in der Struktur gut, wie es ist. Bei uns passt das halt auch mit meinem persönlichen Hintergrund gut zusammen, würde ich jetzt aber nicht sagen, dass es anders nicht funktioniert, nur dass der Governance-Teil der Struktur, muss meiner Meinung nach in der IT angesiedelt sein.

EB02\_Rosengrün, Robin: 31 - 33 (0)

Und würdest du von Kundenseite die Mitarbeiter, die dem Projektteam

Der Experte deutet an, dass er die Projektmitglieder in ihren Abteilungen belassen würde, aber die Notwendigkeit sieht für ein Digitalisierungsteam, welches sich darum kümmert, dass der Kontakt zur IT-Abteilung erhalten wird und die entwickelten Prozesse nicht an den IT-Strukturen vorbei entwickelt werden.

EB02\_Rosengrün, Robin

Der Experte hebt die Notwendigkeit dafür hervor, dass ein Teil der

angehören, aus ihren originären Abteilungen herauslösen? Oder nur zu 50 % herauslösen, damit sie trotzdem noch die Basis haben? Wie händelst du das?

**B: FG** Teils, teils. Also manche Leute sollten sich wirklich Vollzeit dann auf die Plattform konzentrieren. Gerade was das Plattform Enablement angeht, die Admins, die wir brauchen. Die werden in der Regel stärker beansprucht. In den anderen Streams ist es unterschiedlich. Also an mancher Stelle brauche ich einfach nur die Expertise auf Kundenseite, um Entscheidungen zu treffen. Das kann ich natürlich nebenhermachen. Auf der anderen Seite, in Bereichen wie Change-management, wo es dann wirklich darum geht, Communitys zu betreuen, Trainings zu organisieren. Da ist es in der Regel auch über 50 % Leistung, die ich von einem Mitarbeiter brauche und da macht es in der Regel schon Sinn, ihn aus seiner aktuellen Rolle herauszulösen oder im Rahmen seiner aktuellen Rolle für das Thema Plattform zu dedizieren. Also wenn ich ein internes Changemanagementteam habe, dann muss ich ihn ja jetzt da nicht herausnehmen. Ich muss nur sicherstellen, dass er tatsächlich dann die 50 % bis 100 % Kapazität hat, um das Thema Changemanagement für die Power Plattform zu betreiben.

EB03\_Fabian Gackstatter: 30 - 31 (0)

Projektmitglieder in Vollzeit dem Projektteam angehören sollte. Dabei sieht er vor allem Teammitglieder für Enabling und Administration als Vollzeitteammitglieder. In anderen Bereichen ist es durchmischt. Entscheidungsträger und Fachabteilungsmitarbeiter können gegebenenfalls auch nur zu 50 % im Projektteam sein und aus ihrer originären Abteilung herausgelöst werden für diesen Aufgabenanteil. Wichtig ist, dass sichergestellt wird, dass der Mitarbeiter die Arbeitsanteile, die er für das Projekt freigestellt wird, auch wirklich für das Projekt nutzt.

EB03\_Fabian Gackstatter

**B: SG** Ich glaube, das ist immer händelbar, weil es auch aus Erfahrung, es ist ja in anderen Projekten auch nicht anders gewesen. Ob man es jetzt agil oder was weiß ich, sondern es war schon immer so, dass es ein Projekt gibt und ich glaube, was dort wichtig ist, Zugehörigkeit ist immer die Frage: Fühle ich mich dem Projekt zugehörig oder meiner Heimat, wo ich aufgehoben bin? Und ich glaube, man kann zwei Identitäten haben und sollte es auch, weil, das Projekt ist ja irgendwann ausgelaufen und irgendwann fertig, da ist die Plattform zumindest eingeführt, da ist ja meine Identität in gewisser Weise weg. Deshalb, glaube ich, sehe ich da keine Konflikte und wenn ein Mitarbeiter dort Konflikte hat zwischen

Die Expertin hebt hervor, dass es vor allem zum Zwecke der Skalierung, der Wissensmitnahme aus dem Fachbereich sinnvoll ist, die Projektteammitglieder zu einem gewissen Grad in ihren Abteilungen zu belassen. Sie stellt jedoch heraus, dass es wichtig ist, dass es von allen Teammitgliedern ein Projektcommitment geben muss, damit dies funktioniert. Es muss ein Zugehörigkeitsgefühl zu beiden Seiten herrschen, damit der Projekterfolg nicht gefährdet wird.

EB04\_Görtl, Simone

Projekt und Linie, weil jetzt vielleicht 50/50, 60/40 aufgeteilt ist, ist das natürlich auch ein klarer Punkt, den man an eine Projektleitung oder an einen Product Owner bzw. an einen Teamleiter adressieren muss, dass es dort Konflikte gibt und diese auflösen können muss. Aber von der Identität her, es sei denn das ist ein Mitarbeiter, der komplett immer nur Projekte arbeitet, also wie so eine Art inhouse Consulting, der wird sich da schon wohlfühlen. Ich würde einen Mitarbeiter nicht bis ins letzte raus holen, gerade dann, wenn es um Low Code/ No Code- Anwendungen geht und vielleicht er dann natürlich die frohe Botschaft in den Fachbereich zurücktragen kann, wie toll das alles ist.

EB04\_Görtl, Simone: 33 - 33 (0)

Also, die Frage ist nicht so leicht zu beantworten, weil so ein bisschen Wunschenken gegen Realität steht. Und ich weiß, dass ich die Realität nicht ändern kann. Ich sage mal, wenn man eine projektorientierte Organisationsform hat, ist man in einer Art Systemhaus. Es gibt für einen Kunden ein Projekt und dann werden entsprechende Ressourcen zusammengezogen und sind auch voll und ganz dafür zuständig. Das ist natürlich eine Idealform, sage ich mal, dann ist man zu 100 % dabei und wird auch darüber den größten Output generieren und auch, ich sage mal, mit einer guten Qualität, also wird die eigentlichen Projektziele On Time, On Budget, On Scope, die wird man damit am ehesten erreichen. Aber in der Praxis, würde ich schon fast sagen, ist das utopisch, man wird vermutlich nie über eine Matrixorganisation hinauskommen, weil einfach die Unternehmen, in der Regel, sich dem Produktionsbetrieb, also den Operations, zugewandt haben und ein Projekt ist immer on Top. Also man muss verstehen, ein Projekt ist für die meisten Mitarbeiter eine Belastung. Die müssen auf einmal aus ihren gewohnten Operations raus, müssen sich mit neuen Inhalten oder neuen Vorgehensmodellen konfrontieren, haben oft auch eine Abwehrhaltung

Der Experte denkt, dass ein 100 % herauslösen der Projektmitglieder die Idealform ist, welche das Projektergebnis erzielen würde. Unter Projektziele versteht er die Ziele der traditionellen Vorgehensweise On Time, On Scope, On Budget. Er weist jedoch darauf hin, dass dies in der Praxis eine Wunschorganisation ist und so selten bis nie in der Realität vorkommen wird. Er tendiert an Praxis orientiert eher zu einer Matrixorganisation, weil Unternehmen in der Regel ihren Produktionsbetrieb über das Projekt stellen. Er stellt heraus, dass für die meisten Projektmitarbeiter ein Projekt eine zusätzliche Belastung ist, welche sie neben ihren originären Aufgaben erfüllen müssen. Dies führt unter anderem zu einem gewissen Widerstand gegen das Projekt. Dem sollte vorgebeugt werden. Aufgrund dieser Orientierung der Unternehmen sollten Mitarbeiter in ihren Abteilungen bleiben, weil dadurch Fachwissen mit ins Projekt getragen wird. Es wird dazu geraten, einen Mitarbeiter der IT-Abteilung vor Ort zu haben, der bei Fragen und Absprachen hilft.

EB05\_Rohrbeck, Christian

dagegen, weil es ja eine Änderung mit sich bringt. Also, auch das ist ein Thema, was man von der emotionalen Seite vorab beleuchten muss. Wie groß ist der Änderungswille in einem solchen Unternehmen, weil oft auch da Leute im Prinzip das blockieren können und auch die Stimmung herunterreißen können. Vielleicht kannst du mir deine Frage, ich schweife wieder ab, merke ich gerade ein bisschen. Vielleicht kannst du mich noch mal konzentriert auf den Kern zurückbringen?

EB05\_Rohrbeck, Christian: 35 - 35 (0)

Gut also, also danke erst mal, dass du mich auf die Spur zurückgebracht hast. Also ich denke, die meisten, zumindest in größeren Unternehmen, sind in Abteilungen organisiert und die werden auch in ihrem Daily Business dann in dieser Abteilung bleiben. Es macht am meisten Sinn, die auch dort während der Projektphase zu belassen, weil sie da das Fach-Know-how haben und sie einfach durch eine IT-Person vor Ort in dieser Abteilung zu unterstützen oder einen Coach, der sie da anlernen kann. Das ist für diese Art der Entwicklung das beste Vorgehen.

EB05\_Rohrbeck, Christian: 37 - 37 (0)

Das fällt mir schwer, dazu eine pauschale Aussage zu treffen, weil ich finde, dass es weniger eine Fragestellung danach ist, ob es eine Low Code-Lösung oder keine Low Code-Lösung ist, sondern da würde ich mich eher nach anderen Kriterien orientieren. Ja, also ich sage jetzt mal einfach nur, um ein Beispiel zu geben. Der eine Kunde möchte beispielsweise einen Urlaubsantrag, zum Beispiel über eine Low Code-Lösung erhoben wird und die Urlaubsanträge sollen danach in einen Kalender hineinfließen. Und das ist ein klitzekleiner Aufwand, ich sage jetzt mal, das ist innerhalb von einem halben bis einem Tag entwickelt und es ergeben sich auch aus

Der Experte sieht sich nicht in der Lage, dieses Thema allgemein zu beantworten. Für ihn ist es losgelöst von der Fragestellung Einführung NCLC und eher abhängig, wie ein Unternehmen selbst vorgeht und wie groß der Anspruch am Ende an das Projekt ist.

EB06\_Hesse, Moritz

dem Prozess keine weiteren Fragestellungen. Dann ist mir persönlich am liebsten, wenn ich einfach nur den Ansprechpartner vor Ort habe, der das gelöst haben möchte, mit dem kann ich mich kurz irgendwie austauschen. Ganz anders sieht das aber aus. Ein Kunde, der möchte das aus einem SAP-System täglich Daten herausfallen. Die sollen synchronisiert werden in den Datenbestand unserer Lösung. Der Mitarbeiter soll mit unserer Lösung arbeiten, soll dort bestimmte Werte eintragen und am Ende des Tages sollen die Daten dann wieder in das SAP zurückgemeldet werden. Und dass eine war, keine Ahnung, eine kleine Kfz-Werkstatt und das andere ist ein großer DAX-Konzern. Und deshalb, bei dem einen bin ich sehr froh, wenn es schnelle, kurze Dienstwege, wenn was nicht klappt, ruft der mich an, ich kann noch mal reingucken, kann das beheben. Bei dem anderen Kunden, wenn ich da was im Automatismus vielleicht falsch hinterlege, führt das dazu, dass 20 LKWs losrollen, obwohl nur 1 losrollen sollte und mir stellt plötzlich jemand eine Rechnung ins Haus, weil das nicht gut umgesetzt wurde. Dann bin ich natürlich schon gewillt, ein anderes Maß an Dokumentation, Verbindlichkeit, Entscheidungsnachvollziehung, ja, dass jemand eine bestimmte Entscheidung getroffen hat, der entscheidungsbefugt war. Deshalb, das sind Kriterien, die würde ich eher an ganz allgemein, Fragestellung aufmachen, wann ist im Projektmanagement ein PMO erforderlich, wann brauche ich eine Meilensteinplanung. Diese Fragestellung würde ich weniger davon abhängig machen, ob es sich um eine Low Code-Lösung oder eine nicht Low Code-Lösung ist, sondern nach allgemeineren Sachen. Aber ich würde hier auf jeden Fall wieder ins Spiel bringen, Mindset. Das heißt, ich würde behaupten, in einem Umfeld, wo ich Low Code-Lösungen einsetze, könnte es durchaus den Prozess befruchten, das Projekt befruchten, wenn das Mindset insgesamt agil ist und wenn man nicht quasi an alten Strukturen festhält. Aber das ist auch eine

schwierige Aussage, die individuell sicherlich nicht immer zutrifft.

**I: RS** Ok, ich entnehme der Aussage, dass sie sagen: Es ist sehr situationsabhängig in welchem Unternehmen man ist und wie da die Voraussetzungen sind.

**B: MH** Ja.

EB06\_Hesse, Moritz: 37 - 39 (0)

Ich glaube, am Anfang ist es wichtig, dass sie in ihren Fachabteilungen bleiben. Dass sie einfach den Bezug zu den Fachthemen haben. Es sollte ja so sein, dass man mit dem Thema schnell Erfolge produziert, dafür ist es halt wichtig, dass man auch Leute an Bord hat, die wissen, was in der Unternehmensrealität wesentlich ist, die auch wesentliche Probleme in der Unternehmensrealität kennen und wie man diese gegebenenfalls mit einer Low Code/ No Code-Plattform lösen kann. Und wenn man das einmal schafft, so eine Lösung zu kreieren, dann kann man natürlich auch durchs Unternehmen gehen und sagen, dass man im Prinzip diese Ergebnisse mithilfe der Low Code/ No Code-Plattform geschaffen hat und ja, für weitere Unterstützer sozusagen werben. Also am Anfang würde ich die Teammitglieder/ Projektmitglieder in ihren jeweiligen Abteilungen belassen und dann eher zu einem gewissen Prozentsatz für dieses Thema freistellen.

EB07\_Janke, Tino: 29 - 29 (0)

Also herauslösen aus diesen Zusammenhängen, in denen die im Unternehmen sind, ist eigentlich dort nicht vorgesehen, sondern da finden zu ganz gewissen Zeiten Workshops statt. Es werden Ideation Workshops durchgeführt. Das heißt, was wollen wir überhaupt erreichen? Dann geht das dann an einen oder zwei Menschen innerhalb dieser Abteilung, die die sogenannte Citizen Developer- Ausbildung gemacht haben. Die dann sagen, ich setze das mal an der Tastatur mit diesem Low Code/ No Code-Programm um. Dann käme die nächste Station innerhalb dieses

Der Experte vertritt die Meinung, dass vor allem zu Beginn des Projektes die Projektmitglieder in ihren Abteilungen belassen werden sollten und das Fachwissen der Abteilungen mitbringen sollten, um mit diesen Use Cases für das Unternehmen zu entwickeln und den Mehrwert der Plattform aufzuzeigen. In einem späteren Verlauf und einem entsprechend fordernden Scope des Projektes können einige Teammitglieder Vollzeit für das Projekt freigestellt werden.

EB07\_Janke, Tino

Es wird durch den Experten darauf hingewiesen, dass es aus der Praxis kein Unternehmen es sich leisten kann, seine Mitarbeiter völlig für das Projekt freizustellen. Es wird eher das Konzept vertreten, in bestimmten zeitlichen Iterationen das Projektteam zusammenzuziehen und dort über Erfolge, Misserfolge zu sprechen sowie das weitere Vorgehen.

EB08\_Fey, Peter

Workflows und dann würden die Leute wieder zusammengezogen. Aber das wäre jetzt keine tagesfüllende Beschäftigung, sondern das wäre tatsächlich etwas, was man innerhalb der üblichen Arbeitszeit erbringen kann. Vielleicht eben mit der Möglichkeit, die Kollegen dann in den Workshops zusammenzuziehen und dann auch entsprechende Ergebnisse zu haben. Wie man dann mit den Citizen Developern umgeht, muss man schauen. Also das ergibt sich halt auch aus dem Arbeitsumfang. Aber die normale Arbeit der Fachbereiche, die sollte eigentlich ungestört sein und ich kenne im Moment kaum ein Unternehmen, was sagen kann: "Ich habe so viele Mitarbeiter in meinen Fachbereichen, die leistungsstark sind, die ziehe ich ab und die bringe ich jetzt alle nur noch in die Programmentwicklung unter." Dann wäre nichts gewonnen.

EB08\_Fey, Peter: 25 - 25 (0)

5 - Wahl Vorgehensmodell\5.1 - Agiles Vorgehensmodell

Der Experte stellt heraus, dass er ein agiles Vorgehensmodell als das beste Vorgehensmodell sieht. Der Experte hebt hervor, dass er ein traditionelles Vorgehen als nachteilig sieht, da diese Art des Projektes eine schlechte Planbarkeit hat und diese als Voraussetzung für das Projekt gesehen wird. Diese schlechte Planbarkeit begründet sich darin, dass es sich um ein neuen Tool handelt und davon ausgegangen wird, dass das Unternehmen nicht völlig weiß, was es damit alles tun kann und dies erst während der Einführung herausfindet.

Als weiteren Punkt, der für ein agiles Vorgehen angeführt wird, ist, dass die Plattformen agil entwickelt werden. Die Plattformen werden ständig weiterentwickelt und damit kommen regelmäßig neue Funktionen auf die Plattformen, weshalb in iterativen Schleifen immer wieder neu evaluiert werden muss, was jetzt mit der Plattform möglich ist. Ein weiterer Vorteil von agilen Projekten ist die Möglichkeit, mit einem geringeren Aufwand eine Exit-Strategie zu erstellen.



Es wird durch den Experten darauf hingewiesen, dass ein hybrides Vorgehen mit agilem Schwerpunkt möglich ist. Dies könnte Vorteile haben vor der Einführung in der Planungsphase, bei Themen wie Governance und Risikosteuerung. Jedoch wird hier kein Mehrwert in einem hybriden Vorgehen mit Wasserfallanteilen gesehen, da diese Fragen auch in einem agilen Vorgehen geklärt werden können.

EB01\_Eckhard Hauenherm

Der Experte favorisiert ganz klar einen agilen Ansatz. Es kommt zu einer Situation, bei der Wasserfallanteile eingesetzt sind für Introductionworkshops, aber selbst diese Workshops werden agil durchgeführt.

Es wird durchweg in Sprints gearbeitet, mit einem Sprint 0, in dem die Introductionworkshops stattfinden und danach direkt mit der Abarbeitung von Arbeitspaketen der Einführung begonnen. Die Bearbeitung der Themengebiete erfolgt in Teams und können als themenabhängige Scrumteams verstanden werden. Dies ermöglicht ein parallelisiertes Bearbeiten der verschiedenen Themengebiete der Einführung, ohne dass eine Abhängigkeit von einer vorgeschalteten Phase nötig ist. Die Ausnahme bildet der benannte Sprint 0.

EB03\_Fabian Gackstatter

Also was wir jetzt die letzten Jahre auch für Software-Einführungsprojekte, und das ist ja am Ende ein Software-Einführungsprojekt, oder solche Themen gesehen haben, ist einmal die Unterscheidung zwischen klassischem Projektmanagement und agilem Projektmanagement. Was wir jetzt auch häufig schon sehen bei den Unternehmen, dass die auch grundsätzlich ihre IT- Changeprojekte nicht mehr wasserfallartig orchestrieren. Das heißt (**unverständlich**). Es gibt klarere Roadmap, es gibt ein Fachkonzept, dann gibt es ein EV-Konzept und dann wird das umgesetzt, sondern dass da auch agile Methoden, wie Scrum oder wie Spotify oder wie das SAFe Framework angewendet werden. Und

Die Expertin bevorzugt klar ein agiles Vorgehen.

Sie beschreibt beim Vorgehen eine Scrum-Vorgehensweise. Es gibt in diesem Zusammenhang dedizierte Teams, die sich, um die Governance-Einführung kümmern, Teams für technische Umsetzung. Sie hebt hervor, dass es nicht zwingend sinnvoll ist, die Mitarbeiter zu 100 % aus ihren Abteilungen herauszulösen, da beim Scrum volles Commitment erwartet wird. Daher schlägt sie vor, die MA nur während des Sprints aus den Abteilungen zu lösen. Dadurch wird der positive Effekt generiert, dass weiter das Fachwissen der Abteilung im Projekt vorhanden ist.

EB04\_Görtl, Simone

das ist das Erste, das aus meiner Hypothese auch besser geeignet ist für diese Plattform, auch wenn manche Ziele relativ eindeutig sind und wir nicht wie bei einer normalen App sind, Anforderungen sind unklar. Aber das ist etwas, was auf alle Fälle, was ich machen würde, von der Projektorganisation ist, glaube ich, dass auch, was wir sehen, was ein Erfolgsfaktor ist. Ich würde auch nicht mehr von einem Projektleiter sprechen, sondern von einem Product Owner für diese Plattform und das ist im Idealfall der, der sie auch später dann in der Linie, wenn es eingeführt ist, übernimmt. Klar ist es immer wichtig, darüber eine Art Stakeholder-Steuerungskreis zu haben, Sponsoren, die auch Entscheidungen treffen können, das heißt, das sind dann entweder Bereichsleiter, Vorstände oder wie auch immer das in dem Unternehmen aufgesetzt ist, um auch Entscheidungen herbeiführen zu können. Und vom grundsätzlichen Team, die dann in dem Scrum Team sind, zur Einführung, ist immer so ein bisschen die Frage, wie man aufgestellt ist. Was meine ich damit? Es macht vielleicht auch manchmal keinen Sinn, einen Fachbereichsmitarbeiter für eine Einführung komplett aus seiner Umgebung herauszuholen, weil er dann den Bezug dazu vielleicht verliert, andererseits sagt man, wenn man agil unterwegs ist: Mitarbeiter sollen voll committed sein, das heißt, die sollen eigentlich zu 100 % in einem Scrum-Team arbeiten. Das heißt, ich würde das immer so schneiden, dass es ein Scrum-Team gibt, was sich grundsätzlich um die IT kümmert, um die Governanceeinführung, um die technische Umsetzung, ich hätte dann aber hintendran ein Team, das quasi auch einen ersten MVP, einen ersten Prototyp für den Fachbereich mit dem Fachbereich entwickelt. Und das ist ein wichtiger Punkt und von der Organisation her definitiv auch an ein Backlog zu denken und auch das wirklich auch klar zu definieren: Wann ist Definition of Done? Bis wann soll was erledigt sein? Und auch wirklich dort das mit aufzunehmen. Das wären so wichtige Eckpfeiler vom Projektmanagement. Also wirklich

nicht klassisch, sondern agil, aber so agil adaptiert, wie es auch für Unternehmung oder den Konzern passt, weil, Scrum gibt Leitplanken, zum Beispiel, aber es ist ja nicht so, dass man es unbedingt so machen muss, sondern es gibt dann auch bestimmte Kulturen, Werte oder Organisationsformen, die das einfach in einer großen Unternehmung vielleicht nicht anders zulassen.

EB04\_Göttl, Simone: 31 - 31 (0)

Der Experte hebt hervor, dass er, obwohl er einen hybriden Ansatz mit Wasserfallschwerpunkt bevorzugt, die agilen Methodiken, wenn möglich lieber einsetzen würde. Dies ist aber stark abhängig, wie die Unternehmenskultur ist und ob der Ansatz sich als praktikabel erweist. Wenn eine agile Vorgehensweise möglich ist, empfiehlt der Experte agil anzufangen und zu evaluieren, wie das funktioniert. Sollte sich eine agile Methodik als nicht nützlich erweisen, würde er Stück für Stück traditionelle Ansätze verwenden, bis ein optimaler Punkt gefunden ist zwischen agilem und traditionellem Projektmanagement.

EB05\_Rohrbeck, Christian

Der Experte bevorzugt klar eine agile Vorgehensweise. Dies begründet er damit, dass traditionelle Ansätze zu starr und unflexibel sind. Als Stärken von NCLC-Plattformen werden die Flexibilität und Schnelligkeit der Plattform genannt. Diese Fähigkeiten ermöglichen eine bessere Reaktionsgeschwindigkeit auf veränderte Bedingungen. Diese Stärke sollte sich im gewählten Projektmanagementmodell widerspiegeln. Dies wird als begünstigend für das Mindset des Projektes gesehen. Traditionelle Ansätze hingegen sind zu starr. Als Vorteil in der Softwareentwicklung wird gesehen, dass es zwar Phasen gibt, wie Planung, Durchführung, Testing, Release und Problembehebung, diese aber in wesentlich kürzeren Zyklen, wesentlich öfter durchlaufen werden. Dies ermöglicht eine schnelles Reagieren auf geänderte Anforderungen. Der Experte verweist darauf, dass

Anforderungen an NCLC-Plattformen eine so hohe Volatilität aufweisen, dass ein klassischer Ansatz völlig ungeeignet dafür wäre.

Der Experte verweist darauf, dass es durchaus aber eine Option ist, das Einführungsprojekt als Gesamtprojekt in einen Wasserfallrahmen zu packen, aber die aktive Phase der Implementierung als agil umzusetzen. Dies würde das Modell zu einem hybriden Ansatz machen.

EB06\_Hesse, Moritz

Der Experte hebt hervor, dass er eher agiles Vorgehen bevorzugt. Dies begründet er mit dem iterativen Charakter der agilen Ansätze. Bei der aktiven Einführung geht es darum, schnell Ideen zu erfassen und eine Lösung zu entwickeln durch einen späteren Nutzer der Plattform. Diese Art der Entwicklung wird in wiederkehrenden Workshops durchgeführt. Danach kommt es zu Fachinterviews, wo durch den Citizen Developer die Fachabteilungen befragt werden und ein schnelles Feedback erhalten, um dieses in die nächste Iteration mitzunehmen.

EB08\_Fey, Peter

5 - Wahl Vorgehensmodell\5.1 - Agiles Vorgehensmodell\5.1.1 - Option auf Wasserfallelemente

Zu einem hybriden Modell... Kann man muss man aber nicht. Also man könnte natürlich tatsächlich für so ein Projekt auch ein Wasserfall drum herum legen, sage ich mal, für so ein hybrides Modell. Sehe ich aber tatsächlich keinen Mehrwert drin. Es gibt ein paar Dinge die ich Anfangs in der Zielsetzung machen würde, die vielleicht so ein bisschen nicht hundertprozentig Agil sind und da ist wieder mein Lieblingsthema Governance da, das sollte man am Anfang entscheiden wie viel Risiken gehe ich damit ein und wie kann ich die Risiken streuen. Da muss am man tatsächlich ganz klar am Anfang mit IT und Operations und was weiß ich, Security Officer, wer auch immer im Unternehmen dafür zuständig ist, sich wirklich mal klar machen, welche Datenschutzerfordernungen haben wir? Mit welchen Daten gehen wir um? Manchmal führt das zu einem Vorlauf, in

so einem Projekt, das ich erstmal mit Datenklassifikationen und so weiter anfangen muss zu arbeiten. Das ich sicherstelle das bestimmte Daten da nicht rauskommen. Aber das kann man auch Agil nach und nach machen in dem Mann, wenn man das Risiko kennt, tatsächlich guckt wo setzen wir das ein? Wem gebe ich Zugriff auf die Plattform und so weiter? Das kann man schon noch steuern.

EB01\_Eckhard Hauenherm: 43 - 43 (0)

Also rein agil oder sagst du, du willst auch Elemente aus dem Wasserfallmodell haben? Keine Ahnung, Phasen, weil die einen Vorteil bringen.

**B: FG** Eigentlich nicht. Also, wie gesagt, der einzige Wasserfallanteil ist am Anfang so diese Introworkshops, die wir mit dem Kunden machen, über zwei bis drei Wochen, um einfach die richtigen Basisvoraussetzungen zu schaffen und danach geht es aber in den agilen Modus über, wenn wir definieren: "Ok, das sind jetzt die Applikationen, die wir bauen möchten, als Teil der Einführung. Das sind die Organisationsstrukturen, die wir bauen möchten. Das ist das Tooling, das wir selber anfassen müssen." Und das ist aber ein rein agiles Vorgehen an der Stelle.

EB03\_Fabian Gackstatter: 34 - 35 (0)

**B: MH** Ich bin grundsätzlich ein Freund von agilen Vorgehensweisen. Das ist aber weniger meiner Erfahrung aus der Low Code-Ecke geschuldet, sondern das ist einfach eine sehr persönliche Ansicht, aus meiner Erfahrung als Softwareentwickler, mit den Vorteilen oder auch Nachteilen, die die agile Softwareentwicklung oder das agile Projektmanagement mit sich bringen. Ich glaube, bei der Einführung und der nachhaltigen Nutzung von Low Code-Lösungen tun sich Unternehmen keinen Gefallen, an einem starren Wasserfallmodell festzuhalten. Lassen Sie mich ausführen, warum. Eine wesentliche Stärke von Low Code-Lösungen zeichnet sich durch diese Flexibilität und Schnelligkeit aus. Diese

Schnelligkeit ermöglicht den Kunden, zum Beispiel, sehr, sehr schnell auf regulatorische Anforderungen reagieren zu können. Das klassische Wasserfallmodell, das klassische Projektmanagementmodell, hat genau diese Schwäche, dass es nicht flexibel und schnell ist. Sondern im Gegenteil, es baut sich genau an diesen Projektphasen auf, wo die nachgelagerten Phasen immer erst den Abschluss vorgelagerten Phasen erfordern und genau das sind ja die Vorteile der agilen Softwareentwicklung. Dass man quasi durchaus auch diese ganz klassischen Phasen hat. Planung, Durchführung, Testing, Release und die Problembehebung, aber die Zyklen sind einfach viel kleiner. Man hat ja quasi diese klassischen Sprints, im Scrum-Verfahren, im Scrum-Modell, und die ermöglichen einfach ein viel schnelleres Reagieren auf einen Wechsel der Anforderungen und ich habe selten eine so hohe Volatilität der Anforderungen im Verlauf der Implementierung erfahren, wie ich das mit Kunden, dieser Lösung hier, erfahre. Einfach, das bringt das Produkt halt mit sich. Das sind Kunden, die wollen ja Flexibilität, die wollen Schnelligkeit und es ist selten so, dass ich am Ende des Projekts die Anforderung noch so habe, wie sie ursprünglich genannt wurden. Und insofern tut sich ein Unternehmen bei der Einführung von No Code/ Low Code- Lösungen glaube ich keinen großen Gefallen, wenn es zu starr an den alten Strukturen, an alten Modellen, festhält und würde immer eher die agilen Modelle, quasi empfehlen, wenn gleich man hier auch noch einmal unterscheiden kann zwischen der Einführung der Lösung als solche. Das könnte man ja durchaus als ein größeres Projekt im Wasserfallrahmen irgendwie realisieren, wenngleich ich aber insbesondere die Anwendung und dann später die Realisierung von Projekten, von echten Projekten mittels Low Code-Lösungen, dann doch eher agil einführen würde. Also wenn ich dafür die Entscheidungsbefugnis hätte, würde ich mich eher auf ein agiles Modell orientieren.

EB06\_Hesse, Moritz: 41 - 41 (0)

5 - Wahl Vorgehensmodell\5.1 - Agiles Vorgehensmodell\5.1.2 - Begründung agiles Vorgehen

Tatsächlich agil. Also das würde ich tatsächlich nicht als Wasserfall machen. Erstens, weil ich bei der Einführung zwar Ziele mir festlegen kann. Das mache ich bei jedem Cloud-Projekt, dass sozusagen die erste Frage ist, warum wollt ihr in die Cloud? Warum wollt ihr in diese Plattformen? Also gucken, was will das Unternehmen damit erreichen? Ich habe aber eine relativ schlechte Planbarkeit, weil ich erstens nicht genau weiß, welche Funktionen jetzt damit umgesetzt werden können oder wollen oder wo das Unternehmen arbeitet. Das spricht schon gegen einen klassischen, sequenziellen Wasserfallprozess, wo ich ja in der Regel ein fertiges Gewerk am Ende plane. Das habe ich hier nicht, sondern es ist eine Einführung einer Plattform, die sich dann sozusagen immer weiterentwickelt. Und die Plattformen selber werden ja bei den Anbietern auch agil entwickelt. Also so eine Power Platform ist ja auch nicht fertig und ich führe die ein, sondern da kommen jeden Monat neue Funktionalitäten rein. Das heißt, ich kann das eigentlich nur agil machen und immer gucken, in kleinen Zielen: "Wo gucken wir jetzt? Was machen wir als nächstens? Was nehmen wir uns für die nächsten 4 Wochen vor oder 2 Wochen?" Also klassisches Scrum-Projekt oder klassisches agiles Projekt und das passt auch am besten zu den Plattformen selber. Das passt dann gut zusammen und man muss ja auch mit dem Produkt dann lernen. Das ist ja genau das, ich habe mit einem iterativen Projekt meine Feedbackschleife und kann dann sehr schön mit der Plattform lernen, was uns die im Unternehmen bringt oder nicht bringt. Und ich kann auch mit relativ wenig Kosten schnell wieder aussteigen, wenn ich sage: "Nee, das ist für uns nichts." Oder "Die Plattform ist für uns nichts. Wir nehmen mal eine andere." Das kann ich bei einem fertig geplanten, also bei einem klassischen, sequenziellen Projekt habe ich das nicht. Da habe ich am Ende alles eingeführt und keiner kann damit arbeiten. Das wollen wir nicht.

EB01\_Eckhard Hauenherm: 41 - 41 (0)

**B: FG** Also, wir sind normalerweise agil unterwegs. Das heißt, wir fangen an mit einem Sprint Null, in dem wir erst mal eine Reihe von Workshops durchführen, um den Kunden auf ein gewisses Niveau aufzuschauen und nach diesem Sprint Null planen wir dann wirklich Entwicklungssprints, um zu sehen: "Ok, das sind jetzt die Arbeitspakete, die wir tatsächlich implementieren möchten, in dem und dem Zeitraum." Auch immer dem Umstand geschuldet, wie viel Mitarbeit habe ich jetzt auf Kundenseite. Also, habe ich tatsächlich dedizierte Ressourcen, die mir zu 100 % zur Verfügung stehen. Das heißt, ich kann auch wirklich jeden Tag Arbeitspakete abarbeiten. Oder habe ich eben nur Leute, die maximal 50 % zur Verfügung stehen, dann brauche ich natürlich entsprechend länger, um die Pakete umzusetzen. Aber wir versuchen es normalerweise schon in einer agilen Methode umzusetzen, einfach auch, weil die Pakete sehr unterschiedlich sind und die jetzt in einem Wasserfall anzubringen, wäre sehr schwierig.

EB03\_Fabian Gackstatter: 33 - 33 (0)

Aber sagen wir mal, wir sind in einem sehr, sehr jungen, agilen Unternehmen, das schon voll dabei ist, wo die eingespielt sind, wo die auch schon erfolgreich ein agiles Management angewendet haben. Würdest du da sagen dort ist eher der agile Anteil im Übergewicht bzw. nur agiler Ansatz sinnvoll? Oder würdest du da auch sagen, trotzdem Wasserfall?

**B: CR** Ja, also ich stehe dem Agilen gar nicht Kontra gegenüber, im Gegenteil, ich finde das sehr gut. Ich würde in so einem Fall, es ist halt immer schwierig, wie lange man in dem Unternehmen arbeitet und wie sehr man das kennt. Es hängt aber wirklich sehr viel von den Kollegen ab, gerade der Kultur und dem Know-how. Wenn ich, angenommen, neu reinkommen würde, würde ich mich vorarbeiten. Ich würde sehr agil, glaube ich, anfangen und dann schauen, wie es läuft. Und wenn man merkt, es funktioniert nicht, dann könnte man da



unterstützend, sage ich mal, eingreifen, indem man peu à peu anfängt, dann da eventuell einen Weg vorzugeben, bis zu einem Punkt, wo man sieht, ab da geht es wieder selbst dirigiert.

EB05\_Rohrbeck, Christian: 42 - 43 (0)

**B: MH** Ich bin grundsätzlich ein Freund von agilen Vorgehensweisen. Das ist aber weniger meiner Erfahrung aus der Low Code-Ecke geschuldet, sondern das ist einfach eine sehr persönliche Ansicht, aus meiner Erfahrung als Softwareentwickler, mit den Vorteilen oder auch Nachteilen, die die agile Softwareentwicklung oder das agile Projektmanagement mit sich bringen. Ich glaube, bei der Einführung und der nachhaltigen Nutzung von Low Code-Lösungen tun sich Unternehmen keinen Gefallen, an einem starren Wasserfallmodell festzuhalten. Lassen Sie mich ausführen, warum. Eine wesentliche Stärke von Low Code-Lösungen zeichnet sich durch diese Flexibilität und Schnelligkeit aus. Diese Schnelligkeit ermöglicht den Kunden, zum Beispiel, sehr, sehr schnell auf regulatorische Anforderungen reagieren zu können. Das klassische Wasserfallmodell, das klassische Projektmanagementmodell, hat genau diese Schwäche, dass es nicht flexibel und schnell ist. Sondern im Gegenteil, es baut sich genau an diesen Projektphasen auf, wo die nachgelagerten Phasen immer erst den Abschluss vorgelagerten Phasen erfordern und genau das sind ja die Vorteile der agilen Softwareentwicklung. Dass man quasi durchaus auch diese ganz klassischen Phasen hat. Planung, Durchführung, Testing, Release und die Problembehebung, aber die Zyklen sind einfach viel kleiner. Man hat ja quasi diese klassischen Sprints, im Scrum-Verfahren, im Scrum-Modell, und die ermöglichen einfach ein viel schnelleres Reagieren auf einen Wechsel der Anforderungen und ich habe selten eine so hohe Volatilität der Anforderungen im Verlauf der Implementierung erfahren, wie ich das mit Kunden, dieser Lösung hier, erfahre. Einfach, das bringt das Produkt halt mit sich.

Das sind Kunden, die wollen ja Flexibilität, die wollen Schnelligkeit und es ist selten so, dass ich am Ende des Projekts die Anforderung noch so habe, wie sie ursprünglich genannt wurden. Und insofern tut sich ein Unternehmen bei der Einführung von No Code/ Low Code- Lösungen glaube ich keinen großen Gefallen, wenn es zu starr an den alten Strukturen, an alten Modellen, festhält und würde immer eher die agilen Modelle, quasi empfehlen, wenn gleich man hier auch noch einmal unterscheiden kann zwischen der Einführung der Lösung als solche. Das könnte man ja durchaus als ein größeres Projekt im Wasserfallrahmen irgendwie realisieren, wenngleich ich aber insbesondere die Anwendung und dann später die Realisierung von Projekten, von echten Projekten mittels Low Code-Lösungen, dann doch eher agil einführen würde. Also wenn ich dafür die Entscheidungsbefugnis hätte, würde ich mich eher auf ein agiles Modell orientieren.

EB06\_Hesse, Moritz: 41 - 41 (0)

Und was, ich glaube, auch wieder so ein bisschen, die kleine Kfz-Werkstatt oder der kleine Dienstleister würde sich, glaube ich, immer eher das agile Verfahren wählen, wohingegen der Konzern eher zu einem Prozess tendieren würde, der vielleicht ein bisschen langfristiger, langwieriger ist, aber dafür wiederum besser auditierbar, besser die Compliance-Richtlinien erfüllt. Von daher fällt mir das auch schwer die Entscheidung, ob ich agil oder Wasserfall mache oder klassisch. Weiß ich nicht, ob ich das auch wieder von dem Faktor Low Code oder kein Low Code abhängig machen würde. Ich glaube, die Variablen wären da wieder andere, Unternehmensgröße, Unternehmensorganisationsform etc. pp.

EB06\_Hesse, Moritz: 43 - 43 (0)

Aber in jedem Fall wären bei Ihnen, also selbst beim großen Unternehmen, würden Sie selbst, wenn Sie Wasserfall-Anteile mit reinnehmen, primär immer den agilen Anteil bevorzugen?

**B: MH** Ja, aufgrund der Agilität der Lösung und jeder Low Code- Lösung, also ich kenne durchaus nicht alle, aber ich würde behaupten, die bringen alle eine gewisse Leichtigkeit, Agilität mit sich. Das ist ja der Grund, warum man die wählt und deshalb würde ich, allein aus so einer Perspektive des Mindsets einfach argumentieren wollen. Dass es sich beißen würde, einfach so, der eine ist eher so dieser Langsame, Sichere, ich will erst die Unterschrift haben und der andere ist eher so, ja aber guck mal hier, ich kann doch in 3 Minuten die Lösung zusammenklicken, hier haben wir es doch schon. Ich könnte mir vorstellen, dass sich das auf so einer kulturellen Ebene einfach beißt, auf einer organisatorisch-kulturellen Ebene.

EB06\_Hesse, Moritz: 44 - 45 (0)

Nein, es ist im Endeffekt eher im agilen Bereich anzusiedeln. Es geht jetzt nicht darum, dass man einen starren Projektplan hat, sondern es geht wirklich darum, dass man in kurzen Abschnitten etwas schafft. Es geht darum, dass eine Idee aufgegriffen wird. Dass eine Idee relativ schnell auch gezeigt werden kann, durch jemanden, der sich dann mit diesem Low Code/ No Code- Programm auseinandersetzt, der sagen kann, so könnte die Oberfläche aussehen, so könnten die Schnittstellen aussehen. Das ist alles in diesem Bereich drin. Es geht halt auch in vielerlei Hinsicht in den Bereich der User-Stories hinein. Also auch agile Elemente, die ja eine große Rolle spielen, sind da drin. Und es hängt eher damit zusammen, dass man sagt: "Ich habe eine Idee, ich habe die Information durch die Fachabteilung, ich habe mich ausgetauscht, ich habe das nach ganz gewissen Prinzipien festgestellt." Also Workshops wären zum Beispiel eine ganz wichtige Sache. Aber es gehören auch von dem Citizen Developer ohne weiteres dann Fachinterviews dazu. Natürlich dann auf einem sehr hohen fachlichen Niveau, weil, die kommen ja aus dem Fachbereich und der Fragende wie der Befragte kennen sich sehr, sehr gut aus

darin. Und dass man dann sagen kann: In dieser und jener Form wollen wir es realisieren, dann wird auch relativ schnell gezeigt, dann werden erst Prototypen davon gemacht. Also das ist sehr, sehr stark agil und da wollen wir ein sehr, sehr schnelles Feedback.

EB08\_Fey, Peter: 27 - 27 (0)

5 - Wahl Vorgehensmodell\5.2 - Hybrides Vorgehensmodell

Der Experte hebt hervor, dass durch den agilen Charakter der Plattformen ein traditionelles Vorgehen sich nicht eignet. Die hohe Veränderungsgeschwindigkeit, in der sich die Plattformen selbst verändern, erschwert eine langfristige Planung und es ist nötig, schnelle Feedbackschleifen zu haben. Es wird klar ein agiler Schwerpunkt bei der Einführung favorisiert, allerdings wird es Vorteilhaft gesehen, zu Beginn des Projektes in Phasen zu arbeiten, um eine Governance aufzubauen und Enabling-Konzepte zu etablieren. Diese Phasen sollten jedoch nur einen kurzen Zeitraum einnehmen. Es wird also eher ein hybrides Vorgehen favorisiert.

EB02\_Rosengrün, Robin

Die Expertin weist darauf hin, dass sie zwar einen agilen Ansatz bevorzugt, aber das von ihr favorisierte Vorgehensmodell trotzdem eine Mischform ist. Dies begründet sich einerseits darin, dass vor allem in großen Unternehmen noch die Kennzahlen On Time, On Scope, On Budget erwartet werden. Im Weiteren ist Dokumentation ein großes Thema in ihrer Ausführung. Dabei geht sie auf Statusberichte ein, aber erklärt auch, dass der Vorteil bei einem Schwerpunkt auf agilem Vorgehen ist, dass es nicht zu einer Einführung kommt, bei der auf einmal alles freigegeben wird, sondern es zu einer schrittweisen Freigabe über Sprints kommt.

EB04\_Götl, Simone

Der Experte vertritt die Auffassung, dass ein hybrides Modell mit Schwerpunkt auf dem traditionellen Projektmanagement das wohl beste Modell für die Einführung ist. Dies begründet er dadurch, dass die meisten

		<p>Unternehmen noch unerfahren mit agilen Methodiken und IT-Projekten gibt. Ein weiterer Grund, warum der Schwerpunkt des gewählten Ansatzes auf den traditionellen Ansätzen liegt, ist, dass eine bessere Planbarkeit existiert und ein Rahmen für das Projekt geschaffen wird.</p> <p>Vorteile, die aus dem agilen übernommen werden, sind Meetings wie Daily, Weekly oder Retrospektiven.</p> <p>EB05_Rohrbeck, Christian</p>
		<p>Der Experte weist darauf hin, dass er eine agile Einführung bevorzugt, aber es die Möglichkeit gibt, diese in einen Wasserfallrahmen einzubetten, um die Planungsaspekte umzusetzen. Die aktive Implementierung sollte immer agil erfolgen.</p> <p>EB06_Hesse, Moritz</p>
		<p>Der Experte vertritt die Meinung, dass sich die Wahl des Vorgehensmodells daran orientieren sollte, welches Vorgehensmodell die Organisation kennt und wo die Erfahrungen liegen. Es wird als nicht zweckdienlich empfunden, der Organisation ein Modell aufzuzwingen, welches sie nicht beherrscht. Es wird befürchtet, dass sonst ein erhöhtes Maß an Unsicherheit herrschen könnte. Einerseits wegen der neuen Plattform und andererseits wegen des neuen Vorgehensmodells. Jedoch würde der Experte aus seiner Erfahrung einen hybriden Ansatz bevorzugen.</p> <p>EB07_Janke, Tino</p>
<p>5 - Wahl Vorgehensmodell\5.2 - Hybrides Vorgehensmodell\5.2.1 - Ausprägung Hybrides Modell</p>	<p>Also, da ich es auch ein Projekt sehe, was immer am Laufen ist, hat halt keinen klaren Abschluss. Also, eine Low Code-Plattform an sich, ich sag nicht irgendwann, die ist eingeführt also, sondern es gibt dann immer neue Herausforderungen. Es gibt neue Funktionalitäten, die man vielleicht einführen/ schmackhaft machen will. Man hat also gerade speziell auch in der Microsoft Power Plattform, aber ich glaube auch in allen Low Code-Plattformen, hat man sehr hohe Entwicklungs-, also, die Plattform selber hat noch eine hohe Veränderungsgeschwindigkeit, wie die sich verändert. Von</p>	

daher, denke ich, kommt man um einen agilen Ansatz da gar nicht drumherum, weil, Microsoft schmeißt dann einfach noch mal irgendwo einen Stock in die Speichen irgendwas ändert sich und dann musst du schnell darauf reagieren, also da kann ich nicht klassisch in 3 Phasen das durchziehen, denke ich.

EB02\_Rosengrün, Robin: 39 - 39 (0)

Ist natürlich bei dieser initialen Einführung, da müsste man schon mal in den Phasen arbeiten, denke ich, eine Governance aufbauen, Enabling Konzepte aufbauen, aber das würde ich auch nicht, also da würde ich bitte kein Projekt über 1 Jahr draus machen, sondern würde das dann eben...

EB02\_Rosengrün, Robin: 41 - 41 (0)

So, aber wenn wir jetzt von einem reinen Einführungsprojekt sprechen, von einer Software, dann ist das Ziel ja relativ definiert. Ich kann mir ja überlegen, welche Features und Module ich als Erstes löse und deshalb wäre für mich das dann schon agil, aber es gibt halt Elemente darin, die einfach auch in gewisser Weise wahrscheinlich nicht agil sind. Das fängt bei irgendwelchen Statusberichten an. Das fängt damit an, wer hat jetzt Einfluss auf so ein Projekt? Ja und wie werden auch Entscheidungen getroffen letztlich? Also ich glaube, daher ist es wahrscheinlich schon eine Mischform, aber ich würde die agilen Elemente nehmen, weil das einfach auch dazu führt, dass ich auch peu à peu einzelne Features und Module einführen kann und nicht alles als Big Bang machen muss, das heißt, der Mitarbeiter, der Endnutzer, kann es schneller nutzen.

EB04\_Göttl, Simone: 35 - 35 (0)

**I: RS** Nur für mich noch mal zum Verständnis. Was würden Sie, wenn Sie sagen, es ist eine Mischform meistens, aber ganz klarer Schwerpunkt liegt auf agil, aber was würden Sie sagen, abgesehen von Dokumentationsfragen, würden Sie sagen, es gibt noch andere Sachen aus dem traditionellen Projektmanagement, die da häufig mit

vorkommen? Gibt es da Sachen, die sich wiederholen, die wiederholend vorkommen?

**B: SG** Was man auch noch sieht. Also man muss ja jetzt überlegen, ob man einen Projektstatusbericht macht, wenn man zum Beispiel in Tools wie Jira ein Backlog pflegt für so ein Einführungsprojekt, dann steckt ja dort die Wahrheit drin und ich kann mir dort anzeigen lassen, an welchen Punkten welche Sachen im Backlog sind, welche Sachen im aktuellen Sprintbacklog sind und krieg ja auch dann über gewisse Kennzahlen, die so ein Scrum-Team hat, wie viele Feature Points, habe ich jetzt zum Beispiel im letzten Sprint geschafft und so weiter, kriege ich ja Input darüber, wie es läuft. Aber was man bei großen Unternehmen mit viel Geld sehen, das ist noch klassisch verankert, da will irgendjemand, der dann alle Projekte zusammen bündelt, eine Ampel sehen. Da ist natürlich die Frage, also Statusampeln, ist das jetzt agil oder nicht? Da kann man immer drüber streiten, aber da werden andere Kennziffern gemacht. Da wird auch erwartet, dass ein Projektfortschritt gemessen wird, in 85 % ist fertig und solche Sachen. Also das sieht man schon noch, dass dann halt einfach alte Strukturen dazu führen, dass man sich alter Elemente, gerade von Statusberichten, und so was bedient.

EB04\_Göttl, Simone: 36 - 37 (0)

**I: RS** Aber in jedem Fall wären bei Ihnen, also selbst beim großen Unternehmen, würden Sie selbst, wenn Sie Wasserfall-Anteile mit reinnehmen, primär immer den agilen Anteil bevorzugen?

**B: MH** Ja, aufgrund der Agilität der Lösung und jeder Low Code- Lösung, also ich kenne durchaus nicht alle, aber ich würde behaupten, die bringen alle eine gewisse Leichtigkeit, Agilität mit sich. Das ist ja der Grund, warum man die wählt und deshalb würde ich, allein aus so einer Perspektive des Mindsets einfach argumentieren wollen. Dass es sich beißen würde, einfach so, der eine ist

eher so dieser Langsame, Sichere, ich will erst die Unterschrift haben und der andere ist eher so, ja aber guck mal hier, ich kann doch in 3 Minuten die Lösung zusammenklicken, hier haben wir es doch schon. Ich könnte mir vorstellen, dass sich das auf so einer kulturellen Ebene einfach beißt, auf einer organisatorisch-kulturellen Ebene.

EB06\_Hesse, Moritz: 44 - 45 (0)

5 - Wahl Vorgehensmodell\5.2 - Hybrides Vorgehensmodell\5.2.2 - Begründung hybrides Vorgehen

Das kann man nicht, oder ich würde eher jetzt spontan zu einer hybriden Lösung tendieren. Aus meiner Sicht ist es so, dass eine rein agile Lösung im klassischen Sinne, ein reines Modell, unvermischt, laut Lehrbuch, findet man in den meisten Projekten nicht. Es wäre maximal in so einer projektorientierten Umgebung möglich, aus meiner Sicht, wo wir wirklich Experten haben, die auch in diesem Modus schon zusammengearbeitet haben. Die also genau wissen, was sie zu tun haben. Wo ihr Fokus darauf ist und noch in der Lage sind, selbsttempowert, sage ich mal, Entscheidungen zu treffen und, zum Beispiel, jetzt die Sprintziele festzulegen und das Ganze dann auch halt selbst motiviert voranzutreiben.

EB05\_Rohrbeck, Christian: 39 - 39 (0)

Wir haben es jetzt hier aber, glaube ich, in einer No Code/ Low Code-Einführung eher mit einem Umfeld zu tun, wo eine gewisse Unerfahrenheit in IT-Projekten besteht. Also insgesamt in der Projektarbeit, weil man davon ausgehen muss, dass die Kollegen, die das machen, dann tatsächlich diesem Daily Business verhaftet sind, ihren täglichen Routinen. Also da fehlt die Projekterfahrung, deswegen wäre es eher ein Waterfall-Ansatz. Dass es jemanden gibt, der so ein bisschen den Rahmen festlegt und die nächsten Schritte, aber und das kommt dazu vom hybrid her, durchaus hybride Teile einstreut. Das ist, glaube ich, das was ich sehe, was sich bewährt. Gerade, wenn da keine Projektkultur sich schon



entfaltet hat. Agile Elemente, die wirklich sich sehr bewährt haben, sind zum Beispiel, ein Daily zu machen, ein Daily Stand-up - 15 Minuten am Anfang des Tages und auch Retrospektiven sowie auch eine graphische Übersicht der Todos an einem Kanban-Board. Das sind Elemente aus dem agilen Leben, die man wunderbar in eine Waterfall-Logik einstreuen könnte. Das wäre aus meiner Erfahrung heraus, hierfür glaube ich ein machbarer, praktikabler Ansatz.

**I: RS** Also würdest du sagen, dass, wenn du dem jetzt ein Gewicht geben müsstest, es schon einen Schwerpunkt auf dem Waterfall gibt innerhalb des hybriden Modells, aber agile Tools in der Realisierungsphase, vermute ich mal, meinst du, dann zur Anwendung kommen? Also dort ein iterativer Gedanke mit eingestreut wird.

**B: CR** Ja genau, das würde ich so sagen und Hintergrund ist einfach, dass sich da noch keine Projektkultur entfaltet hat bzw. nicht gelebt wird und dass die Kollegen da erst abgeholt werden müssen.

EB05\_Rohrbeck, Christian: 39 - 41 (0)

Und was, ich glaube, auch wieder so ein bisschen, die kleine Kfz-Werkstatt oder der kleine Dienstleister würde sich, glaube ich, immer eher das agile Verfahren wählen, wohingegen der Konzern eher zu einem Prozess tendieren würde, der vielleicht ein bisschen langfristiger, langwieriger ist, aber dafür wiederum besser auditierbar, besser die Compliance-Richtlinien erfüllt. Von daher fällt mir das auch schwer die Entscheidung, ob ich agil oder Wasserfall mache oder klassisch. Weiß ich nicht, ob ich das auch wieder von dem Faktor Low Code oder kein Low Code abhängig machen würde. Ich glaube, die Variablen wären da wieder andere, Unternehmensgröße, Unternehmensorganisationsform etc. pp.

EB06\_Hesse, Moritz: 43 - 43 (0)

Wo die Organisation, die das Thema einführen möchte, beziehungsweise beleuchten möchte, wo die herkommt.

Wenn sie zum Beispiel bisher sehr klassischen Projektansatz hat, dass dort die Projekte immer mit Wasserfall zum Beispiel bearbeitet werden, mit Wasserfallmethoden, dann macht das, glaub ich, keinen Sinn, in diesen Unternehmen das neue Thema Low Code/ No Code und Citizen Development in einem agilen Vorgehen einzuführen, weil es einfach zu einer doppelten Unsicherheit führt. Also ich würde in der Regel das Vorgehen nehmen, was ich in der Organisation finde. Wobei ich denke, dass wenn man die Wahl hat, glaube ich, ein hybrider Ansatz eine ganz gute Lösung ist.

EB07\_Janke, Tino: 31 - 31 (0)

6 - Vorteile gewähltes Vorgehensmodell\6.1 - Vorteile agiles Vorgehen

Es gibt ein paar Dinge die ich Anfangs in der Zielsetzung machen würde, die vielleicht so ein bisschen nicht hundertprozentig Agil sind und da ist wieder mein Lieblingsthema Governance da, das sollte man am Anfang entscheiden wie viel Risiken gehe ich damit ein und wie kann ich die Risiken streuen. Da muss man man tatsächlich ganz klar am Anfang mit IT und Operations und was weiß ich, Security Officer, wer auch immer im Unternehmen dafür zuständig ist, sich wirklich mal klar machen, welche Datenschutzanforderungen haben wir? Mit welchen Daten gehen wir um? Manchmal führt das zu einem Vorlauf, in so einem Projekt, das ich erstmal mit Datenklassifikationen und so weiter anfangen muss zu arbeiten. Das ich sicherstelle das bestimmte Daten da nicht rauskommen. Aber das kann man auch Agil nach und nach machen in dem Mann, wenn man das Risiko kennt, tatsächlich guckt wo setzen wir das ein? Wem gebe ich Zugriff auf die Plattform und so weiter? Das kann man schon noch steuern.

EB01\_Eckhard Hauenherm: 43 - 43 (0)

Also, die Vorteile vom agilen sind, ich habe es schon ein bisschen so erwähnt, tatsächlich die Tatsache, dass ich relativ schnell Feedback kriege über das, was ich mache. Ich arbeite ja sozusagen in so einem Projekt typischerweise nicht einfach so, dass ich erst die Plattform einführe und dann mit Anwendern

Der Experte vertritt die Auffassung, dass auch Planungsbestandteile in Projekten, für zum Beispiel Governance, agil durchgeführt werden können.

EB01\_Eckhard Hauenherm

arbeite, sondern ich arbeite mit einigen Anwendern daran, die Plattform einzuführen und zu gucken, wie kann man die nutzen in ihrem Arbeitsbereich. Dann habe ich ein schnelles Feedback und kann damit wieder überlegen, was muss ich in der Konfiguration der Plattform machen für die nächste Gruppe von Anwendern. Also diese kurzen Feedbackschleifen sind ein extremer Vorteil in solchen Projekten. Zweitens kann ich sehr viel schneller reagieren, zum Beispiel, auf Änderungen, die vielleicht in der Plattform auch erfolgen. Ich habe eben schon gesagt, die Plattformen selber entwickeln sich agil weiter. Plötzlich kann ich etwas machen, was ich vor einem Monat noch nicht machen konnte. Kann ich sofort überlegen: "Sollen wir nicht mal versuchen, das umzusetzen. Hier, jetzt suchen wir mal 3 Leute. Machen mit denen mal 2 Wochen Sprint und versuchen das mal umzusetzen in irgendeiner Applikation, ob das was für euch wäre. Hört sich gut an, könntet ihr nutzen." So was kann ich machen. Die Schnelligkeit, auch Änderungen einarbeiten zu können, ist ein sehr starker Vorteil des agilen Ansatzes und damit natürlich auch verbunden, ohne große upfront Investitionen schnell wieder aus dem Projekt auszusteigen, wenn man feststellt: "Nee, das ist nichts für uns. Unsere Anwender ziehen nicht mit." Oder "Die Risiken sind uns zu groß." Oder "Die Plattform kann nicht das, was wir brauchen." Das sehe ich relativ schnell mit einem agilen Vorgehen. Das würde ich bei einem plangetriebenen Ansatz oder auch bei einem hybriden Ansatz habe ich es häufig so, dass ich sehr viele Vorinvestitionen habe, um erst mal einen Plan aufzubauen, um erst mal die Plattform zu checken, um erst mal die Plattform auszuwählen und so weiter und so fort. Das wird deutlich teurer und führt auch eher zu Frustration. Der motivatorische Faktor bei agilem Vorgehen spielt auch noch eine Rolle. Mit einer kleinen Gruppe von Anwendern schnell mal eben was umzusetzen, passt eigentlich wunderbar zu Low Code/ No Code. Also da würde ich immer sehr

stark dazu tendieren, das ist agil. PMI nennt das ja sogar hyber agil.  
EB01\_Eckhard Hauenherm: 45 - 45 (0)

Ich bin grundsätzlich ein Freund von agilen Vorgehensweisen. Das ist aber weniger meiner Erfahrung aus der Low Code-Ecke geschuldet, sondern das ist einfach eine sehr persönliche Ansicht, aus meiner Erfahrung als Softwareentwickler, mit den Vorteilen oder auch Nachteilen, die die agile Softwareentwicklung oder das agile Projektmanagement mit sich bringen. Ich glaube, bei der Einführung und der nachhaltigen Nutzung von Low Code-Lösungen tun sich Unternehmen keinen Gefallen, an einem starren Wasserfallmodell festzuhalten. Lassen Sie mich ausführen, warum. Eine wesentliche Stärke von Low Code-Lösungen zeichnet sich durch diese Flexibilität und Schnelligkeit aus. Diese Schnelligkeit ermöglicht den Kunden, zum Beispiel, sehr, sehr schnell auf regulatorische Anforderungen reagieren zu können. Das klassische Wasserfallmodell, das klassische Projektmanagementmodell, hat genau diese Schwäche, dass es nicht flexibel und schnell ist. Sondern im Gegenteil, es baut sich genau an diesen Projektphasen auf, wo die nachgelagerten Phasen immer erst den Abschluss vorgelegerten Phasen erfordern und genau das sind ja die Vorteile der agilen Softwareentwicklung. Dass man quasi durchaus auch diese ganz klassischen Phasen hat. Planung, Durchführung, Testing, Release und die Problembhebung, aber die Zyklen sind einfach viel kleiner. Man hat ja quasi diese klassischen Sprints, im Scrum-Verfahren, im Scrum-Modell, und die ermöglichen einfach ein viel schnelleres reagieren auf einen Wechsel der Anforderungen und ich habe selten eine so hohe Volatilität der Anforderungen im Verlauf der Implementierung erfahren, wie ich das mit Kunden, dieser Lösung hier, erfahre. Einfach, das bringt das Produkt halt mit sich. Das sind Kunden, die wollen ja Flexibilität, die wollen Schnelligkeit und es ist selten so, dass ich am Ende des Projekts die Anforderung noch so habe, wie sie ursprünglich genannt

Der Experte hebt neben der Möglichkeit, flexibel und schnell auf geänderte Anforderungen reagieren zu können, Parallelisierung von Aufgaben vornehmen zu können, hervor. Dies kann dazu beitragen, Bottlenecks zu vermeiden.

EB06\_Hesse, Moritz

wurden. Und insofern tut sich ein Unternehmen bei der Einführung von No Code/ Low Code- Lösungen glaube ich keinen großen Gefallen, wenn es zu starr an den alten Strukturen, an alten Modellen, festhält und würde immer eher die agilen Modelle, quasi empfehlen, wenn gleich man hier auch noch einmal unterscheiden kann zwischen der Einführung der Lösung als solche. Das könnte man ja durchaus als ein größeres Projekt im Wasserfallrahmen irgendwie realisieren, wenngleich ich aber insbesondere die Anwendung und dann später die Realisierung von Projekten, von echten Projekten mittels Low Code-Lösungen, dann doch eher agil einführen würde. Also wenn ich dafür die Entscheidungsbefugnis hätte, würde ich mich eher auf ein agiles Modell orientieren.

EB06\_Hesse, Moritz: 41 - 41 (0)

6 - Vorteile gewähltes Vorgehensmodell\6.1 - Vorteile agiles Vorgehen\6.1.1 - Vorteil agil schnelles Feedback

Das passt dann gut zusammen und man muss ja auch mit dem Produkt dann lernen. Das ist ja genau das, ich habe mit einem iterativen Projekt meine Feedbackschleife und kann dann sehr schön mit der Plattform lernen, was uns die im Unternehmen bringt oder nicht bringt.

EB01\_Eckhard Hauenherm: 41 - 41 (0)

Also, die Vorteile vom agilen sind, ich habe es schon ein bisschen so erwähnt, tatsächlich die Tatsache, dass ich relativ schnell Feedback kriege über das, was ich mache. Ich arbeite ja sozusagen in so einem Projekt typischerweise nicht einfach so, dass ich erst die Plattform einführe und dann mit Anwendern arbeite, sondern ich arbeite mit einigen Anwendern daran, die Plattform einzuführen und zu gucken, wie kann man die nutzen in ihrem Arbeitsbereich. Dann habe ich ein schnelles Feedback und kann damit wieder überlegen, was muss ich in der Konfiguration der Plattform machen für die nächste Gruppe von Anwendern. Also diese kurzen Feedbackschleifen sind ein extremer Vorteil in solchen Projekten.

EB01\_Eckhard Hauenherm: 45 - 45 (0)

Als Vorteil wird gesehen, dass es möglich ist, durch den iterativen Charakter ein schnelles Feedback zu erhalten. Mit jeder Iteration und der Präsentation eines MVP kann Feedback eingeholt werden, was dazu dient, dieses Feedback in die nächste Iteration mit einfließen zu lassen. Es muss nicht erst die Plattform komplett eingeführt werden, um Feedback zu erhalten.

EB01\_Eckhard Hauenherm

	<p>Also rein agil macht dann Sinn, auch wenn ich wirklich das Ziel noch nicht konkret vor Augen habe und sage, ich brauche eine App oder eine Anwendung für Problem XY und dann immer wieder auch häppchenweise an den Kunden, also wie ein Start-up, an den Kunden eine App oder eine Anwendung hinausgeht.</p> <p>EB04_Göttl, Simone: 35 - 35 (0)</p>	<p>Ein Vorteil, der durch die Expertin hervorgehoben wird, ist, dass bei einem Vorgehen, wo die Zielstellung nicht klar ist, eine Entwicklung auf Basis von kontinuierlichem Feedback erfolgen kann.</p> <p>EB04_Göttl, Simone</p>
	<p>Auf der anderen Seite würde ich von der Seite der agilen Methoden, kann ich mir vorstellen, dass man durchaus auch in Sprints arbeitet, also wenn man zum Beispiel darüber spricht, dass man in einer iterativen Vorgehensweise einzelne Fachbereiche in dieses Projekt hinzuholt und die darin schult. Das wäre eine Sache, wo man iterativ vorgehen könnte, einfach damit man die Gelegenheit hat, aus den Erfahrungen des einen Onboarding des Fachbereiches für den nächsten zu lernen.</p> <p>EB07_Janke, Tino: 35 - 35 (0)</p>	<p>Der Experte hebt die Möglichkeit hervor, dass aufgrund der iterativen Vorgehensweise die Erfahrungen vorangegangener Iterationen in die nächste mit einbezogen werden können.</p> <p>EB07_Janke, Tino</p>
	<p>Es geht jetzt nicht darum, dass man einen starren Projektplan hat, sondern es geht wirklich darum, dass man in kurzen Abschnitten etwas schafft. Es geht darum, dass eine Idee aufgegriffen wird. Dass eine Idee relativ schnell auch gezeigt werden kann, durch jemanden, der sich dann mit diesem Low Code/ No Code- Programm auseinandersetzt, der sagen kann, so könnte die Oberfläche aussehen, so könnten die Schnittstellen aussehen. Das ist alles in diesem Bereich drin.</p> <p>EB08_Fey, Peter: 27 - 27 (0)</p> <p>Und dass man dann sagen kann: In dieser und jener Form wollen wir es realisieren, dann wird auch relativ schnell gezeigt, dann werden erst Prototypen davon gemacht. Also das ist sehr, sehr stark agil und da wollen wir ein sehr, sehr schnelles Feedback.</p> <p>EB08_Fey, Peter: 27 - 27 (0)</p>	<p>Der Experte hebt den Vorteil hervor, dass ein schnelles Feedback erfolgt auf eine Anforderung und erste Ideen evaluiert werden können, wie diese Anforderungen umgesetzt werden können. Nach der Umsetzung kann in einem zweiten Schritt schnell ein Feedback zur Lösung eingeholt werden.</p> <p>EB08_Fey, Peter</p>
<p>6 - Vorteile gewähltes Vorgehensmodell\6.1 - Vorteile agiles</p>	<p>Das habe ich hier nicht, sondern es ist eine Einführung einer Plattform, die sich dann sozusagen immer weiterentwickelt. Und die Plattformen selber werden ja bei</p>	<p>Als Vorteil wird wahrgenommen, dass durch ein schnelles Feedback eine hohe Reaktionsgeschwindigkeit erzeugt wird auf geänderte Anforderungen.</p>

Vorgehen\6.1.2 -  
Vorteil agil Reakti-  
onsgeschwindigkeit

den Anbietern auch agil entwickelt. Also so eine Power Plattform ist ja auch nicht fertig und ich führe die ein, sondern da kommen jeden Monat neue Funktionalitäten rein. Das heißt, ich kann das eigentlich nur agil machen und immer gucken, in kleinen Zielen: "Wo gucken wir jetzt? Was machen wir als nächstens? Was nehmen wir uns für die nächsten 4 Wochen vor oder 2 Wochen?" Also klassisches Scrum-Projekt oder klassisches agiles Projekt und das passt auch am besten zu den Plattformen selber.

EB01\_Eckhard Hauenherm: 41 - 41 (0)

Und ich kann auch mit relativ wenig Kosten schnell wieder aussteigen, wenn ich sage: "Nee, das ist für uns nichts." Oder "Die Plattform ist für uns nichts. Wir nehmen mal eine andere." Das kann ich bei einem fertig geplanten, also bei einem klassischen, sequenziellen Projekt habe ich das nicht. Da habe ich am Ende alles eingeführt und keiner kann damit arbeiten. Das wollen wir nicht.

EB01\_Eckhard Hauenherm: 41 - 41 (0)

Zweitens kann ich sehr viel schneller reagieren, zum Beispiel, auf Änderungen, die vielleicht in der Plattform auch erfolgen. Ich habe eben schon gesagt, die Plattformen selber entwickeln sich agil weiter. Plötzlich kann ich etwas machen, was ich vor einem Monat noch nicht machen konnte. Kann ich sofort überlegen: "Sollen wir nicht mal versuchen, das umzusetzen. Hier, jetzt suchen wir mal 3 Leute. Machen mit denen mal 2 Wochen Sprint und versuchen das mal umzusetzen in irgendeiner Applikation, ob das was für euch wäre. Hört sich gut an, könntet ihr nutzen." So was kann ich machen. Die Schnelligkeit, auch Änderungen einarbeiten zu können, ist ein sehr starker Vorteil des agilen Ansatzes und damit natürlich auch verbunden, ohne große upfront Investitionen schnell wieder aus dem Projekt auszusteigen, wenn man feststellt: "Nee, das ist nichts für uns. Unsere Anwender ziehen nicht mit." Oder "Die Risiken sind uns zu groß." Oder "Die Plattform kann nicht das, was

Diese geänderten Anforderungen ergeben sich im Weiteren auch durch Änderungen an der Plattform selbst, da diese einer ständigen Weiterentwicklung durch die Plattformbetreiber unterliegen. Diese Reaktionsgeschwindigkeit ermöglicht ein schnelles Testing von neuen Features und Anforderungen sowie eine Prüfung, wie diese eventuell im Unternehmenskontext eingesetzt werden.

Ein weiterer Vorteil, der sich aus einem schnellen Reagieren ergibt, ist die Möglichkeit einer kostengünstigen Exit-Strategie. Es wird frühzeitig festgestellt, wenn eine Plattformsentscheidung nicht den gewünschten Mehrwert gebracht hat und somit kann in einem frühen Stadium festgestellt werden, ob die Plattform gewechselt werden soll oder die Implementierung abgebrochen wird und eine neue Evaluation der Situation nötig ist.

EB01\_Eckhard Hauenherm

wir brauchen." Das sehe ich relativ schnell mit einem agilen Vorgehen.  
EB01\_Eckhard Hauenherm: 45 - 45 (0)

Also, da ich es auch ein Projekt sehe, was immer am Laufen ist, hat halt keinen klaren Abschluss. Also, eine Low Code-Plattform an sich, ich sag nicht irgendwann, die ist eingeführt also, sondern es gibt dann immer neue Herausforderungen. Es gibt neue Funktionalitäten, die man vielleicht einführen/ schmackhaft machen will. Man hat also gerade speziell auch in der Microsoft Power Platform, aber ich glaube auch in allen Low Code-Plattformen, hat man sehr hohe Entwicklungs-, also, die Plattform selber hat noch eine hohe Veränderungsgeschwindigkeit, wie die sich verändert. Von daher, denke ich, kommt man um einen agilen Ansatz da gar nicht drumherum  
EB02\_Rosengrün, Robin: 39 - 39 (0)

Es wird durch den Experten hervorgehoben, dass die Plattformen sich selbst verändern und agile Projektmanagement-Strukturen etabliert, diese Änderungen während des Einführungsprojektes mit umzusetzen. Ein Vorteil, der in der Folge sich daraus generiert, ist, dass für den Betrieb der Plattform diese Strukturen zum Umgang mit den Veränderungen weiter genutzt werden können, da sie bereits gebildet wurden.  
EB02\_Rosengrün, Robin

Weil die Bestandteile zum einen recht unabhängig voneinander sind. Also das Thema Changemanagement und Plattform enablement, zum Beispiel, kann völlig parallel laufen. Es ist von unterschiedlichen Stakeholdern und Mitarbeitern betroffen, die auch mit unterschiedlichen Geschwindigkeiten unterwegs sind.  
EB03\_Fabian Gackstatter: 37 - 37 (0)

Der Experte hebt die Möglichkeit der Parallelisierung von Arbeitspaketen im agilen Vorgehen hervor. Das ermöglicht eine sehr hohe Reaktionsgeschwindigkeit, ohne dabei auf andere Tasks der Einführung Rücksicht nehmen zu müssen.  
EB03\_Fabian Gackstatter

Also die agile Methode bietet sich einfach hier eher an, weil ich schneller zu meinen Arbeitspaketen komme und die auch unabhängiger von den anderen Bereichen entwickeln kann.  
EB03\_Fabian Gackstatter: 37 - 37 (0)

Also rein agil macht dann Sinn, auch wenn ich wirklich das Ziel noch nicht konkret vor Augen habe und sage, ich brauche eine App oder eine Anwendung für Problem XY und dann immer wieder auch häppchenweise an den Kunden, also wie ein Start-up, an den Kunden eine App oder eine Anwendung hinausgeht.  
EB04\_Görtl, Simone: 35 - 35 (0)

Die Möglichkeit eines schnellen Feedbacks ermöglicht auch ein schnelles Reagieren auf neue Anforderungen.  
EB04\_Görtl, Simone

die Vorteile der agilen Softwareentwicklung. Dass man quasi durchaus auch diese ganz klassischen Phasen hat.

Der Experte sieht es als vorteilhaft an, dass durch den iterativen Charakter von agilen Vorgehensweisen, ein



Planung, Durchführung, Testing, Release und die Problembhebung, aber die Zyklen sind einfach viel kleiner. Man hat ja quasi diese klassischen Sprints, im Scrum-Verfahren, im Scrum-Modell, und die ermöglichen einfach ein viel schnelleres Reagieren auf einen Wechsel der Anforderungen und ich habe selten eine so hohe Volatilität der Anforderungen im Verlauf der Implementierung erfahren, wie ich das mit Kunden, dieser Lösung hier, erfahre. Einfach, das bringt das Produkt halt mit sich. Das sind Kunden, die wollen ja Flexibilität, die wollen Schnelligkeit und es ist selten so, dass ich am Ende des Projekts die Anforderung noch so habe, wie sie ursprünglich genannt wurden.

EB06\_Hesse, Moritz: 41 - 41 (0)

schnelles Reagieren auf geänderte Anforderungen möglich ist. Vor allem in Zusammenhang mit NCLC-Plattformen wird dies als Vorteil wahrgenommen. Es ist ein hohes Maß an Flexibilität und Schnelligkeit möglich, da sich Anforderungen bei diesen Projekten fast immer ändern im Projektverlauf.

EB06\_Hesse, Moritz

Ich würde bei den agilen Methoden auch Wert darauf legen, schnell Erfolge sichtbar zu machen, dass man also nicht mit einem beispielsweise Big Bang arbeitet und alles passiert im Kämmerchen bis das ganze Thema leicht geht, sondern man muss eben zwischendrin auch die Erfolge sichtbar machen. Also immer ein Stückchen mehr, ja, Inkrement sozusagen, ein Stückchen mehr zeigen, was man geschaffen hat, um die Projektmitglieder einfach am Ball zu behalten und auch den Stakeholdern zu zeigen, dass man hier konsequent Mehrwert schafft für die Organisation.

EB07\_Janke, Tino: 35 - 35 (0)

Der Experte sieht als Vorteil, dass bei einem agilen Vorgehen eine iterative Vorgehensweise möglich ist. Dies bedeutet, die Ergebnisse einer jeden Iteration zu präsentieren, neue Anforderungen über Feedback einzuholen und diese direkt in der nächsten Iteration umzusetzen. Dadurch soll der systematische Mehrwert für die Organisation aufgezeigt werden.

EB07\_Janke, Tino

Es geht jetzt nicht darum, dass man einen starren Projektplan hat, sondern es geht wirklich darum, dass man in kurzen Abschnitten etwas schafft. Es geht darum, dass eine Idee aufgegriffen wird. Dass eine Idee relativ schnell auch gezeigt werden kann, durch jemanden, der sich dann mit diesem Low Code/ No Code- Programm auseinandersetzt, der sagen kann, so könnte die Oberfläche aussehen, so könnten die Schnittstellen aussehen. Das ist alles in diesem Bereich drin.

EB08\_Fey, Peter: 27 - 27 (0)

Der Experte hebt hervor, dass durch den iterativen Charakter eine schnelle Anforderungsaufnahme erfolgen kann und es möglich ist, in eine direkte Umsetzung zu starten und im Anschluss einen ersten Prototyp vorstellen zu können.

EB08\_Fey, Peter

	<p>Und dass man dann sagen kann: In dieser und jener Form wollen wir es realisieren, dann wird auch relativ schnell gezeigt, dann werden erst Prototypen davon gemacht. Also das ist sehr, sehr stark agil und da wollen wir ein sehr, sehr schnelles Feedback.</p> <p>EB08_Fey, Peter: 27 - 27 (0)</p>	
<p>6 - Vorteile gewähltes Vorgehensmodell\6.1 - Vorteile agiles Vorgehen\6.1.3 - Vorteil agil Motivation</p>	<p>Der motivatorische Faktor bei agilem Vorgehen spielt auch noch eine Rolle. Mit einer kleinen Gruppe von Anwendern schnell mal eben was umzusetzen, passt eigentlich wunderbar zu Low Code/ No Code. Also da würde ich immer sehr stark dazu tendieren, das ist agil. PMI nennt das ja sogar hyper agil.</p> <p>EB01_Eckhard Hauenherm: 45 - 45 (0)</p>	<p>Der motivierende Faktor für Projektteams, Änderungen schnell umzusetzen, wird durch den Experten hervorgehoben.</p> <p>EB01_Eckhard Hauenherm</p>
	<p>Also ich sage mal, der agile Anteil ist sehr wichtig für die, wie soll ich sagen, für die mentale Zufriedenheit der Kollegen, die es umsetzen. Diese Selbstbestimmung, diese Möglichkeit, Feedback zu geben. Das sind ganz wichtige Punkte, um einfach jetzt ein erfülltes Arbeitsleben zu haben.</p> <p>EB05_Rohrbeck, Christian: 45 - 45 (0)</p>	<p>Der Experte hebt den motivatorischen Faktor als einen Vorteil von agilem Vorgehen hervor. Die Selbstbestimmung und die Möglichkeit, Feedback zu geben, werden als wichtige Punkte wahrgenommen.</p> <p>EB05_Rohrbeck, Christian</p>
	<p>Ich würde bei den agilen Methoden auch Wert darauf legen, schnell Erfolge sichtbar zu machen, dass man also nicht mit einem beispielsweise Big Bang arbeitet und alles passiert im Kämmerchen bis das ganze Thema leicht geht, sondern man muss eben zwischendrin auch die Erfolge sichtbar machen. Also immer ein Stückchen mehr, ja, Inkrement sozusagen, ein Stückchen mehr zeigen, was man geschaffen hat, um die Projektmitglieder einfach am Ball zu behalten und auch den Stakeholdern zu zeigen, dass man hier konsequent Mehrwert schafft für die Organisation.</p> <p>EB07_Janke, Tino: 35 - 35 (0)</p>	<p>Der Experte weist darauf hin, dass die Präsentation regelmäßiger Ergebnisse die Motivation für Weiterentwicklung und Fortschritt des Projektes steigert.</p> <p>EB07_Janke, Tino</p>
<p>6 - Vorteile gewähltes Vorgehensmodell\6.2 - Vorteile traditionelles Vorgehensmodell</p>	<p>Es gibt ein paar Dinge die ich Anfangs in der Zielsetzung machen würde, die vielleicht so ein bisschen nicht hundertprozentig Agil sind und da ist wieder mein Lieblingsthema Governance da, das sollte man am Anfang entscheiden wie</p>	<p>Als Vorteil von traditionellen Ansätzen wird die gute Planbarkeit wahrgenommen und die Steuerung von Risiken. Als Beispiel wird die Planung der Governance, Bestimmung der</p>

viel Risiken gehe ich damit ein und wie kann ich die Risiken streuen. Da muss am man tatsächlich ganz klar am Anfang mit IT und Operations und was weiß ich, Security Officer, wer auch immer im Unternehmen dafür zuständig ist, sich wirklich mal klar machen, welche Daten-schutzanforderungen haben wir? Mit welchen Daten gehen wir um? Manchmal führt das zu einem Vorlauf, in so einem Projekt, das ich erstmal mit Datenklassifikationen und so weiter anfangen muss zu arbeiten. Das ich sicherstelle das bestimmte Daten da nicht rauskommen.

EB01\_Eckhard Hauenherm: 43 - 43 (0)

Datenschutzanforderungen und die Risikosteuerung hervorgehoben.

EB01\_Eckhard Hauenherm

Der Experte hebt hervor, dass Vorteile des traditionellen Ansatzes sich vor allem zu Beginn eines Projektes anbieten, da dort eine Planung von Governance und Enabling-Konzepten erfolgt.

EB02\_Rosengrün, Robin

Also rein agil oder sagst du, du willst auch Elemente aus dem Wasserfallmodell haben? Keine Ahnung, Phasen, weil die einen Vorteil bringen.

**B: FG** Eigentlich nicht. Also, wie gesagt, der einzige Wasserfallanteil ist am Anfang so diese Introworkshops, die wir mit dem Kunden machen, über zwei bis drei Wochen, um einfach die richtigen Basisvoraussetzungen zu schaffen und danach geht es aber in den agilen Modus über, wenn wir definieren: "Ok, das sind jetzt die Applikationen, die wir bauen möchten, als Teil der Einführung. Das sind die Organisationsstrukturen, die wir bauen möchten. Das ist das Tooling, das wir selber anfassen müssen." Und das ist aber ein rein agiles Vorgehen an der Stelle.

EB03\_Fabian Gackstatter: 34 - 35 (0)

Der Experte sieht wenig Vorteile von traditionellen Ansätzen im Zusammenhang mit NCLC-Plattformen. Der einzige Anteil wäre für Befähigungsworkshops zu Beginn des Projektes.

EB03\_Fabian Gackstatter

Der Experte hebt Vorteile von traditionellen Ansätzen im Einführungsprojekt hervor, vor allem im Bereich der Planbarkeit und der konzeptionellen Fragen, die zur Implementierung beantwortet werden sollten. Vor allem in Bezug auf Planbarkeit sieht der Experte hier eine Stärke, um Termine einzuhalten.

EB05\_Rohrbeck, Christian

Der Experte hebt hervor, dass ein Vorteil von traditionellen Ansätzen im Zusammenhang mit dem Einführungsprojekt darin besteht, dass eine stärkere Kontrolle über den Gesamtfortschritt besteht. Die Planung und Risikosteuerung ist bei dieser Methodik wesentlich besser zu realisieren als bei agilen Methodiken.

EB06\_Hesse, Moritz

6 - Vorteile gewähltes Vorgehensmodell\6.2 - Vorteile traditionelles Vorgehensmodell\6.2.1 - Vorteil trad. Planungssicherheit

Ist natürlich bei dieser initialen Einführung, da müsste man schon mal in den Phasen arbeiten, denke ich, eine Governance aufbauen, Enabling Konzepte aufbauen, aber das würde ich auch nicht, also da würde ich bitte kein Projekt über 1 Jahr draus machen, sondern würde das dann eben...

EB02\_Rosengrün, Robin: 41 - 41 (0)

Eigentlich nicht. Also, wie gesagt, der einzige Wasserfallanteil ist am Anfang so diese Introworkshops, die wir mit dem Kunden machen, über zwei bis drei Wochen, um einfach die richtigen Basisvoraussetzungen zu schaffen und danach geht es aber in den agilen Modus über, wenn wir definieren: "Ok, das sind jetzt die Applikationen, die wir bauen möchten, als Teil der Einführung. Das sind die Organisationsstrukturen, die wir bauen möchten. Das ist das Tooling, das wir selber anfassen müssen." Und das ist aber ein rein agiles Vorgehen an der Stelle.

EB03\_Fabian Gackstatter: 35 - 35 (0)

Einfach um Dinge, diese konzeptionellen Probleme relativ frühzeitig erkennen zu können. Als aber auch hier eine Direktive vorzugeben. Wo wollen wir denn hin und was sind unsere Meilensteine? Also diese Definition, damit wir auch wissen, was sind unsere Deliverables in der Zeit X. Das muss ganz klar auf dem Schirm sein und meine Erfahrung ist einfach, auch wenn man agil im Prinzip den Scope immer wieder entscheidet, das große Ganze, dieser Endtermin, das darf nicht aus dem Blick geraten und hier bedarf es schon einer klassischen

Unterstützung von einem Projektmanager an der Stelle.

EB05\_Rohrbeck, Christian: 45 - 45 (0)

Und ich glaube bei so Wasserfallprozessen einfach, natürlich man hat ein bisschen länger Zeit, sich in der Planung Gedanken darüberzumachen, ggf. Seiteneffekte, die durch bestimmte Vorhaben auftreten. Auch so diese übersichtliche Gesamtblick-Brille, diese Berühmte. Die ist sicherlich auch in so einem Wasserfallprojekt eher gegeben, als dass es, ja, man verliert in der Umsetzung bei so kleinen Sprints dann manchmal eher den Überblick, als das bei so Wasserfallverfahren der Fall ist.

EB06\_Hesse, Moritz: 43 - 43 (0)

6 - Vorteile gewähltes Vorgehensmodell\6.3 - Vorteile hybrides Vorgehensmodell

Also ich sage mal, der agile Anteil ist sehr wichtig für die, wie soll ich sagen, für die mentale Zufriedenheit der Kollegen, die es umsetzen. Diese Selbstbestimmung, diese Möglichkeit, Feedback zu geben. Das sind ganz wichtige Punkte, um einfach jetzt ein erfülltes Arbeitsleben zu haben. So, dem gegenüber steht natürlich auch der gewisse Projektdruck, den man hat, ich sage mal, Zeit und Budget hängen einem im Nacken. Das ist immer ein Problem. Mit der Zeit, das ist wie Raum und Zeit bei Einstein, die korrelieren miteinander. Überschreite ich die Zeit, überschreite ich quasi auch das Budget, das ist ein Punkt letztendlich. Und man hat natürlich auch die Umsetzungsqualität im Nacken, weil man natürlich als Projektmanager weiß, dass die schweren Probleme, die schweren Fehler, die werden ganz am Anfang gemacht. Das sind in der Regel Designfehler, konzeptionelle Fehler und hier, genau hier in diesem Punkt, halte ich das total wichtig, eine Waterfall-Schiene mit einzubauen. Einfach um Dinge, diese konzeptionellen Probleme relativ frühzeitig erkennen zu können. Als aber auch hier eine Direktive vorzugeben. Wo wollen wir denn hin und was sind unsere Meilensteine? Also diese Definition, damit wir auch wissen, was sind unsere Deliverables in der Zeit X. Das muss

Bei einem hybriden Vorgehen sieht der Experte den Vorteil darin, dass die agilen Anteile vor allem positiv auf die Motivation der Teammitglieder wirken sowie die Möglichkeit, ein kontinuierliches Feedback zu erhalten. In der Kombination mit einem traditionellen Vorgehen bleibt auch eine Planbarkeit erhalten und die Projektziele On Scope, On Budget und On Schedule gehen im Gesamtkontext nicht verloren. Ein Projektmanager behält den Gesamtüberblick darüber und überwacht die Qualität des Projektes als Ganzes. Dieser Gesamtüberblick ermöglicht es, konzeptionelle Fehler frühzeitig zu entdecken und diese zu beseitigen sowie einen Projektplan vorzugeben, um so Terminplanung einzuhalten.

EB05\_Rohrbeck, Christian

ganz klar auf dem Schirm sein und meine Erfahrung ist einfach, auch wenn man agil im Prinzip den Scope immer wieder entscheidet, das große Ganze, dieser Endtermin, das darf nicht aus dem Blick geraten und hier bedarf es schon einer klassischen Unterstützung von einem Projektmanager an der Stelle.

EB05\_Rohrbeck, Christian: 45 - 45 (0)

Auf der anderen Seite würde ich von der Seite der agilen Methoden, kann ich mir vorstellen, dass man durchaus auch in Sprints arbeitet, also wenn man zum Beispiel darüber spricht, dass man in einer iterativen Vorgehensweise einzelne Fachbereiche in dieses Projekt hinzuholt und die darin schult. Das wäre eine Sache, wo man iterativ vorgehen könnte, einfach damit man die Gelegenheit hat, aus den Erfahrungen des einen Onboarding des Fachbereiches für den nächsten zu lernen. Ich würde bei den agilen Methoden auch Wert darauf legen, schnell Erfolge sichtbar zu machen, dass man also nicht mit einem beispielsweise Big Bang arbeitet und alles passiert im Kämmerchen bis das ganze Thema leicht geht, sondern man muss eben zwischendrin auch die Erfolge sichtbar machen. Also immer ein Stückchen mehr, ja, Inkrement sozusagen, ein Stückchen mehr zeigen, was man geschaffen hat, um die Projektmitglieder einfach am Ball zu behalten und auch den Stakeholdern zu zeigen, dass man hier konsequent Mehrwert schafft für die Organisation.

EB07\_Janke, Tino: 35 - 35 (0)

Der Experte hebt hervor, dass positiv beim Hinzufügen von agilen Methoden das arbeiten in Sprints ist und dadurch iterativ neue Bereiche in das Projekt integriert und so eine Skalierung möglich ist. Mit den Erfahrungen, die mit jeder Iteration gesammelt werden, kann eine Verbesserung für die nächste Abteilung, die hinzugezogen wird, erhalten werden.

EB07\_Janke, Tino

7 - Nachteile gewähltes Vorgehensmodell\7.1 - Nachteile agiles Vorgehensmodell\7.1.1 - Nachteil agil fehlende Planungssicherheit

So, dem gegenüber steht natürlich auch der gewisse Projektdruck, den man hat, ich sage mal, Zeit und Budget hängen einem im Nacken. Das ist immer ein Problem. Mit der Zeit, das ist wie Raum und Zeit bei Einstein, die korrelieren miteinander. Überschreite ich die Zeit, überschreite ich quasi auch das Budget, das ist ein Punkt letztendlich. Und man hat natürlich auch die Umsetzungsqualität im Nacken, weil man natürlich als Projektmanager weiß, dass die schweren Probleme, die schweren Fehler, die werden

Der Experte verweist bei den Nachteilen von agilen Vorgehensweisen darauf, dass in Projekten häufig ein starker Projektdruck vorherrscht. Der zeitliche Faktor wird als sehr wichtig bezeichnet und wird dieser überschritten, kann dies andere negative Konsequenzen nach sich ziehen, wie erhöhte Kosten oder ein Problem mit der Umsetzungsqualität. Dies würde sich durch eine ausführliche Konzeptionsphase vermeiden lassen.

EB05\_Rohrbeck, Christian

ganz am Anfang gemacht. Das sind in der Regel Designfehler, konzeptionelle Fehler und hier, genau hier in diesem Punkt, halte ich das total wichtig, eine Waterfall-Schiene mit einzubauen.

EB05\_Rohrbeck, Christian: 45 - 45 (0)

Ich glaube, was beim Wasserfallmodell sicherlich noch mal ein Vorteil sein dürfte gegenüber agilen Methoden ist, bei agilen Methoden verliert man sich dann schon vielleicht zu schnell in den Detailfragestellungen, weil man sich ja mit der Umsetzung der kleinsten, von einem MVP beschäftigt, also so einem Minimal Viable Product, das ist so eine kleine Einheit, die nach jedem Sprint dann irgendwie fertig ist und womit man dann arbeiten kann. Und ich glaube bei so Wasserfallprozessen einfach, natürlich man hat ein bisschen länger Zeit, sich in der Planung Gedanken darüberzumachen, ggf. Seiteneffekte, die durch bestimmte Vorhaben auftreten. Auch so diese übersichtliche Gesamtblick-Brille, diese Berühmte. Die ist sicherlich auch in so einem Wasserfallprojekt eher gegeben, als dass es, ja, man verliert in der Umsetzung bei so kleinen Sprints dann manchmal eher den Überblick, als das bei so Wasserfallverfahren der Fall ist.

EB06\_Hesse, Moritz: 43 - 43 (0)

Übersicht behalten und sich trotzdem gewisse Konventionen schaffen, einen gewissen Fahrplan schaffen, einen gewissen Rahmen schaffen, ein gewisses Framework schaffen, was es ermöglicht, das klingt widersprüchlich, aber die Agilität ein bisschen einzugrenzen. Das klingt widersprüchlich, weil, ich will ja eigentlich genau durch diese Agilität das Team sich selbst befähigen und sprudeln lassen und auch Kreativität enablen. Nichtsdestotrotz glaube ich, bei Low Code Lösungen und auch bei agilen Projekten ist es hilfreich, wenn man sich initial zu Beginn des Projekts oder eben auf unternehmerischer Ebene, als Konvention für alle Projekte der Zukunft, sich gemeinsam bestimmte Konventionen, Restriktionen entwickelt oder Best Practices sich vor

Der Experte weist auf die negativen Konsequenzen hin, die entstehen können, wenn durch das kleinteilige Arbeiten in agilen Projekten der Gesamtüberblick verloren geht. Es besteht die Gefahr, dass die Planungssicherheit verloren geht, welche bei den ausführlichen Planungsaspekten im traditionellen Vorgehen nicht entstehen. Es wird sich zu wenig Zeit für die ausführliche Planung genommen, was dazu führt, dass gewisse Risiken nicht oder zu spät berücksichtigt werden.

EB06\_Hesse, Moritz

Augen führt, wie man da diesen Wildwuchs, den Verlust der Übersicht einfach kompensieren kann.

EB06\_Hesse, Moritz: 49 - 49 (0)

Eine Weiterentwicklung des Programms. Also ich will ja nicht nur Fehlerpflege machen, sondern ich will ja vielleicht auch ganz neue Applikationen dann hineinbringen, nach einem halben Jahr oder einem Jahr, oder wenn es ein Superprogramm war, vielleicht auch nach drei bis vier Jahren erst, aber ich will es. Auch da muss in dieser Applikation, in dieser Low Code/ No Code-Applikation, muss schon angelegt sein, da und da sind die Schrauben, um dieses oder jenes zu verändern. Und auch das muss gut vorbereitet sein. Dann die Möglichkeit, schnell die Sache nicht nur zu entwickeln, sondern es ist ja tatsächlich dann auch ein IT-Programm zu implementieren. Das heißt also, auch diese Implementierung, die muss dann abgestimmt vollzogen werden. Auch da muss man sich drüber Gedanken machen. Also, Releasemanagement lässt hier auch grüßen. Also, die Entstehung dieses Programms ist eine Sache, auch IT-Abteilungen können damit arbeiten. Die Dinge Pflege, Weiterentwicklung, Implementierung, diese ganzen Dinge, die sind halt auch immer mit zu bedenken, mit zu planen.

EB08\_Fey, Peter: 35 - 35 (0)

Es wird darauf hingewiesen durch den Experten, dass um das Projekt herum verschieden Abstimmungsfragen zu beantworten sind. Diese gehen über die Einführung hinaus und sollten Handlungsmöglichkeiten dafür bieten, wenn wie ein späteres Releasemanagement funktioniert und wie die Implementierung im Unternehmen erfolgt. Dafür ist eine gewisse Planung notwendig, die bei agilem Vorgehen verloren gehen kann.

EB08\_Fey, Peter

7 - Nachteile gewähltes Vorgehensmodell 7.1 - Nachteile agiles Vorgehensmodell 7.1.2 - Nachteile Projektcommitment

Stolpern, also eine Hürde ist bei so einem Vorgehen natürlich, da man bei solchen Projekten häufig auch mit wechselnden Teilnehmern, je nachdem, mit welchem Fachbereich man gerade zu tun hat, arbeitet. Dass man so ein bisschen die Konstanz in der Arbeit verliert. Dass manche Leute punktuell mal was machen und dann bleibt das alles liegen. Dann habe ich so meine halbfertigen Applikationen, oder Applikationen, die gut genug sind, dass sie sie anwenden können und dann kümmert sich keiner mehr darum. Also das ist natürlich eine Sache, die bei agilem Vorgehen schnell mal passiert. Insbesondere nämlich, wenn das Team nicht konstant ist.

EB01\_Eckhard Hauenherm: 49 - 49 (0)

Der Experte weist auf die Problemstellung hin, dass bei wechselnden Projektteilnehmern die Projektarbeit an Konstanz verliert. Als Beispiel führt er an, dass ein Zwischenstand erreicht ist, der zwar suboptimal, aber funktional ist und dass dieser nicht zu Ende entwickelt wird.

EB01\_Eckhard Hauenherm



Ich glaube, was beim Wasserfallmodell sicherlich noch mal ein Vorteil sein dürfte gegenüber agilen Methoden ist, bei agilen Methoden verliert man sich dann schon vielleicht zu schnell in den Detailfragestellungen, weil man sich ja mit der Umsetzung der kleinsten, von einem MVP beschäftigt, also so einem Minimal Viable Product, das ist so eine kleine Einheit, die nach jedem Sprint dann irgendwie fertig ist und womit man dann arbeiten kann. Und ich glaube bei so Wasserfallprozessen einfach, natürlich man hat ein bisschen länger Zeit, sich in der Planung Gedanken darüberzumachen, ggf. Seiteneffekte, die durch bestimmte Vorhaben auftreten. Auch so diese übersichtliche Gesamtblick-Brille, diese Berühmte. Die ist sicherlich auch in so einem Wasserfallprojekt eher gegeben, als dass es, ja, man verliert in der Umsetzung bei so kleinen Sprints dann manchmal eher den Überblick, als das bei so Wasserfallverfahren der Fall ist.  
EB06\_Hesse, Moritz: 43 - 43 (0)

Übersicht behalten und sich trotzdem gewisse Konventionen schaffen, einen gewissen Fahrplan schaffen, einen gewissen Rahmen schaffen, ein gewisses Framework schaffen, was es ermöglicht, das klingt widersprüchlich, aber die Agilität ein bisschen einzugrenzen. Das klingt widersprüchlich, weil, ich will ja eigentlich genau durch diese Agilität das Team sich selbst befähigen und sprudeln lassen und auch Kreativität enablen. Nichtsdestotrotz glaube ich, bei Low Code Lösungen und auch bei agilen Projekten ist es hilfreich, wenn man sich initial zu Beginn des Projekts oder eben auf unternehmerischer Ebene, als Konvention für alle Projekte der Zukunft, sich gemeinsam bestimmte Konventionen, Restriktionen entwickelt oder Best Practices sich vor Augen führt, wie man da diesen Wildwuchs, den Verlust der Übersicht einfach kompensieren kann.  
EB06\_Hesse, Moritz: 49 - 49 (0)

Der Experte weist darauf hin, dass durch das agile Arbeiten und der damit verbundenen Umsetzung eines minimal funktionalen Zwischenprodukts es zu Problemen kommen kann, wenn diese nicht weiterentwickelt werden, weil das Zwischenprodukt trotzdem funktional genug ist, um produktiv eingesetzt zu werden, allerdings nicht in vollem Umfang der Möglichkeiten.  
EB06\_Hesse, Moritz

7 - Nachteile gewähltes Vorgehensmodell\7.1 -

Ja, ist jetzt aber weniger der Power Plattform geschuldet als der agilen Methodik insgesamt. Wenn ich jetzt in ein

Der Experte weist auf eine allgemeine Problemstellung von agilen Vorgehensweisen hin, welche nicht zwingend

Nachteile agiles Vorgehensmodell 7.1.3 - Nachteile agil Fehlende Erfahrung

Unternehmen komme, die mit der agilen Methode bisher noch nie zusammengearbeitet haben, die tun sich natürlich erst mal schwer, weil ich dann erst mal eine agile Projektvorgehensweise mit dem Kunden klären muss, bevor es überhaupt an die Inhalte gehen kann. Also, das ist so die Hürde. Ist aber losgelöst von dem Thema Power Platform an sich. Das Gleiche hätte ich, wenn es jetzt um eine Dynamics Einführung geht, zum Beispiel. Im schlimmsten Fall geht man dann halt auch wirklich auf ein Wasserfallmodell über, wenn der Kunde sagt: "Agil ist definitiv nicht unseres. Wir brauchen den kompletten Plan von vorn bis hinten." Dann machen wir das auch.

EB03\_Fabian Gackstatter: 39 - 39 (0)

einen Zusammenhang mit NCLC-Einführungsprojekten hat. Bei fehlender Erfahrung mit agilen Vorgehensweisen ist es nötig, erst eine Befähigungsphase in Unternehmen durchzuführen oder zwangsweise mit traditionellen Vorgehensweisen zu arbeiten. Dies kann negative Konsequenzen auf den geplanten Projektverlauf haben.

EB03\_Fabian Gackstatter

Na gut, das kommt darauf an. Also wenn eine Unternehmung schon mehrere agile Projekte gefahren hat und weiß, wie es geht, was meine ich damit, die Mitarbeiter haben schon Erfahrung damit, dann sehe ich da keine Hürden. Aber wenn das sozusagen als Vehikel benutzt werden würde, um eines der ersten agilen Projekte zu nutzen, dann ist das schon kompliziert, weil, es kennt noch keiner so richtig die Rollen. Scrum Master, was macht der eigentlich den ganzen Tag? Für was ist der eigentlich zuständig? Ach so, der soll so Daily moderieren, aber da brauch ich doch keinen für. Also Daily, wieso soll ich jeden Morgen sagen, welche Sachen ich mache und wieso soll ich mich jetzt committen, was ich in zwei- oder vierwöchigen Sprints erledige. Und das sind, glaube ich, Punkte, das ist natürlich eine Hürde, die dann definitiv mit reinspielt, wie weit ist dieses Mindset dafür und was natürlich auch eine Hürde sein kann, ist immer bisschen der Punkt, dass Agilität so ein bisschen manchmal als Flexibilität wahrgenommen wird. Das heißt, dass vielleicht gewisse Stakeholder oder Personen auch das Backlog stark beeinflussen möchten und auch vielleicht die Priorisierung des Backlogs dabei. Was natürlich nicht immer so an der Stelle vorgesehen ist, wenn man es einmal abgestimmt hat. Und das sind so ein bisschen die Hürden, die man schon

Die Expertin hebt die Problemstellung hervor, dass Unternehmen, die keine Erfahrung mit agilem Vorgehen haben und dieses Vorgehens in einem Projekt das erste Mal durchführen, auf methodische Problemstellungen stoßen können. Es kann auch zu Widerständen kommen, wenn die Methodiken als nicht sinnvoll durch Projektteilnehmer erachtet werden, weil dort die Erfahrung fehlt.

Eine weitere Hürde, auf die Unternehmen stoßen können, ist, dass einzelne Personen aufgrund ihrer persönlichen Anforderungen den Backlog sehr stark beeinflussen, in Reihenfolge und Priorisierung. Diese personenbezogene Priorisierung ist aber nicht zwingend auch die beste für das Projekt, im Gegensatz zu einem klassisch durchgeführten Projekt, wo Aufgaben klar sind und eine Planung im Vorfeld erfolgt.

EB04\_Göttl, Simone

	<p>so sieht und die früher in einem klassischen Prinzip, da weiß halt jeder, was er zu tun hat, was er sein lassen darf und muss oder was er machen darf.</p> <p>EB04_Göttl, Simone: 39 - 39 (0)</p>	
<p>7 - Nachteile gewähltes Vorgehensmodell\7.1 - Nachteile agiles Vorgehensmodell\7.1.4 - Nachteil agil schlechte Dokumentation</p>	<p>Genau. Die Nachteile sind in der Regel ähnlich wie bei der IT. Niemand, der ein Programm schreibt, liebt es zu dokumentieren. Also es werden Dokumentationen gemacht, aber ich habe bisher noch keinen getroffen, der außer Lippenbekenntnissen zur Dokumentation wirklich dann auch dafür brennt, dass was er tut, da gut dokumentieren. Dass das getan wird, das ist ein Ausdruck der Professionalität und das ist ganz gut. Das heißt, je besser ein Programm, was ich beschrieben habe, dokumentiert ist, desto einfacher ist es und jetzt kommt das Stichwort, das ganz wichtig ist: zu pflegen. Wenn ich ein Programm schlecht dokumentiere und ich bin dann weg, wo auch immer hin.</p> <p>EB08_Fey, Peter: 35 - 35 (0)</p>	<p>Der Experte hebt die Problemstellung hervor, dass, ähnlich wie in vielen IT-Projekten, eine Dokumentation häufig mangelhaft ist. Auf diese Problemstellung sollte geachtet werden.</p> <p>EB08_Fey, Peter</p>
<p>7 - Nachteile gewähltes Vorgehensmodell\7.2 - Nachteile traditionelles Vorgehensmodell\7.2.1 - Nachteil trad. zu starr</p>	<p>Also das würde ich tatsächlich nicht als Wasserfall machen. Erstens, weil ich bei der Einführung zwar Ziele mir festlegen kann. Das mache ich bei jedem Cloud-Projekt, dass sozusagen die erste Frage ist, warum wollt ihr in die Cloud? Warum wollt ihr in diese Plattformen? Also gucken, was will das Unternehmen damit erreichen? Ich habe aber eine relativ schlechte Planbarkeit, weil ich erstens nicht genau weiß, welche Funktionen jetzt damit umgesetzt werden können oder wollen oder wo das Unternehmen arbeitet. Das spricht schon gegen einen klassischen, sequenziellen Wasserfallprozess, wo ich ja in der Regel ein fertiges Gewerk am Ende plane.</p> <p>EB01_Eckhard Hauenherm: 41 - 41 (0)</p> <p>Das ist ja genau das, ich habe mit einem iterativen Projekt meine Feedbackschleife und kann dann sehr schön mit der Plattform lernen, was uns die im Unternehmen bringt oder nicht bringt. Und ich kann auch mit relativ wenig Kosten schnell wieder aussteigen, wenn ich sage: "Nee, das ist für uns nichts." Oder "Die Plattform ist für uns nichts. Wir</p>	<p>Der Experte weist auf die schlechte Planbarkeit von NCLC-Einführungsprojekten hin. Angesichts dessen sieht er wenig Möglichkeiten für den umfassenden Einsatz von traditionellen Projektmanagement-Methoden. Diese schlechte Planbarkeit ergibt sich aus einem Mangel an Wissen darüber, wie die Plattform in das Unternehmen implementiert werden und welche Funktionen diese genau erfüllen kann.</p> <p>Es wird auch darauf hingewiesen, dass Veränderungen am Projektumfang oder ein schnelles Reagieren auf veränderte Anforderungen nur schwer möglich ist mit einem traditionellen Vorgehen, da bei diesen ein Gesamtergebnis am Ende präsentiert wird oder in verhältnismäßig großen Zeitabschnitten Zwischenergebnisse.</p> <p>Aufgrund des plangetriebenen Charakters von traditionellen Systemen ist der initiale Kostenaufwand auch wesentlich höher. Diese werden im Wesentlichen durch die umfangreichen Planungen im Vorfeld der Einführung nötig.</p>

nehmen mal eine andere." Das kann ich bei einem fertig geplanten, also bei einem klassischen, sequenziellen Projekt habe ich das nicht. Da habe ich am Ende alles eingeführt und keiner kann damit arbeiten.

EB01\_Eckhard Hauenherm: 41 - 41 (0)

Das würde ich bei einem plangetriebenen Ansatz oder auch bei einem hybriden Ansatz habe ich es häufig so, dass ich sehr viele Vorinvestitionen habe, um erst mal einen Plan aufzubauen, um erst mal die Plattform zu checken, um erst mal die Plattform auszuwählen und so weiter und so fort. Das wird deutlich teurer und führt auch eher zu Frustration.

EB01\_Eckhard Hauenherm: 45 - 45 (0)

Das heißt, wenn ich die jetzt alle in ein Wasserfall versuchen würde reinzupacken. Der Wasserfall ist immer nur so schnell wie das langsamste Teammitglied und in der agilen Methode habe ich eben die Chance, Sachen auch vernünftig zu parallelisieren und zu sagen: "Ok, im Entwickler-Stream bauen wir jetzt diese zehn Applikationen. Das ist unser Entwicklungsstream und im Thema Changemanagement, das sind die drei Communitys, die wir aufsetzen, die parallel dazu laufen."

EB03\_Fabian Gackstatter: 37 - 37 (0)

Ich glaube, bei der Einführung und der nachhaltigen Nutzung von Low Code-Lösungen tun sich Unternehmen keinen Gefallen, an einem starren Wasserfallmodell festzuhalten. Lassen Sie mich ausführen, warum. Eine wesentliche Stärke von Low Code-Lösungen zeichnet sich durch diese Flexibilität und Schnelligkeit aus. Diese Schnelligkeit ermöglicht den Kunden, zum Beispiel, sehr, sehr schnell auf regulatorische Anforderungen reagieren zu können. Das klassische Wasserfallmodell, das klassische Projektmanagementmodell, hat genau diese Schwäche, dass es nicht flexibel und schnell ist. Sondern im Gegenteil, es baut sich genau an diesen Projektphasen auf, wo die nachgelagerten Phasen

EB01\_Eckhard Hauenherm

Der Experte vertritt die Ansicht das es wesentlich schwieriger ist, verschiedene Aufgaben zu parallelisieren bei traditionellem Vorgehen, als das bei agilem Vorgehen der Fall ist.

EB03\_Fabian Gackstatter

Der Experte ist überzeugt, dass es ein wesentlicher Nachteil ist, bei der Einführung auf ein starres Wasserfallmodell zu setzen. Dies begründet sich in der Schnelligkeit und Flexibilität der Lösung. Er vertritt dabei die Ansicht, dass klassische Modelle diese Eigenschaften nicht besitzen durch die Abhängigkeit der verschiedenen Phasen voneinander.

EB06\_Hesse, Moritz

	<p>immer erst den Abschluss vorgelagerten Phasen erfordern EB06_Hesse, Moritz: 41 - 41 (0)</p>	
<p>7 - Nachteile gewähltes Vorgehensmodell\7.2 - Nachteile traditionelles Vorgehensmodell\7.2.2 - Nachteil trad. Dauer</p>	<p>Das heißt, wenn ich die jetzt alle in ein Wasserfall versuchen würde reinzupacken. Der Wasserfall ist immer nur so schnell wie das langsamste Teammitglied und in der agilen Methode habe ich eben die Chance, Sachen auch vernünftig zu parallelisieren und zu sagen: "Ok, im Entwickler-Stream bauen wir jetzt diese zehn Applikationen. Das ist unser Entwicklungsstream und im Thema Changemanagement, das sind die drei Communitys, die wir aufsetzen, die parallel dazu laufen." EB03_Fabian Gackstatter: 37 - 37 (0)</p> <p>Es ist halt unsere Empfehlung das zu tun, weil wir damit in der Regel bessere Ergebnisse erzielen, wenn der Kunde den Weg aber so nicht mitgehen kann, dann machen wir es auch ganz normal im Wasserfall. Die Geschichte dauert in der Regel länger und ist dann halt auch schwieriger für den Kunden zu greifen, weil es dann halt auch einfach länger dauert, bis ich mal was in der Hand habe hinterher. EB03_Fabian Gackstatter: 39 - 39 (0)</p>	<p>Der Experte verweist darauf, dass die traditionellen Vorgehensmodelle wesentlich mehr Zeit benötigen, aufgrund der Abhängigkeiten der Phasen zueinander. Eine vernünftige Parallelisierung ist dabei nicht möglich. Dies zeigen auch Erfahrungswerte, welche der Experte in der Praxis gesammelt hat. EB03_Fabian Gackstatter</p>
	<p>Ich glaube, bei der Einführung und der nachhaltigen Nutzung von Low Code-Lösungen tun sich Unternehmen keinen Gefallen, an einem starren Wasserfallmodell festzuhalten. Lassen Sie mich ausführen, warum. Eine wesentliche Stärke von Low Code-Lösungen zeichnet sich durch diese Flexibilität und Schnelligkeit aus. Diese Schnelligkeit ermöglicht den Kunden, zum Beispiel, sehr, sehr schnell auf regulatorische Anforderungen reagieren zu können. Das klassische Wasserfallmodell, das klassische Projektmanagementmodell, hat genau diese Schwäche, dass es nicht flexibel und schnell ist. Sondern im Gegenteil, es baut sich genau an diesen Projektphasen auf, wo die nachgelagerten Phasen immer erst den Abschluss vorgelagerten Phasen erfordern EB06_Hesse, Moritz: 41 - 41 (0)</p>	<p>Der Experte vertritt die Auffassung, aufgrund einer schlechter zu realisierenden Parallelisierung von Aufgaben benötigen diese wesentlich mehr Zeit. EB06_Hesse, Moritz</p>

7 - Nachteile gewähltes Vorgehensmodell\7.3 - Nachteile hybrides Vorgehensmodell	<p>Zu einem hybriden Modell... Kann man muss man aber nicht. Also man könnte natürlich tatsächlich für so ein Projekt auch ein Wasserfall drum herum legen, sage ich mal, für so ein hybrides Modell. Sehe ich aber tatsächlich keinen Mehrwert drin. Es gibt ein paar Dinge die ich anfangs in der Zielsetzung machen würde, die vielleicht so ein bisschen nicht hundertprozentig Agil sind und da ist wieder mein Lieblingsthema Governance da, das sollte man am Anfang entscheiden wie viel Risiken gehe ich damit ein und wie kann ich die Risiken streuen. Da muss am man tatsächlich ganz klar am Anfang mit IT und Operations und was weiß ich, Security Officer, wer auch immer im Unternehmen dafür zuständig ist, sich wirklich mal klar machen, welche Datenschutzanforderungen haben wir? Mit welchen Daten gehen wir um? Manchmal führt das zu einem Vorlauf, in so einem Projekt, das ich erstmal mit Datenklassifikationen und so weiter anfangen muss zu arbeiten. Das ich sicherstelle das bestimmte Daten da nicht rauskommen. Aber das kann man auch Agil nach und nach machen in dem Mann, wenn man das Risiko kennt, tatsächlich guckt wo setzen wir das ein? Wem gebe ich Zugriff auf die Plattform und so weiter? Das kann man schon noch steuern.</p>	<p>Der Experte sieht keinen Mehrwert in der Realisierung des Einführungsprojektes über einen hybriden Ansatz. Es gibt zwar Bestandteile eines Projekts, die in einem Wasserfallrahmen funktionieren würden, wobei er die Planung vor allem zu Beginn des Projekts anführt, allerdings lässt sich dies mit der gleichen Qualität auch in einer agilen Methodik umsetzen.</p> <p>EB01_Eckhard Hauenherm</p>
7 - Nachteile gewähltes Vorgehensmodell\7.3 - Nachteile hybrides Vorgehensmodell\7.3.1 - Nachteil hybrid zu starr	<p>Das würde ich bei einem plangetriebenen Ansatz oder auch bei einem hybriden Ansatz habe ich es häufig so, dass ich sehr viele Vorinvestitionen habe, um erst mal einen Plan aufzubauen, um erst mal die Plattform zu checken, um erst mal die Plattform auszuwählen und so weiter und so fort. Das wird deutlich teurer und führt auch eher zu Frustration.</p> <p>EB01_Eckhard Hauenherm: 43 - 43 (0)</p> <p>EB01_Eckhard Hauenherm: 45 - 45 (0)</p>	<p>Der Experte stellt heraus, dass starres plangetriebenes Vorgehen, welches einige hybride Modelle haben können, aufgrund der Anteile des traditionellen Projektmanagements zu höheren Kosten führt.</p> <p>EB01_Eckhard Hauenherm</p>
7 - Nachteile gewähltes Vorgehensmodell\7.3 - Nachteile hybrides Vorgehensmodell\7.3.2 - Nachteil hybrid Kosten	<p>Das würde ich bei einem plangetriebenen Ansatz oder auch bei einem hybriden Ansatz habe ich es häufig so, dass ich sehr viele Vorinvestitionen habe, um erst mal einen Plan aufzubauen, um erst mal die Plattform zu checken, um erst mal die Plattform auszuwählen und so</p>	<p>Der Experte stellt heraus, dass die initialen Kosten für einen hybriden Ansatz höher sind, da die Planungsaufgaben zu Beginn des Projekts bereits einen hohen Arbeitsaufwand bedeuten, ohne dabei messbaren Mehrwert zu generieren.</p>

	<p>weiter und so fort. Das wird deutlich teurer und führt auch eher zu Frustration. EB01_Eckhard Hauenherm: 45 - 45 (0)</p>	<p>EB01_Eckhard Hauenherm</p>
<p>7 - Nachteile gewähltes Vorgehensmodell\7.3 - Nachteile hybrides Vorgehensmodell\7.3.3 - Nachteil hyb. Fehlende Erfahrung mit einem der Modelle</p>	<p>Na gut, das kommt darauf an. Also wenn eine Unternehmung schon mehrere agile Projekte gefahren hat und weiß, wie es geht, was meine ich damit, die Mitarbeiter haben schon Erfahrung damit, dann sehe ich da keine Hürden. Aber wenn das sozusagen als Vehikel benutzt werden würde, um eines der ersten agilen Projekte zu nutzen, dann ist das schon kompliziert, weil, es kennt noch keiner so richtig die Rollen. Scrum Master, was macht der eigentlich den ganzen Tag? Für was ist der eigentlich zuständig? Ach so, der soll so Daily moderieren, aber da brauch ich doch keinen für. Also Daily, wieso soll ich jeden Morgen sagen, welche Sachen ich mache und wieso soll ich mich jetzt committen, was ich in zwei- oder vierwöchigen Sprints erledige. Und das sind, glaube ich, Punkte, das ist natürlich eine Hürde, die dann definitiv mit reinspielt, wie weit ist dieses Mindset dafür und was natürlich auch eine Hürde sein kann, ist immer bisschen der Punkt, dass Agilität so ein bisschen manchmal als Flexibilität wahrgenommen wird. Das heißt, dass vielleicht gewisse Stakeholder oder Personen auch das Backlog stark beeinflussen möchten und auch vielleicht die Priorisierung des Backlogs dabei. Was natürlich nicht immer so an der Stelle vorgesehen ist, wenn man es einmal abgestimmt hat. Und das sind so ein bisschen die Hürden, die man schon so sieht und die früher in einem klassischen Prinzip, da weiß halt jeder, was er zu tun hat, was er sein lassen darf und muss oder was er machen darf. EB04_Görtl, Simone: 39 - 39 (0)</p>	<p>Die Expertin weist darauf hin, dass es bei der Umsetzung zu Problemen kommen kann, da ein erhöhtes Maß an Wissen für beide Vorgehensweisen erfüllt sein muss. Es muss Wissen vorhanden sein, wie traditionelle Ansätze funktionieren und wie agile Ansätze funktionieren, welche Tools und Rollen vorhanden sind. Bei einem hybriden Vorgehen ist es erforderlich, dass für beide Modelle, je nach Vorwissen der Organisation, eine Befähigung erfolgt, um negative Konsequenzen zu vermeiden. EB04_Görtl, Simone</p>
<p>7 - Nachteile gewähltes Vorgehensmodell\7.3 - Nachteile hybrides Vorgehensmodell\7.3.4 - Nachteil hybrid. kombi. Probleme reine Vorgehensweisen</p>	<p>Ja, wenn man halt schnell durchstarten will, werden manche Governance Themen zu kurz kommen, sage ich einfach mal so. Zum Beispiel haben wir noch nicht die perfekte Lösung, wie wir das in unseren normalen Softwareentwicklungsprozess, wie die Abstimmung mit Datenschutz, Cybersecurity, Betriebsrat zu Themen ist. Also, da haben wir jetzt</p>	<p>Der Experte stellt heraus, dass bei einem hybriden Vorgehen vor allem eine Problematik entsteht bei planungstechnischen Fragestellungen. Wird die Planung unzureichend ausgeführt und Rahmenbedingungen und Prozesse werden nicht bereits im Vorfeld optimal definiert, kann es dazu kommen, dass diese dann während der Umsetzung</p>

noch keinen zu hundert Prozent sauberen Prozess, der vollautomatisch im Hintergrund abläuft. Da wird es, glaube ich, immer dann Übergangsphasen geben, wo man dann noch offene Fragen hat und dann mit der Hand am Arm arbeitet. Zum Beispiel hat es ein bisschen gedauert, bis unsere, wenn ich Premium Features nutzen will in der Power Plattform, bis ich mir dann eine Lizenz bestellen kann, ist jetzt mittlerweile in unserem normalen Bestellsystem, wie ich mir auch einen Monitor und Laptop bestelle, da kann ich mir auch die Lizenz bestellen. Aber das war natürlich nicht von Minute eins da und ja, solche Strukturen wachsen dann halt mit der Plattform, statt dass sie von Anfang an voll perfekt abgebildet sind. Ist einfach ein Trade Off, den man dann hinnehmen muss, dafür dass man schneller unterwegs ist.  
EB02\_Rosengrün, Robin: 43 - 43 (0)

des Projekts zu Verzögerungen und Prozessablaufproblemen führen.  
EB02\_Rosengrün, Robin

Also, definitiv ist das auch das nicht die optimale Lösung. Ich kann dir jetzt einfach nur sagen, die Nachteile sind halt die, die jetzt auch die beiden Einzel- oder reinen Modelle mitbringen würde jeweils auf ihrer Seite. Also es gibt jetzt keinen zusätzlichen hybriden Nachteil.

Der Experte hebt hervor das ein hybrider Ansatz immer auch die Nachteile der verschiedenen PM-Ansätze mit sich bringen. Diese können je nach Schwerpunkt variieren.  
EB05\_Rohrbeck, Christian

EB05\_Rohrbeck, Christian: 51 - 51 (0)

8 - Tools Vorgehensmodell\8.1 - Tools agiles Vorgehen

Was nehme ich denn da noch gerne?  
Also, Scrum ist so für den Prozess eine ganz gute Geschichte, so mit Sprints arbeiten, regelmäßig ein Review dazu machen, so mal darüber zu gucken. Ja gut, aber wo man guckt, was man typischerweise hat. Für so ein Einführungsprojekt, also ich muss immer so ein bisschen anpassen, dass ich mich darauf konzentriere. Wir haben jetzt so ein Cloud- Einführungsprojekt bzw. ein No Code/ Low Code-Einführungsprojekt. Da brauche ich eigentlich nicht zwingend große Design-Thinking oder ähnliche Techniken oder Produktgestaltungstechniken spielen da für mich erst mal keine große Rolle. Also mir gehts tatsächlich um den Ablauf, um die Feedbackschleifen. Was man natürlich gut nutzen kann, ist

Der Experte hebt als bevorzugtes Tool des agilen Methodenbaukastens das Scrum-Model hervor. Die Möglichkeit, in Sprints zu arbeiten und regelmäßige Reviews durchzuführen, wird als positiv wahrgenommen. Als weiteres Tool, das als nützlich betrachtet wird, wurde die Kanban Methodik angeführt, um den Entwicklungs- und Applikationsprozess abzubilden. Tools wie Design Thinking und User Journey werden nur als bedingt zielführend erachtet, da diese zu komplex sind für die ersten Anwendungen, die in einem Einführungsprojekt erstellt werden.  
EB01\_Eckhard Hauenherm



eventuell so was wie Kanban, um so ein bisschen den Applikationsprozess oder den Entwicklungslebenszyklus sozusagen abzubilden in einem Board. Das kann man natürlich ganz gut nutzen. Wenn man so weit ist, aber das ist typischerweise erst in späteren Schritten bei solchen Projekten, dass man wirklich Applikationen für eine Zielgruppe entwickelt, dann könnte man natürlich auch mit so Design Thinking Personas oder ähnlichen Dingen, User Journeys, arbeiten und so was. Aber meistens sind die Applikationen, die man im ersten Schritt in so einem Projekt macht, eigentlich so einfach gestrickt, dass ich mit solchen Tools noch nicht viel machen kann. Das wäre sozusagen in späteren Schritten, kämen dann wirklich so User Journey und Story Mapping und solche Sachen, könnten da eine Rolle spielen.

EB01\_Eckhard Hauenherm: 53 - 53 (0)

Aber, wir haben unsere Weekly's, wo wir die Themen besprechen, dann bisschen langfristige Planung, wo wir hinwollen, wo wir zusammenarbeiten, aber nutzen da jetzt auch keine festen Tools aus dem agilen Baukasten.

EB02\_Rosengrün, Robin: 45 - 45 (0)

Der Experte führt als nützliches Tools an, Weeklys durchzuführen, unter anderem, um eine langfristige Planung zu entwickeln sowie das weitere Vorgehen zu besprechen und Aufgaben zu verteilen.

EB02\_Rosengrün, Robin

Also, wir sind normalerweise agil unterwegs. Das heißt, wir fangen an mit einem Sprint Null, in dem wir erst mal eine Reihe von Workshops durchführen, um den Kunden auf ein gewisses Niveau aufzuschauen und nach diesem Sprint Null planen wir dann wirklich Entwicklungssprints, um zu sehen: "Ok, das sind jetzt die Arbeitspakete, die wir tatsächlich implementieren möchten, in dem und dem Zeitraum." Auch immer dem Umstand geschuldet, wie viel Mitarbeit habe ich jetzt auf Kundenseite. Also, habe ich tatsächlich dedizierte Ressourcen, die mir zu 100 % zur Verfügung stehen. Das heißt, ich kann auch wirklich jeden Tag Arbeitspakete abarbeiten. Oder habe ich eben nur Leute, die maximal 50 % zur Verfügung stehen, dann brauche ich natürlich entsprechend länger, um die Pakete umzusetzen.

EB03\_Fabian Gackstatter: 33 - 33 (0)

Der Experte hebt hervor das er Tools nutzt, die auf der Scrum Methodik aufgebaut sind. Dies beginnt mit einem Sprint 0, der zur Befähigung der Projektteilnehmer dient und danach wird in parallelen Sprints weitergearbeitet. Kanban ist ebenfalls ein Tool, das eingesetzt wird, jedoch primär wird eine Scrumlogik verfolgt.

EB03\_Fabian Gackstatter

Also wir haben in Microsoft intern unsere eigene Methodik, die sehr stark an Scrum angelehnt ist.

**I: RS** Benutzt ihr noch andere oder ist das wirklich so diese Scrum Logik?

**B: FG** Scrum Methodik ist die Basis für uns.

EB03\_Fabian Gackstatter: 41 - 43 (0)

Benutzen wir im Sinne der Tools, wird Kanban eingesetzt. Aber, wie gesagt, wir haben eine Scrum ähnliche Methodik, unsere Surestep Methodik, die größtenteils auf Scrum aufsetzt, die aber zum Beispiel auch Kanban Boards im (**unverständlich**) umsetzt. Also wir versuchen da so ein bisschen, das praktischste aus beiden Welten oder aus mehreren Welten zusammenzusuchen, was sich in unseren Implementierungsprojekten bewährt hat. Spiegelt sich dann auch in den Artefakten wider, die erstellt werden. Spiegelt sich auch in den Rollen wider, die wir besetzen innerhalb dieser Projekte. Also kein hundertprozentiges Scrum, aber jemand, der Scrum beherrscht, wird sich bei uns sehr schnell zu Hause fühlen

EB03\_Fabian Gackstatter: 45 - 45 (0)

Was man auch noch sieht. Also man muss ja jetzt überlegen, ob man einen Projektstatusbericht macht, wenn man zum Beispiel in Tools wie Jira ein Backlog pflegt für so ein Einführungsprojekt, dann steckt ja dort die Wahrheit drin und ich kann mir dort anzeigen lassen, an welchen Punkten welche Sachen im Backlog sind, welche Sachen im aktuellen Sprintbacklog sind und krieg ja auch dann über gewisse Kennzahlen, die so ein Scrum-Team hat, wie viele Feature Points, habe ich jetzt zum Beispiel im letzten Sprint geschafft und so weiter, kriege ich ja Input darüber, wie es läuft.

EB04\_Görtl, Simone: 37 - 37 (0)

Die Expertin hebt hervor, dass die meist genutzte Methodik ein Scrumsystem ist in dem Umfeld, das sie in ihrer Praxis sieht. Dabei hebt sie hervor, dass Tools wie Kanban mit genutzt werden, allerdings nicht als das primäre Modell, sondern als Bestandteil des Scrumvorgehens. Ein weiteres Tool zur Fortschrittsmessung kann ein Projektstatusbericht sein, der sich aus dem Product Backlog und den Sprint Backlogs ergibt. Für die Backlogs sollte allerdings ein Monitoringtool verwendet werden, die Expertin nennt hier beispielhaft Jira.

EB04\_Görtl, Simone

**B: SG** Kanbanboards nutzen an der Stelle ist natürlich auch irgendwie immer. Ich habe die Erfahrung, zumindest dort,

wo ich unterwegs bin, dass Scrum irgendwie so eine gängige Praxis ist. Es ist so ein bisschen auch in Mode gekommen und auch, von dem Framework her, relativ einfach. Das ist so, ich nenne es mal so, dass es so klein ist, also klein ist relativ, aber es ist kein Riesen- Einführungsprojekt, wo ich mehrere hundert Mann habe, dann würde ich zu so einem skalierten Framework tendieren, wie SAFe, oder man könnte auch, wenn es mehrere Teams sind, an ein Scrum of Scrum denken oder ein Nexus. Die Erfahrung zeigt aber, zumindestens dort, wo wir unterwegs sind, dass viele Scrum als a und o nutzen oder in abgewandelter Form stattfindet. Das ist so ein bisschen Mode.

EB04\_Göttl, Simone: 41 - 41 (0)

Agile Elemente, die wirklich sich sehr bewährt haben, sind zum Beispiel, ein Daily zu machen, ein Daily Stand-up - 15 Minuten am Anfang des Tages und auch Retrospektiven sowie auch eine graphische Übersicht der To-dos an einem Kanban-Board.

EB05\_Rohrbeck, Christian: 39 - 39 (0)

Ich habe eigentlich gute Erfahrung mit Scrum gemacht. Insbesondere auch weil ein Leitsatz von Scrum selber ja ist „Halte dich nicht zu sehr an die Regeln, sondern lege es dir im Zweifelsfall auch so ein bisschen selber aus.“ Insofern sind ja alle agilen Frameworks nicht zu verstehen, also nach meinem Kenntnisstand, nicht zu verstehen als ein festes hartes Korsett, in dem gearbeitet wird, sondern eine Empfehlung, wie man bestehende nicht agile Methoden verbessern kann oder wie man die Schwächen quasi ausgleichen kann. Insofern, ich habe in verschiedenen Unternehmen mit Scrum gearbeitet, die eine unterschiedliche Ausprägung hatten, also die unterschiedlichen Parameter mitgebracht haben, was Sprintlänge anging, was Artefakte in dem Prozess anging. Also die einen haben eine Sprint-Retrospektive gemacht, die anderen haben es nicht gemacht. Und insofern sehe ich da jetzt

Der Experte hebt als agile Methode die Möglichkeit von Dailys hervor und Retrospektiven. Als Visualisierungstool sieht er Kanabnboards als geeignetes Tool, um Aufgaben darstellen.

EB05\_Rohrbeck, Christian

Der Experte führt als bevorzugte agile Methodik die Scrum-Methode an. Er verweist dabei darauf, dass Scrum viele flexible Möglichkeiten bietet, sich an die entsprechende Projektumgebung anzupassen. Es wird nicht als starres Regelwerk wahrgenommen, sondern eher als Handlungsempfehlungen, von denen auch abgewichen werden kann. Dies ermöglicht es, bestehende Schwächen auszugleichen und durch Abänderung einen besseren Projekterfolg zu erzielen.

EB06\_Hesse, Moritz

keine Schwächen, wo ich sagen würde: OH, da ist Scrum überhaupt nicht geeignet, da müsste man eher Kanban nehmen.

EB06\_Hesse, Moritz: 55 - 55 (0)

Auf der anderen Seite würde ich von der Seite der agilen Methoden, kann ich mir vorstellen, dass man durchaus auch in Sprints arbeitet, also wenn man zum Beispiel darüber spricht, dass man in einer iterativen Vorgehensweise einzelne Fachbereiche in dieses Projekt hinzuholt und die darin schult.

EB07\_Janke, Tino: 35 - 35 (0)

Und ich bin selbst auch immer ein großer Freund von Retrospektiven, also, das wäre auch ein Thema, was ich aus der agilen Welt mit einbeziehen würde, dass man sich einfach regelmäßig hinterfragt, was haben wir geschafft? Wie lief das? Also, wie war der Weg bis hierher? Was können wir einfach für die Zukunft noch besser machen?

EB07\_Janke, Tino: 35 - 35 (0)

Es geht halt auch in vielerlei Hinsicht in den Bereich der User-Stories hinein. Also auch agile Elemente, die ja eine große Rolle spielen, sind da drin. Und es hängt eher damit zusammen, dass man sagt: "Ich habe eine Idee, ich habe die Information durch die Fachabteilung, ich habe mich ausgetauscht, ich habe das nach ganz gewissen Prinzipien festgestellt." Also Workshops wären zum Beispiel eine ganz wichtige Sache. Aber es gehören auch von dem Citizen Developer ohne weiteres dann Fachinterviews dazu.

EB08\_Fey, Peter: 27 - 27 (0)

Ein Begriff aus dem Linearen und wenn ich jetzt Lessons Learned übersetze, mit Retro, was tatsächlich halt auch von der Frau Derby so bezeichnet worden ist, also einer der klassischen Retromasterinnen, wenn man das so sagen darf. Dann sind viele Elemente, die im linearen Ablauf sind, halt auch im agilen drin. Das Gleiche gilt zum Beispiel für Workshops. Also wir haben auch im linearen

Der Experte hebt als ein wichtiges Tool der agilen Vorgehensmodelle die Möglichkeit in Sprints zu arbeiten hervor, welche Teil des Scrum Methodenmodells sind. Dies kann ebenfalls nützlich für eine spätere Skalierung in andere Abteilungen sein. Ein weiteres Tool, das verwendet wird, ist die Möglichkeit, Retrospektiven durchzuführen. Diese Feedback-Methodik für ein Team wird als sehr wertvoll angesehen und kann sich für die weitere Planung und Vorgehen als positiv auswirken.

EB07\_Janke, Tino

Der Experte hebt die Möglichkeit hervor, Workshops durchzuführen und somit regelmäßiges Feedback und Aufgabenanforderungen auszutauschen.

Er vertritt eine Betrachtungsweise, in der die scharfe Trennung von einigen agilen und traditionellen Methodiken nicht wirklich gegeben ist. Lessons Learned im traditionellen Projektmanagement vergleicht dieser mit dem Tool der Retrospektive im agilen Projektmanagement. Er empfindet Workshops auch als sehr sinnvoll zur Evaluierung von Anforderungsaufnahme und Planung des weiteren Vorgehens. Diese Workshops können in beiden Vorgehensmodellen angewendet werden und sind nicht exklusiv agil oder traditionell. Diese Workshops, die einen Planungscharakter haben, können so auch in agilen Methodiken verwendet werden. Er warnt davor, dass ein zu starres Denken bei der Umsetzung nicht zwingend positiv ist und eher

Bereich ganz viele Workshops durchgeführt, um herauszubekommen: Was machen wir? Was wollen wir tun? Und so weiter und sofort. Das ist da eigentlich drin. Das heißt, die Trennung zwischen agilen und linearen Abläufen, die würde ich da ziehen. Es gibt in diesem Sinne keinen Projektplan. Es gibt ein Ziel, was aber immer wieder adaptiert werden kann auf die neuen Erkenntnisse. Es gibt also diese Möglichkeiten, das zu tun, die sind relativ groß. Es gibt aber auch nicht ein, ich sage es einfach mal in diesen Worten, starren Product Backlog. Es gibt keinen starren Sprint Backlog. Also diese Begrifflichkeiten, die man aus diesem Umfeld des Agilen kennt und die ja eben nicht mit Beliebigkeit zu tun haben, sondern die sehr, sehr stark für 14 Tage oder 3 Wochen, 4 Wochen, die Arbeit eines Teams bestimmen. Der Sprint Backlog, diese starren Vorgaben, die sind da weniger gefragt. Also hier geht es eher darum, in einem laufenden Prozess sich immer wieder auszutauschen. Also es gibt viele Elemente wie Workshops oder halt auch Interviews, die wir im normalen linearen Projektmanagement haben. Den gibt es da drin und die werden notwendigerweise durchgeführt. Viele andere agile Bereiche, wie Sprint Backlog und sonstige Dinge, wie: "Dann muss das fertig sein." oder "Wie weit bist du?" oder Product Owner gibt es in dieser Form nicht. Sondern das ist der Fachbereich selber und da hat vielleicht einer den Hut auf und sagt: "Mensch, wenn wir das machen, dann würde ich das gerne umsetzen" Aber das käme eher aus dem Fachbereich und da gibt es keine großartigen Verbindungen zur Geschäftsführung außer den üblichen, die ein Fachbereich hat und dann wird das abgesprochen. Also das wäre dann in dem Sinne ein dritter Weg, aber eher in der Tendenz zum agilen, zum schnellen Reflektieren darüber, zum schnellen Review: "Wie waren wir?" "Haben wir das richtig gemacht?" "Wollen wir das so haben?" Aber ein Review, der jetzt nicht von außen kommt, vom Kunden, sondern hier ist ja der Erstellende der Kunde selber

nach Bedarf die Tools verwendet werden sollten.

Der Experte skizziert ein Phasenmodell, das in Iterationen durchgeführt wird während der Umsetzung. Dabei werden eine Vorbereitungsphase zuerst durchgeführt, wo für die Entwicklung wichtige Fragen geklärt und erste Lösungsansätze diskutiert werden. Dann kommt eine Phase der Umsetzung mit anschließendem Feedback. Diese 3 Phasen werden immer wieder durchlaufen, bis das Produkt in der gewünschten Qualität vorliegt. Danach wird in eine Testing-Phase übergegangen, wo die IT-Sicherheit und Governance einbezogen werden. Im Falle, das Produkt erfüllt alle Rahmenbedingungen, kann ein Release erfolgen.

EB08\_Fey, Peter

und der würde dann sich selber auf die Arbeit schauen.

EB08\_Fey, Peter: 29 - 29 (0)

Dann die nächste Phase der Umsetzung, die ist jetzt aber nicht viel später, sondern die passiert relativ zeitnah zu dem, was man da erarbeitet hat. Man führt diese preparation-Phasen durch, stellt die richtigen Fragen, überlegt sich die richtigen Lösungen und dann geht man mit dieser Lösungsidee auf denjenigen zu, der sich mit diesem Tool, ich sagte es ja gerade, das muss man dann schon ein bisschen üben. Das ist ähnlich wie bei Power User von Excel, ergibt sich das nicht von alleine. Das heißt, derjenige, der setzt das dann um und zeigt es sofort am Bildschirm und dann kann das erste Feedback schon stattfinden: "Ah, das können wir anders noch machen!" So, und dann geht das von diesem Moment an weiter in diesen iterativen Schleifen, bis man dann sagt: "Ok, jetzt haben wir eigentlich diese Lösung, die wir ganz zum Anfang mal angedacht haben. Die haben wir jetzt in der und der Form realisiert." Und dann geht es in das Testing rein, also, das muss ja, so gerade, wenn man nach außen geht, zum Kunden, muss das ja absolut sicher sein, in das Testing rein und dann muss man natürlich, bevor man das überhaupt auf die Welt loslässt, also "Hello World", wie wir alle früher mal gelernt haben, da muss man dann natürlich auch die ITler da und die IT-Governance befragen. Kommen wir mit diesen Programmierungen, die wir da getan haben, kommen wir damit um die Ecke oder stoßen wir da an Sicherheitsgrenzen. Was müssen wir zusätzliche bedenken? Also das fängt schon so eine Low Code/ No code-Applikation ganz gut ab, aber die, wie gesagt, das ist eine allgemeine Vorgabe, eine allgemeine Funktionalität und die Prüfungen, die müssen tatsächlich dann auch durch die IT-Security erfolgen, also Spezialgebiet, und, wie wir jetzt durch die vielen aktuellen Dinge gelernt haben, wird das immer wichtiger, dass man diese IT-Sicherheit betrachtet.

EB08\_Fey, Peter: 31 - 31 (0)

8 - Tools Vorgehensmodell\8.2 - Tools trad. Vorgehen

Ist natürlich bei dieser initialen Einführung, da müsste man schon mal in den Phasen arbeiten, denke ich, eine Governance aufbauen, Enabling Konzepte aufbauen, aber das würde ich auch nicht, also da würde ich bitte kein Projekt über 1 Jahr draus machen, sondern würde das dann eben...

EB02\_Rosengrün, Robin: 41 - 41 (0)

Der Experte sieht Phasen in der Vorbereitung zum Aufbau von Governance und Enabling als wichtiges Tool aus dem traditionellen Vorgehen an.

EB02\_Rosengrün, Robin

Aber was man bei großen Unternehmen mit viel Geld sehen, das ist noch klassisch verankert, da will irgendjemand, der dann alle Projekte zusammen bündelt, eine Ampel sehen. Da ist natürlich die Frage, also Statusampeln, ist das jetzt agil oder nicht? Da kann man immer drüber streiten, aber da werden andere Kennziffern gemacht. Da wird auch erwartet, dass ein Projektfortschritt gemessen wird, in 85 % ist fertig und solche Sachen. Also das sieht man schon noch, dass dann halt einfach alte Strukturen dazu führen, dass man sich alter Elemente, gerade von Statusberichten, und so was bedient.

EB04\_Göttl, Simone: 37 - 37 (0)

Die Expertin hebt hervor, dass in großen Unternehmen, mit viel Kapital im Hintergrund, der Wunsch nach Fortschrittsmessung besteht. Dies führt sie auf veraltete Strukturen zurück.

EB04\_Göttl, Simone

Also insgesamt in der Projektarbeit, weil man davon ausgehen muss, dass die Kollegen, die das machen, dann tatsächlich diesem Daily Business verhaftet sind, ihren täglichen Routinen. Also da fehlt die Projekterfahrung, deswegen wäre es eher ein Waterfall-Ansatz. Dass es jemanden gibt, der so ein bisschen den Rahmen festlegt und die nächsten Schritte, aber und das kommt dazu vom hybrid her, durchaus hybride Teile einstreut. Das ist, glaube ich, das was ich sehe, was sich bewährt.

EB05\_Rohrbeck, Christian: 39 - 39 (0)

Der Experte vertritt die Auffassung, dass ein primärer Wasserfallansatz mit allen Tools verwendet werden sollte. Dabei sollten auch agile Tools mit zur Anwendung kommen. Dies führt er auf die mangelnde Projekterfahrung zurück.

EB05\_Rohrbeck, Christian

Bei Project Management Institute zum Beispiel heißen die ersten beiden Projektphasen Initiierung und Planung, dass man sich einfach anschaut: Wer ist denn ein geeignetes Team, um dieses Projekt anzugehen, sind die wirklich alle motiviert dazu? Ich muss mir in der Phase auch anschauen, wer sind die Stakeholder? Mit wem muss ich wie umgehen

Der Experte sieht vor allem zu Beginn den Bedarf für Phasen aus dem klassischen Vorgehen, um eine gewisse Planungssicherheit zu erzeugen und die Rahmenbedingungen zu klären. Diese gemeinsame Zielsetzung und klare Aufgabenverteilung soll dazu führen, dass ein gemeinsames Projektcommitment erzeugt wird.

während der Projektlaufzeit und ich muss auch mit allen Beteiligten den Scope klar abgrenzen, also es muss sozusagen klar sein, wo die Reise hingehen soll. Dass, ja, dass es eben dazu führt, dass eben alle an einem Strang ziehen.  
EB07\_Janke, Tino: 35 - 35 (0)

Ja, natürlich gibt es da Phasen. Also diese Phasen, wird von dem Vorgehen der PMI, sind im Endeffekt ähnlich wie bei allen anderen, auch bei agilen Projekten.  
EB08\_Fey, Peter: 31 - 31 (0)

Der Experte sieht die Notwendigkeit von Phasen als gegeben an und ergänzt, dass diese auch in agilen Projekten vorhanden sind.  
EB08\_Fey, Peter

Das ist sozusagen der erste Prototyp, wo man sich dann wieder trifft, aber nicht nach vier Wochen, sondern vielleicht schon mal nach zwei Tagen. Bei manchen Meetings ist es auch so, die einen setzen sich dabei in die Ecke und beantworten ihre E-Mails und trinken eine Tasse Kaffee dabei. Und danach gehen sie zurück und dann ist es in dem Meeting schon, dass der Citizen Developer dann sagt: "Ich hab mal was vorbereitet. Geht das in die richtige Richtung?" Also diese Phasen der preparation, der Betrachtung, die sind da  
EB08\_Fey, Peter: 31 - 31 (0)

9 - Änderungswünsche\9.1 - Changemanagement agiles Vorgehen

**B: EH** Wie gehe ich mit Änderungswünschen um. Ja, so wie man klassischerweise im agilen damit umgeht. Man nimmt den Änderungswunsch auf, egal woher er kommt und bespricht mit den Leuten, die sozusagen für diese eigene Low Code Plattform-Einführung tatsächlich zuständig sind und den technischen Umsetzern vielleicht, ob das eine sinnvolle Änderung ist oder nicht. Also im Scrum würde man jetzt sagen, ich bespreche mit dem Product Owner, der für Einpriorisierung zuständig ist und ich nehme mir dann aber noch die Entwicklerin, also in meinem Fall dann die IT vielleicht dazu, um zu gucken, ist das technisch eine sinnvolle Geschichte. Vielleicht muss ich nochmal bei manchen Änderungswünschen nochmal den Datenschutz machen. Also ich mache dann letzten Endes so eine Bewertung der

Mit den Änderungswünschen würde der Experte gleich umgehen, unabhängig davon, ob die Änderung aus dem Projektteam oder von außerhalb des Projektes kommt. Dabei wird der Änderungswunsch mit den Projektteilnehmern, welche bei dem Themenkomplex involviert waren, abgesprochen und bewertet. Wird dieser als sinnvoll angesehen, wird er in den Product Backlog aufgenommen und später in die Sprintplanung.

EB01\_Eckhard Hauenherm



Änderung. Irgendwo muss ja einer mal darüber gucken und sagen, macht das Sinn, das so zu machen. Es ist eigentlich egal, ob sie von innen oder von außen kommt. Und dann wird sie halt in der Regel relativ zeitnah im Projekt umgesetzt, das heißt, typischerweise in die Sprintplanung mit aufgenommen.

**I: RS** Und aber vorher ins Backlog genommen?

**B: EH** Ja.

EB01\_Eckhard Hauenherm: 55 - 57 (0)

Also, wenn wir mit klassischen, mit agilen Projekt vorgehen, ist jede Änderung, die reinkommt, wird auch so übers Changemanagement abgewickelt, dass wir sagen: "Die kommt ins Backlog mit rein." Und dann wird diese Änderung gegen das bisherige Backlog gemessen. Also wenn die wichtig ist, dann rutscht die entsprechend weiter nach vorn, dann fällt halt hinten irgendwas anderes runter in der Geschichte. Ganz klassisches Changemanagement. Was ein bisschen schwierig wird in einer Plattform-Einführung in dem Stream ist, dass ich eben nicht nur Applikationen baue, sondern ich habe ja mehrere Streams, die parallel agieren, wenn ich da irgendwelche Changes habe, die mehrere Streams betreffen, ist die Koordination natürlich teilweise tricky. Wenn jetzt, zum Beispiel, irgendwelche Regulatoren sich ändern, die jetzt eben nicht nur meine Entwicklung betreffen, sondern alle fünf Streams die ich habe, muss ich halt gucken, wie kann ich den jetzt sauber einbauen, dass der Change auch tatsächlich dann irgendwie parallel in verschiedenen Streams umgesetzt wird. Ansonsten nützt es mir nix, wenn meine Applikationen das jetzt schon umgesetzt haben, mein Changemanagement aber frühestens in sechs Monaten dazu kommt. Also die Priorisierung muss dann halt Streamübergreifend erfolgen, wenn der Change das benötigt. Das macht es dann etwas komplexer, wäre aber in jedem normalen Umsetzungsprojekt auch so.

Der Experte verfolgt beim Thema Changemanagement eine klassische Scrum-Logik. Ein Change wird in den Backlog übernommen und mit den anderen Backlog-Aufgaben abgeglichen und priorisiert. Das wird im entsprechenden Scrum-Team bearbeitet. Eine größere Herausforderungen ist, wenn die Änderung verschiedene, parallel arbeitende Scrum-Teams betrifft und eine hohe Priorität hat. Dort sollte eine Möglichkeit gefunden werden, diese mit der entsprechenden Priorität in die Backlogs hinzuzufügen, ohne dabei zu große Probleme in den Teams zu verursachen. Dies wird als komplexere Problemstellung durch den Experten wahrgenommen.

EB03\_Fabian Gackstatter

**I: RS** Also nur noch mal kurz, um es zusammenzufassen. Es ist die ganz klassische Scrumlogik, wie ihr damit umgeht.

**B: FG Ja**

EB03\_Fabian Gackstatter: 51 - 53 (0)

Wir arbeiten auch agil. Die Änderungswünsche werden in den entsprechenden Backlog überführt bzw. mit den bestehenden Backlog Items abgeglichen. Also entweder es ist eine neue Anforderung oder es ist eine geänderte Anforderung oder auch der Wegfall der Anforderung. Das wird entsprechend im Backlog vermerkt und dann findet das wieder Einsortierung in den regulären agilen Fluss, in das agile Projektmanagement. Also Bewertung der Items ziehen in den aktuellen Sprint und Umsetzen.

EB06\_Hesse, Moritz: 61 - 61 (0)

Der Experte favorisiert ein agiles Modell und würde bei Änderungswünschen dem klassisch agil Vorgehen den Vorzug geben.. Eine Änderungsanfrage würde in den Backlog aufgenommen und mit den bestehenden Aufgaben im Backlog abgeglichen und durch das Scrum-Team einsortiert.

EB06\_Hesse, Moritz

9 - Änderungswünsche | 9.2 - Changemanagement | hybrides Vorgehen

Die Änderungswünsche sind immer, also, das ist meistens nichts, was in 1 Woche umzusetzen ist, ist das Thema, weil wir uns da tief in IT-Strukturen befinden, jetzt zum Beispiel auf unsere Bestellsysteme so eine Lizenz drauf zu bekommen. Das ist wieder ein größeres Projekt. Das läuft dann über Wochen, viel Abstimmungsbedarf mit anderen Abteilungen, oder das andere Beispiel, was ich genannt hatte, auch mit Datenschutz. Alles gilt zusammenzuarbeiten, das ist auch nichts, was in 1 Woche geht. Das dauert und so, ja, lässt sich das auch schlecht in Sprints gießen. Wenn wir aber, ja, wenn wir dann tatsächlich auch Sachen entwickeln, wir entwickeln auch Sachen auf der Low Code-Plattform, für das Low Code-Plattform-Management oder so bietet auch Microsoft da schon Tools an, dann versuchen wir da, bis es mal initial implementiert ist, schon auch in einer Sprint-Logik so ein bisschen zu arbeiten, also definieren von Arbeitspaketen und dann 2-Wochen-Rhythmus oder so was, da dann Fortschritte zu sehen, wieder neu releasen, erste Features zu releasen, aber machen wir nicht, brauchen wir nicht durchgehend, dafür ist unsere Arbeit auch zu unterschiedlich,

Änderungswünsche werden in eine längerfristige Planung aufgenommen. Bei der Bearbeitung wird versucht, ein Sprintvorgehen anzuwenden, dies ist aber nicht immer möglich aufgrund der Größe des Teams und der entsprechenden unterschiedlich gelagerten Aufgaben. Die Abarbeitung erfolgt in der Regel über ein Ticketsystem, bei dem sich ein Verantwortlicher die Aufgabe nimmt und bearbeitet.

EB02\_Rosengrün, Robin

also dass wir in dem Team auch zu unterschiedliche Aufgaben haben, von Governance hat, sage ich jetzt einfach mal auch sehr viele. Die haben ein Postfach und arbeiten dann quasi Tickets ab von Sachen, also ein großer Teil, den die da haben, wo man jetzt auch keine agile Logik für braucht. Ja, und ich plane einfach viele, viele Schulungen oder so was mit meinen Kollegen. Wo es auch wieder sehr viel Tagesgeschäft einfach ist. Mittlerweile sind wir jetzt in der Phase angekommen.

EB02\_Rosengrün, Robin: 49 - 49 (0)

Also erstens: in Backlog aufnehmen. Das ist, glaube ich, der erste Punkt und dann ist natürlich ein zweiter Punkt. Ich habe mein Backlog ja vorab priorisiert bzw. priorisiere das immer wieder innerhalb eines Sprints und dann ist natürlich der Punkt, ist es etwas, was eine hohe Priorisierung hat, weil es ein wichtiges, ein kritisches Item ist, weil es vielleicht auch für den Endkunden, für den Nutzer, wichtig ist? Und das ist natürlich ein Punkt, immer wieder zu priorisieren, die Punkte aufzuwerten und das auch dem Scrum-Team dann dort auch zu überlassen, was in das Backlog kommt und dort mit den Stakeholdern auszutauschen. Was natürlich nicht passieren soll, ist, dass jemand einfach reingrätscht und sagt, das ist jetzt mein Killerfeature und den ganzen Plan durcheinander bringt. Das ist eigentlich nicht vorgesehen und wer ist da in der Rolle? Der Scrum Master ist da definitiv in der Rolle, die Methodiken einzuhalten und dafür zu sorgen, dass das Sprintteam auch adäquat arbeiten kann. So, das ist ein wichtiger Punkt und dort das dann auch zu leben. Wir wissen alle, dass irgendwie Theorien gut funktionieren, wenn jetzt so eine, ich bleibe mal beim Datenschutz, wenn eine EU DSGVO erst nicht aktiv ist und während so einem Einführungsprojekt reinfliegt, weil es plötzlich wichtig ist, dann hat man natürlich ein Risiko, dass man etwas nicht rechtlich Konformes einführen würde und das ist natürlich eine andere Priorisierung vielleicht, als wenn irgendwie ein Feature, ob eine Funktionalität,

Die Expertin favorisiert ein hybrides Vorgehen, allerdings bei Änderungswünschen am Projekt wird ein agiler Ansatz verfolgt. Bei Änderungswünschen würden diese in den Backlog eingepflegt und es würde durch das Scrum-Team bewertet, ob es eine kritische Änderung ist oder eher mit einer niedrigeren Priorisierung in das Backlog eingepflegt wird. Die Expertin weist auf die Problemstellung hin, dass es passieren kann, dass es nicht passieren sollte, dass eine Person die Priorisierung alleine bestimmt. Zu diesem Zweck ist der Scrum Master sehr wichtig, welcher auf die Einhaltung der Methodik achten muss.

EB04\_Görtl, Simone

wie kann ich jetzt mobil arbeiten oder nicht mobil arbeiten. Kann ich Externe hierzu einladen oder nicht, das ist was ganz anderes, das hat eine ganz andere Kritikalität.

EB04\_Göttl, Simone: 43 - 43 (0)

Ich überlege gerade, ob es einen Unterschied macht, von wem eine solche Änderung herangetragen wird. Also ich sage mal, ob es einen Unterschied macht, ob es von intern kommt oder von extern. Ich glaube, es kommt gar nicht so sehr auf die Quelle drauf an, sondern auf die Art der Veränderung. Es gibt natürlich schwerwiegende Veränderungen, wo man im Prinzip alles umstoßen kann, als auch kleinere. Ich glaube, wichtig bei jeder Veränderung ist einmal, die erst mal zu analysieren. Was ist der Impact? Wenn man, ich sage mal, an das Projektteam herantritt, da ist eine Änderung, da fragt sich natürlich jeder: „Was machen wir jetzt damit?“ Das muss klar sein, dass man sich einen alternativen Weg klarmacht. Das kann jetzt je nach Methode einfach mal im Leitungsrahmen sein oder im Team, dass man sagt: "Hör zu, hier ist eine Änderung. Lasst uns darüber diskutieren über die Auswirkungen." Dass man diese Veränderung, wie auch immer, zusammen erarbeitet und die Konsequenzen klar auf dem Schirm hat. Zeitlich und so weiter, personell zum Teil auch, wenn man neue Leute braucht für neue Sachen. Und sobald das klar ist, sollte man das zurückspiegeln an denjenigen, der die Veränderung, den Change, initiiert hat und ihm sagen: "Hör zu, das sind dann die Konsequenzen." Eventuell, je nach Projektform, dann auch mit einem Angebot, Änderungsangebot. In der Regel führt das zu einer Verzögerung. Das auch klar kommunizieren.

EB05\_Rohrbeck, Christian: 55 - 55 (0)

Es kommt tatsächlich beides vor und das hängt an der Gewichtigkeit der

Der Experte vertritt die Ansicht, dass ein hybrides Modell beim Vorgehen die beste Variante ist. Dabei ist er der Meinung, dass es nicht wichtig aus welcher Quelle die Veränderungsanforderung kommt, sondern welche Art und Auswirkung die Änderung hat. Bei Veränderungsanforderungen sollte zuerst analysiert werden, welchen Einfluss die Änderung auf das Gesamtprojekt hat. Nach der Bewertung durch das Projektteam sollte eine klare Kommunikation zum Anforderer erfolgen und die Auswirkungen aufgezeigt werden, da in der Regel Änderungsanforderungen zu einer Verzögerung führen, was darin bedingt ist, dass der Experte ein Modell favorisiert, das auf Wasserfallbasis funktioniert, mit agilen Anteilen. Je nach Intensität der Änderung unterscheidet er jedoch zwischen leichten Änderungen, welche mit in das Backlog des Projekts aufgenommen werden oder schwerwiegenden Änderungen, wie dem Ausfall eines Projektmitglieds. Im Falle einer Änderung mit hoher Intensität empfiehlt er den Weg über ein Change Request und einer entsprechenden neuen Planung im Projekt, welche die Auswirkungen der Änderung berücksichtigt.

EB05\_Rohrbeck, Christian

Änderung. Natürlich würde ich Änderungen, die im normalen Rahmen sind, einfach ins Backlog packen. Wenn bestimmte Sachen, das machen wir auch hier, wenn wir sagen, bestimmte Sachen können wir in dieser Etappe nicht durchführen, dann wird das ins Backlog zurückgeschrieben und dann ist das Thema durch. In meinem aktuellen Projekt habe ich auch mit größeren Sachen zu kämpfen. Größere Sachen sind für mich Fluktuation. Hier ist eine Kollegin, die bekommt bald ein Kind, da freue ich mich sehr für sie. Sie ist aber trotz allem spontan ausgefallen und wir hatten keinen Ersatz, weil unser Team so klein ist. Und das war jetzt eine Sache, da mussten wir wirklich eine große Welle schieben, weil diese ganze Veränderung bedeutet auch, dass, weil diese Kollegin auf dem kritischen Pfad unterwegs war, dass wir jetzt direkt eine Verzögerung haben, bis der neue Kollege eingearbeitet ist. Und das bedeutet in dem Fall, jetzt mal 2 Monate, von jetzt auf gleich. Und da muss man im Prinzip die entsprechenden Stellen informieren und eine Art, na ja, nicht Change Request in dem Fall, aber so eine Art Verzögerungsmeldung, mit dem Impact der zeitlichen Verzögerung, herausgeben. Also auch hier hat man, zumindest wenn man diesen hybriden Ansatz fährt, alle Optionen des klassischen als auch des agilen Managements dann zur Verfügung. Und das macht es wirklich sehr angenehm, weil man dann sehr, sehr angemessen auf den jeweiligen Change reagieren kann.

EB05\_Rohrbeck, Christian: 57 - 57 (0)

Also, ich würde das, glaube ich, mit dem Team zum einen besprechen, das Team muss es ja vertreten. Die Änderungswünsche, es muss sie umsetzen, das heißt, das Team muss da einfach dabei sein, dass das eine sinnvolle Änderung ist. Und ich würde auch immer den Abgleich machen mit dem Ziel, was ich ursprünglich mal für das Projekt ausgegeben habe, sind wir denn noch auf dem richtigen Weg? Also so ein bisschen dieser Blick zurück, schaffen wir es, mit

Der Experte vertritt ein hybrides Modell und hebt die Bedeutung von Projektteam und Zielstellung, in Bezug auf Änderungsanforderungen, hervor. Er beschreibt die Bedeutung von Kommunikation mit dem Projektteam, da dieses die Änderung umsetzen und auch nach außen vertreten muss. Als weiteren wichtigen Punkt sieht er den Faktor, dass eine Änderung dem Projektziel zuträglich sein muss. Also ist die

dieser konkreten Änderung jetzt, dem ursprünglichen Projektziel näherzukommen? Schaffen wir Mehrwert für die Organisation? Das wären so die Fragen oder die beiden Seiten, die ich da beleuchten würde, also einmal: Das Team muss da dabei sein und das mit vertreten und auf der anderen Seite muss ich dem ursprünglichen Ziel durch die Änderung näherkommen und wenn das beides passt, dann kann man das gerne mit in die Umsetzung nehmen, die Änderungswünsche.

EB07\_Janke, Tino: 37 - 37 (0)

Änderung mit den Projektzielen vereinbar oder steht sie dem entgegen.

EB07\_Janke, Tino

Genau. Also, wenn man diesen Backlog dann so definiert, als einige Karten, die ich zum Beispiel in ein Kanbanboard einsetze, dann wäre das eine Möglichkeit, das wäre eine Ebene zu sagen: "Ich als User hätte gerne und dann so und so, kann man es umsetzen. Kannst du das machen?" Das ist richtig, aber das würde dann innerhalb des Projektes, innerhalb der Fachabteilung, wie ich es gerade beschrieben habe, laufen. Also da gäbe es dann ein Backlog, sozusagen, des Fachbereichs für den Fachbereich und der würde es dann umsetzen und würde es direkt zeigen können auch.

EB08\_Fey, Peter: 39 - 39 (0)

Der Experte vertritt eine Art hybrides Modell mit starkem Fokus auf agile Vorgehensweisen. Änderungswünsche werden in den Product Backlog des Projekts aufgenommen, dafür sieht er das Kanban als eine geeignetes Tool zur Visualisierung.

EB08\_Fey, Peter

10 - Rollen und Stakeholder im Projekt\10.1 - Projektverantwortlicher

Also die Rollen, unter den Namen brauche ich sie nicht unbedingt, aber ich brauche natürlich einen, der tatsächlich sozusagen dahintersteht, dass No Code/ Low Code eingeführt wird. Der das will. Also, so eine Art Product Owner, der sagt: "Ok, das ist für mich wirklich eine Applikation." Oder "Es ist für mich ein Ziel, das ich erreichen möchte. Ich habe da eine Vision, das irgendwie im Unternehmen zu nutzen." Im klassischen Projekt wäre das sowas wie ein Sponsor oder sowas, der dann entsprechend dahintersteht. Bei so einem agilen Projekt entspricht das der Rolle eines Product Owners, würde ich sagen.

EB01\_Eckhard Hauenherm: 59 - 59 (0)

Der Experte sieht die Rolle eines Projektverantwortlichen im Unternehmen als wichtig an. Diese Rolle sollte die Projektzielstellung definieren und die Anwendung im Unternehmen vorantreiben.

EB01\_Eckhard Hauenherm

Also schon einen Product Owner, Low Code-Plattform hätte ich gerne, hatten

Der Product Owner wird als wichtig angesehen. Diese Rolle sollte eine

wir nicht, hätte ich gerne, der über alle Themen so ein bisschen darüber guckt und danach guckt, dass die Fäden alle richtig zusammen laufen.

EB02\_Rosengrün, Robin: 51 - 51 (0)

verantwortliche Position vertreten und das Projekt voranbringen.

EB02\_Rosengrün, Robin

Du hast ja jetzt schon Rollen angesprochen. Es ist ja ein Scrum-ähnliches. Hast du dann trotzdem diese gleichen Rollen: Scrum Master, Product Owner, das Developmentteam?

**B: FG** Ja die nutzen wir auch.

EB03\_Fabian Gackstatter: 46 - 47 (0)

Der Product Owner wird als wichtig angesehen.

EB03\_Fabian Gackstatter

Ich würde auch nicht mehr von einem Projektleiter sprechen, sondern von einem Product Owner für diese Plattform und das ist im Idealfall der, der sie auch später dann in der Linie, wenn es eingeführt ist, übernimmt.

EB04\_Göttl, Simone: 31 - 31 (0)

Product Owner als verantwortliche Rolle wird als wichtig angesehen. Dieser sollte gleichzeitig auch die Rolle der Projektorganisations übernehmen.

EB04\_Göttl, Simone

Das wäre quasi auch meine nächste Frage gewesen, welche wichtigen Stakeholder Sie im Projekt sehen. Bloß aber noch mal zur Klarstellung, aber innerhalb des Projektes sehen Sie die ganz klassischen Scrumrollen wie Scrum Master, Product Owner, das Development-Team, einen Sponsor, das sind so die Rollen, die Sie sehen?

**B: SG** Genau.

EB04\_Göttl, Simone: 46 - 47 (0)

Also definitiv die Rolle des Sponsors, ist natürlich ganz klar. Ohne Finanzierung geht es in der Regel nicht. Was ich als essenziell ansehe, neben dem Team natürlich, als in dieser Teamrolle ist ein Product Owner. Jemanden, der sich für das, was man entwickelt, dann auch verantwortlich fühlt und vor allen Dingen, der auch in der Lage ist zu sagen, wie er sein Produkt gestalten möchte. Also eine relativ klare Ansage. Weil, man kommt in der Entwicklung immer wieder an diese Entscheidungswege oder Abzweige, wo man sagt: "In welche Richtung soll es jetzt weitergehen?" Und hier eine kompetente Person zu haben, die sagt: "Ich möchte das genau so haben" ist dann

Die Rolle eines Sponsors/ Product Owner wird als sehr wichtig angesehen. Diese Rolle sollte sich um die Finanzierung des Projekts kümmern. Der Product Owner sollte sich für das Projekt verantwortlich fühlen und eine Zielstellung vorgeben, was der Output des Projektes sein sollte. Im Verlauf von Projekten müssen Entscheidungen getroffen werden und dort wird der Product Owner als verantwortlich betrachtet.

EB05\_Rohrbeck, Christian

von absoluter Entscheidung oder entscheidender Bedeutung. Der Product Owner spielt für mich eine zentrale Rolle.

EB05\_Rohrbeck, Christian: 59 - 59 (0)

So ein Project Owner, Product Owner im Sinne eines, nicht der fachliche Projektleiter, sondern im Sinne eines Auftraggebers, eines Sponsors. Den habe ich natürlich auch.

EB06\_Hesse, Moritz: 65 - 65 (0)

Ja, also beim Sponsor bin ich mir unsicher, ob er zum Projektteam gehört oder nicht. Auf jeden Fall ist der Sponsor auch eine wichtige Rolle, weil er das Thema, glaube ich, auch noch mal auf einer anderen Ebene vertritt. Also wenn man sich vorstellt, dass der Sponsor regelmäßig aus einer anderen Hierarchieebene kommen wird. Das heißt, er bewegt sich auf einer anderen Hierarchieebene und in der Organisation und kann das eben auf dieser höheren Ebene auch besser vertreten als zum Beispiel der Projektleiter. Aber ansonsten muss der Sponsor, glaube ich, offen sein für das Neue, da ist wieder dieses Thema Mindset. Das muss er auf jeden Fall mitbringen und ob er dann dafür noch aktiv Werbung macht für dieses Thema oder nicht? Das würde ich jetzt dem Sponsor überlassen. Ich glaube, viel Wert ist einfach, wenn dieses buy in vom Sponsor da ist, dass der das Projekt unterstützt und sich da auch so ein bisschen dahinter stellt, genau, wie aktiv der jetzt wird, das möchte ich gar nicht sagen. Ich glaube, da gibt es unterschiedliche Ansätze. Es genügt auch, wenn der Sponsor das Budget gibt und am Anfang in die Verantwortung geht und dann kann, glaube ich, auch die Projektleitung viel, viel ziehen und viel leisten.

EB07\_Janke, Tino: 41 - 41 (0)

Product Owner haben wir in diesem Sinne nicht. Einen Project Owner wäre ganz gut, wenn man den hat. Also das ist ja dann der Sponsor, der sagt: "Ja ich stehe dafür gerade, dass ihr zweimal in der Woche einen Workshop von vier Stunden machen könnt." Und das wird ja

Wird als wichtig angesehen, im Sinne eines Auftraggebers.

EB06\_Hesse, Moritz

Der Experte sieht die Rolle des Sponsors nicht zwingend im Projekt. Diese sollte jedoch vorhanden sein. Der Sponsor sollte von einer höheren Hierarchieebene kommen und sollte das Projekt in der Organisation vertreten. Der Sponsor sollte Verständnis für das Thema haben und ein offenes Mindset besitzen. Der Sponsor wird als ressourcenverantwortliche Rolle gesehen.

EB07\_Janke, Tino

Der Experte sieht eine projektverantwortliche Rolle als wichtig an.

EB08\_Fey, Peter



	<p>nicht über jahrzehntelang laufen, aber während der Entwicklungszeit wäre das zum Beispiel eine ganz gute Sache, dass man da zwei Vormittage, zwei Nachmittage einfach mal an dieser Sache arbeitet gemeinsam als Fachbereich, das wäre ganz gut.</p> <p>EB08_Fey, Peter: 45 - 45 (0)</p>	
<p>10 - Rollen und Stakeholder im Projekt\10.2 - Technische Rollen</p>	<p>Worauf ich immer Wert lege ist, dass das Team selber gemischt ist. Jetzt nicht im Sinne von Rollen, aber dass ich Technik dabei habe, also IT dabei habe, dass ich Business dabei habe. Dass ich zumindest Zugriff auf sozusagen Datenbewertung, Datenschutz, also irgendwie juristisches Know-how irgendwo mit im Team habe.</p> <p>EB01_Eckhard Hauenherm: 59 - 59 (0)</p> <p>Ich habe meine Datenschutzexpertise mit drin. Ich habe die IT, die für die technische Umsetzung der Governance, zum Beispiel, zuständig ist, für die Wartung der Plattform, Pflege der Plattform, die muss mit drin sein. Und ich habe, wie gesagt, das Business, eventuell mit wechselnden Teammitgliedern, je nachdem, mit welchem Geschäftsbereich wir gerade arbeiten, mit drin.</p> <p>EB01_Eckhard Hauenherm: 61 - 61 (0)</p>	<p>Es wird durch den Experten hervorgehoben, dass es wichtig ist, ein gemischtes Team zu haben. Darunter wird aufgefasst, dass die IT-Abteilung Teil des Projektteams ist. Dies sichert den Zugang zu Informationen für Datenbewertung und Datenschutz. Weitere Aufgaben für diese Projektmitglieder sind die technische Umsetzung von Governance-Richtlinien, Wartung und Pflege der Plattform.</p> <p>EB01_Eckhard Hauenherm</p>
	<p>Dann habe ich meinen String des Developer enablements und hier trainiere ich jetzt tatsächlich die Kundenressourcen und auch die Citizen Developer, bereite Applikationen vor.</p> <p>EB03_Fabian Gackstatter: 29 - 29 (0)</p> <p>Und als Letztes den Bereich die Supportorganisation. Wie verknüpfe ich jetzt meine neue Plattform, meine Citizen Developer, mit den entsprechenden internen Supportstrukturen.</p> <p>EB03_Fabian Gackstatter: 29 - 29 (0)</p>	<p>Der Experte hat auf der einen Seite technische Rollen mit im Projekt für die zugrunde liegende Support-Strukturen und um die Plattform mit diesen Strukturen zu verknüpfen.</p> <p>Gibt es Applikationsentwickler aus der IT-Abteilung, werden diese ebenfalls mit im Team gesehen für eine frühzeitige Befähigung.</p> <p>EB03_Fabian Gackstatter</p>
	<p>Keine Ahnung, ein großer DAX-Logistikdienstleister oder so was, möchte uns einsetzen, dann bin ich schon darauf angewiesen, dass dort mir entsprechenden</p>	<p>Der Experte benötigt die IT-Abteilung mit im Team für die Umsetzung der Plattform und die Anbindung an die technischen Systeme zu haben.</p>

	<p>IT-Leute an der Seite sitzen, die den Counterpart, also alle Gegenseiten beraten können, also ich kann ja immer nur meine Seite beraten, ich kann ja immer nur meine Schnittstelle beraten. In die Kundenseite kann ich mich quasi nur reinarbeiten. Also am besten ist da jemand vor Ort, der mich beraten kann. Also quasi der technische Counterpart. EB06_Hesse, Moritz: 65 - 65 (0)</p> <p>Ich würde behaupten, wirklich essenziell ist der Citizen und, je nachdem ob ich technische Umsysteme integrieren muss, auch die IT Counterpartseite. EB06_Hesse, Moritz: 65 - 65 (0)</p>	<p>EB06_Hesse, Moritz</p>
<p>10 - Rollen und Stakeholder im Projekt\10.3 - andere Rollen</p>	<p>Also, es muss mindestens eine Person mit Erfahrung reinkommen. Natürlich, die das Ganze ein bisschen steuert. Ob extern oder intern ist da egal, es muss von Anfang an eine Governance Struktur mit dabei sein, also Administrierung von der Plattform muss es dafür geben und muss auch eine eigene Rolle sein, die nicht unbedingt, also es muss kein Entwickler sein, sondern dieser Person muss dafür eine gewisse Zeit eingeräumt werden. EB02_Rosengrün, Robin: 29 - 29 (0)</p>	<p>Als eine weitere Rolle im Team wird ein Berater gesehen, welcher bereits Erfahrung mitbringt. Dieser sollte bei der Administrierung der Plattform und dem Aufbau der Governance unterstützen. Diese Rolle muss nicht zwingend ein Entwickler sein. EB02_Rosengrün, Robin</p>
	<p>Also was ganz wichtig ist. Ich weiß nicht, ob es Rollen oder als Position noch im Unternehmen ist. Das sind ja IT-Einführungsprojekte, das heißt, es ist einerseits wichtig, so eine Compliance Abteilung für das Thema Datenschutz mit einzubeziehen, die Informationssicherheit, was auch zulässig ist und, nicht zu guter Letzt, das sollte man in Konzernen nie vergessen, das ist der Betriebsrat oder die Personalversammlung. Das heißt, all das beeinflusst ja Mitarbeiter und da hat der Betriebsrat auch ein Mitsprachrecht und auch solche Tools müssen in so eine Konzern- oder einer Betriebsvereinbarung aufgenommen werden. Das heißt, diese Personen oder Personengruppen mit zu involvieren von Tag eins, ist unheimlich wichtig, weil, am Ende können das die Show Stopper sein, dass ein Tool nicht ausgerollt wird. Habe ich auch schon erlebt. Es war bei einer MS Teams- Einführung bei einem Kunden</p>	<p>Die Expertin hebt hervor, dass eine wichtige Person im Team aus der Compliance-Abteilung sein sollte, um bei den Themen Datenschutz und Informationssicherheit zu unterstützen. Als weiteres wichtiges Teammitglied wird die Personalvertretung der Mitarbeiter gesehen, sofern diese vorhanden ist. Die Einführung einer NCLC-Plattform beeinflusst die Mitarbeiter, weshalb dort ein Mitspracherecht besteht. Im Ernstfall könnten diese das Projekt blockieren. EB04_Göttl, Simone</p>

hat sich der Rollout zwei Wochen verzögert, weil es keine Freigabe des Betriebsrats gab. Und das ist, glaube ich, etwas, was außerhalb dessen ist, was man vielleicht im Framework explizit sieht. Es ist vielleicht ein Stakeholder, aber ein Stakeholder, der halt nicht als Stakeholder bezeichnet werden sollte, sondern eine sehr wichtige Rolle einnimmt.

EB04\_Göttl, Simone: 45 - 45 (0)

Dann gibt es aber auch den Citizen Developer Strategist. Der Strategist hat die Aufgabe, aus Unternehmenssicht heraus zu sagen: "In dieser und jener Form wollen wir diese Applikationen aufbauen. Da und darauf sollen die ja einzahlen." Also, auch wenn die No Code/ Low Code-Applikationen durch den Fachbereich betrieben werden können, ist das keine App für 14,90 €, sondern das sind veritable Investitionen, die ein Unternehmen leisten muss, die sind teuer. Also das muss man ganz klar sagen.

EB08\_Fey, Peter: 43 - 43 (0)

Der Experte hebt die Rolle des Citizen Developer Strategists hervor. Dieser handelt nicht aus Sicht der Fachabteilungen, sondern aus Sicht des Unternehmens, welche Applikationen sollen entwickelt werden, wofür sind diese gedacht.

EB08\_Fey, Peter

10 - Rollen und Stakeholder im Projekt\10.4 - Enabler

Das ist gerade am Anfang, bis alle Prozesse stehen, eine zeitraubendere Geschichte und es muss ein Enabling geben. Also muss Leute wie mich geben, die einfach mit Rat und Tat zur Seite stehen, weil es sonst ein sehr zäher Prozess auch wird. Das reicht dann von: Man muss den Leuten zeigen, was sie damit machen können, als man muss Demos zeigen, was damit geht, man muss die Leute befähigen, das heißt kurze Schulung, man muss einen kurzen Draht zu dem Mensch haben können. Also ganz klassisch, manche Leute beißen sich einen Tag lang an einer Formel die Zähne aus, die ich wahrscheinlich in einer Minute runtertippen könnte. Da hilft es, also ich habe da immer ein offenes Ohr für die Leute, die können sich direkt an mich wenden und ich versuche schnell zu helfen und durch die Rolle von einem Enabler haben wir auch eine Community etabliert,

EB02\_Rosengrün, Robin: 29 - 29 (0)

Der Experte hebt die Rolle des Enablers hervor. Dieser erfüllt die Funktion nahe an den Applikationsentwicklern zu sein und diese bei den Anwendungen zu unterstützen, welche teilweise schon während der Einführung erstellt werden.

EB02\_Rosengrün, Robin

Dann habe ich den String des Platform enablements. Also wie muss die

Die Rolle des Enablers wird durch den Experten definiert, als eine Person oder

	<p>Plattform tatsächlich organisiert sein? Welche Rollen brauche ich? Welche technischen Capabilities müssen gegeben sein? Welche Policies müssen im Unternehmen eingeschaltet sein?</p> <p>EB03_Fabian Gackstatter: 29 - 29 (0)</p>	<p>Personengruppe, welche die Plattform am Ende organisiert, die benötigten Rollen evaluiert, welche technischen Beschränkungen sind vorhanden und welche Rahmenbedingungen müssen etabliert sein.</p> <p>EB03_Fabian Gackstatter</p>
<p>10 - Rollen und Stakeholder im Projekt\10.5 - Governance Rolle</p>	<p>Also, es muss mindestens eine Person mit Erfahrung reinkommen. Natürlich, die das Ganze ein bisschen steuert. Ob extern oder intern ist da egal, es muss von Anfang an eine Governance Struktur mit dabei sein, also Administration von der Plattform muss es dafür geben und muss auch eine eigene Rolle sein, die nicht unbedingt, also es muss kein Entwickler sein, sondern dieser Person muss dafür eine gewisse Zeit eingeräumt werden. Das ist gerade am Anfang, bis alle Prozesse stehen, eine zeitraubendere Geschichte und es muss ein Enabling geben. Also muss Leute wie mich geben, die einfach mit Rat und Tat zur Seite stehen, weil es sonst ein sehr zäher Prozess auch wird. Das reicht dann von: Man muss den Leuten zeigen, was sie damit machen können, als man muss Demos zeigen, was damit geht, man muss die Leute befähigen, das heißt kurze Schulung, man muss einen kurzen Draht zu dem Mensch haben können. Also ganz klassisch, manche Leute beißen sich einen Tag lang an einer Formel die Zähne aus, die ich wahrscheinlich in einer Minute runtertippen könnte. Da hilft es, also ich habe da immer ein offenes Ohr für die Leute, die können sich direkt an mich wenden und ich versuche schnell zu helfen und durch die Rolle von einem Enabler haben wir auch eine Community etabliert, was ich hier auch als Erfolgsfaktor sehen würde</p> <p>EB02_Rosengrün, Robin: 29 - 29 (0)</p>	<p>Der Experte sieht eine Rolle, die sich um den Aufbau der Governance-Struktur kümmert, als wichtig an.</p> <p>EB02_Rosengrün, Robin</p>
	<p>Einmal das Thema Governance an sich. Also wie sieht die Projektstruktur aus und auch auf Kundenseite.</p> <p>EB03_Fabian Gackstatter: 29 - 29 (0)</p>	<p>Der Experte sieht die Governance als Teil des Projektteam.</p> <p>EB03_Fabian Gackstatter</p>
	<p>Ja, muss nicht zwangsläufig, da müssen wir das jetzt auch wieder trennen. Es gibt das eine Projekt Firma XY, die Wasserwunder GmbH, möchte jetzt eine Low</p>	<p>Der Experte stellt heraus, dass für ein Einführungsprojekt jemand vorhanden ist, der sich um die Governance kümmert. Dies soll vermeiden, dass</p>

	<p>Code-Lösung einführen, bei sich im Unternehmen. Das ist ja ein Projekt und dann haben wir das Projekt mittels der Low Code-Lösung XY soll der Prozess ABC umgesetzt werden. Das sind ja zwei komplett unterschiedliche Prozesse, die andersartiger nicht sein könnten. Insofern, da muss man auch noch mal unterscheiden. Ich würde aber sagen, für die Umsetzung, also nicht Einführung, sondern für die Umsetzung der Anforderungen braucht es nur den Citizen Developer. Für die Einführung der Lösung im Unternehmen braucht es ganz dringend nicht nur den Citizen Developer, sondern hier ist es ganz wichtig, dass wir diese Restriktion, Konventionen im Blick behalten, das heißt, dass diese Governancebrille aufgenommen wird, damit eben nicht diese Nachteile und Schwächen, die wir vorhin schon mal diskutiert haben, eintreten. Insofern halte ich es für essenziell, ich würde aber auch gleichzeitig behaupten, dass der Citizen Developer diese Notwendigkeit nicht sieht und es im Gegenteil als einen Knüppel zwischen den Beinen empfindet.</p> <p>EB06_Hesse, Moritz: 69 - 69 (0)</p>	<p>negative Konsequenzen durch fehlende Governance auftreten.</p> <p>EB06_Hesse, Moritz</p>
<p>10 - Rollen und Stakeholder im Projekt\10.6 - Projekt-Schnittstellen-Rolle</p>	<p>wir setzen uns normalerweise zusammen, ein Projektteam auf unserer Seite, mit einem insgesamt Architekten. Also jemand wie ich, der die Fäden zusammenhält und dann habe ich verschiedene Consultants und weitere Architekten dabei, die die einzelnen Teilströme abdecken.</p> <p>EB03_Fabian Gackstatter: 29 - 29 (0)</p> <p>Dann der Architekt selber, der irgendwo zwischen den Stühlen sitzt, der Teil der Entwicklung dieses Teams ist, aber auch sehr eng mit einem Project Manager und dem Product Owner zusammensitzt, einfach im Sinne von Übersetzung.</p> <p>EB03_Fabian Gackstatter: 49 - 49 (0)</p>	<p>Der Experte sieht eine Rolle für die Kommunikation als sinnvoll an. Er beschreibt es als Rolle Architekt. Er nennt aber auch einen Projektleiter als vorhandene Rolle. Diese Rollen verbinden die unterschiedlichen Teams des Projekts.</p> <p>EB03_Fabian Gackstatter</p>
	<p>Ich würde auch nicht mehr von einem Projektleiter sprechen, sondern von einem Product Owner für diese Plattform und das ist im Idealfall der, der sie auch später dann in der Linie, wenn es eingeführt ist, übernimmt. Klar ist es immer</p>	<p>Die Expertin sieht einen direkten Projektleiter nicht als Teil des Teams, sondern sieht diese Rolle eher bei einem Product Owner.</p> <p>EB04_Görtl, Simone</p>

wichtig, darüber eine Art Stakeholder-Steuerungskreis zu haben, Sponsoren, die auch Entscheidungen treffen können, das heißt, das sind dann entweder Bereichsleiter, Vorstände oder wie auch immer das in dem Unternehmen aufgesetzt ist, um auch Entscheidungen herbeiführen zu können.

EB04\_Göttl, Simone: 31 - 31 (0)

Ja, ist durchaus sinnvoll, je nach Größe. Aber eher so in der coachenden Rolle, so dass man im Prinzip eine Instanz hat, die bei kritischen Themen dann noch mal entscheidend darüber geht und noch mal klares Gewicht reingibt. So wie der Bundespräsident, so was in der Art. Der hat ganz kraftvolle Worte, aber am Ende des Tages ist er jetzt nicht der Entscheidungsträger. So eine Rolle ist wirklich gut, wenn man die im Projekt noch besetzen kann. Das auch jeder weiß, zur Not kann ich mich an den wenden. Also der hätte dann neben der Projektleiterrolle noch andere Rollen impliziert. Die des Schlichters zum Beispiel, des Beraters.

EB05\_Rohrbeck, Christian: 61 - 61 (0)

Ich brauche durchaus bei größeren Projekten auch jemand, der einen Blick auf das Projekt hat. Also ob das jetzt ein fachlicher Projektleiter ist oder ein PMO, im Sinne eines eher administrativen Officers. Der fachliche Projektleiter muss sowieso irgendwie mit drinne sein. Entweder das implizit der Kunde und ich, irgendwie so, je nachdem ob das der Kunde und ich ist oder ob das eine dedizierte Instanz ist, die irgendwie benannt wird und so eine administrative PMO-Stelle braucht man sicherlich auch, wenn es ein umfangreiches Projekt ist. Wenn das so ein 2-Jahres-Läufer ist, mit 35 beteiligten Abteilungen und paar 100, da brauche ich schon eher einen PMO als wenn es das kleine 2-Personen-Projekt ist.

EB06\_Hesse, Moritz: 65 - 65 (0)

Also, sehr wichtig aus meiner Sicht ist halt jemand, der das ganze Thema

Der Experte sieht die Rolle eines Projektleiters weniger in der organisatorischen Rolle, sondern eher in einer zusätzlichen Entscheidungsinstanz, die einem durch das Team beschlossenen Vorgang mehr Gewicht verleiht. Dieser Rolle wird ebenfalls die Aufgabe eines Schlichters zugeordnet.

EB05\_Rohrbeck, Christian

Der Experte hebt hervor, dass er die Projektorganisationsrolle im Team sieht. Dabei unterscheidet er nach Projektleiter und für größere Projekte, welche viele Unternehmensorganisationen betreffen, nach PMO.

EB06\_Hesse, Moritz

Der Experte sieht die Rolle des Projektleiters als eine Rolle, die zwischen der

	<p>vorantreibt. Ja, so ein bisschen Begeisterung für das Thema schafft, der das einmal auf die Agenda legt. Bei den ganzen Rollen in der Organisation, die darüber entscheiden. Also das ist, glaube ich, eine zentrale Rolle, weil das am Anfang erfahrungsgemäß so ein kleiner Kampf gegen Windmühlen ist. Man darf nicht davor zurückschrecken, sondern man muss das Thema eben kontinuierlich treiben und das klappt am besten, wenn man selber dahintersteht, also im besten Fall hat der, ich nenne ihn jetzt mal Projektleiter für dieses Thema, eben eine intrinsische Motivation für das Ganze, EB07_Janke, Tino: 39 - 39 (0)</p>	<p>Organisation und dem Projektteam vermittelt. Diese Rolle sollte auch innerhalb der Organisation Erfolge und Motivation dafür kommunizieren. EB07_Janke, Tino</p>
	<p>Wenn so eine Fahrradmanufaktur auch jetzt nicht über tausende Menschen verfügt, aber es können schon einige hundert sein, die da eingebunden sind. Und da ist dann auch immer eine Abstimmung notwendig. Und das wird dann zum Beispiel mit dem Moderator durchgeführt. EB08_Fey, Peter: 43 - 43 (0)</p>	<p>Der Experte nennt die Rolle des Moderators als Kommunikations- und Abstimmungsschnittstelle. EB08_Fey, Peter</p>
<p>10 - Rollen und Stakeholder im Projekt\10.7 - Changemanagement Rolle</p>	<p>Dann der Bereich des Changemanagements, also wie muss sich die Organisation vorbereiten, im Sinne von Training? Und als Letztes den Bereich die Supportorganisation. Wie verknüpfe ich jetzt meine neue Plattform, meine Citizen Developer, mit den entsprechenden internen Supportstrukturen. EB03_Fabian Gackstatter: 29 - 29 (0)</p> <p>Was bei uns halt wichtig ist, sind solche Leute wie ein Changemanagement-Koordinator, zum Beispiel, der eine eigene Rolle ist. EB03_Fabian Gackstatter: 49 - 49 (0)</p>	<p>Der Experte hebt die Bedeutung von Changemanagement-Verantwortlichen im Projektteam hervor. Diese Projektmitglieder sollten sich darum kümmern, die Organisation auf die Veränderungen vorzubereiten, die durch die Einführung der Plattform entstehen. Dabei werden die Themenfelder Vorbereitung der Trainings, Veränderung der Supportorganisation, Verknüpfung der Plattform mit dem Unternehmen und Supportstrukturen für die Citizen Developer. Zur Koordination wird die Rolle des Changemanagement-Koordinators benannt. EB03_Fabian Gackstatter</p>
<p>10 - Rollen und Stakeholder im Projekt\10.8 - Scrum Master</p>		<p>Der Experte sieht die Rolle eines Scrum Masters als nicht essenziell an. Er beschreibt den Aufgabenumfang eher als moderativ. Dies wird dadurch begründet, dass viele Probleme durch das Team selbst beseitigt werden können. Für technische Probleme ist die IT-Abteilung mit im Projekt, für Entscheidungsprobleme ist der Product Owner mit im Projekt, für</p>

	<p>Datenschutzprobleme ist ein entsprechender Datenschutzverantwortlicher im Projekt und für juristische Aspekte ist ein kaufmännischer Fachverantwortlicher mit im Projekt. Der moderative Faktor des Scrum Masters wird eher als Teilaufgabe gesehen, die durch den Projektleiter ausgeführt werden kann.</p> <p>EB01_Eckhard Hauenherm</p>
	<p>Der Scrum Master wird nicht als wichtig angesehen und die Aufgaben durch die Projektmitglieder erfüllt.</p> <p>EB02_Rosengrün, Robin</p>
	<p>Der Scrum Master wird als Rolle verwendet.</p> <p>EB03_Fabian Gackstatter</p> <p>Die Expertin bestätigt die Nachfrage, dass ein Scrum Master verwendet wird.</p> <p>EB04_Göttl, Simone</p>
	<p>Der Scrum Master als eigenständige Rolle wird als nicht wichtig wahrgenommen aus der Praxiserfahrung. Diese Rolle kann als Teilaufgabe des Projektleiters betrachtet werden.</p> <p>EB05_Rohrbeck, Christian</p>
	<p>Der Scrum Master wird nicht als wichtige Rolle für das Projekt gesehen. Die Aufgaben, die ein Scrum Master erfüllt, können durch die Projektteammitglieder abgeleistet werden.</p> <p>EB08_Fey, Peter</p>
<p>10 - Rollen und Stakeholder im Projekt\10.8 - Scrum Master\10.8.1 - Scrum Master sollte im Projekt sein</p>	<p>Du hast ja jetzt schon Rollen angesprochen. Es ist ja ein Scrum-ähnliches. Hast du dann trotzdem diese gleichen Rollen: Scrum Master, Product Owner, das Developmentteam?</p> <p><b>B: FG</b> Ja die nutzen wir auch.</p> <p>EB03_Fabian Gackstatter: 46 - 47 (0)</p> <p>Das wäre quasi auch meine nächste Frage gewesen, welche wichtigen Stakeholder Sie im Projekt sehen. Bloß aber noch mal zur Klarstellung, aber innerhalb des Projektes sehen Sie die ganz klassischen Scrumrollen wie Scrum Master, Product Owner, das Development-Team, einen Sponsor, das sind so die Rollen, die Sie sehen?</p>



**B: SG** Genau.

EB04\_Göttl, Simone: 46 - 47 (0)

10 - Rollen und Stakeholder im Projekt\10.8 - Scrum Master\10.8.2 - Vorbehalte reine Scrum Master Rolle

Wenn ich über solche Plattformen nachdenke und einen richtigen Scrum Master, meistens ist das eher so eine moderative Funktion, weil viele der Probleme, die dabei auftauchen, tatsächlich im Team selber gelöst werden können. Also es gibt selten Dinge, wo wirklich von außen, also wo ein Scrum Master mal so ein impediment aus dem Weg räumen muss oder sowas, sondern meistens, wenn ich technische Probleme habe, habe ich die IT dabei, insofern kann ich die technischen Probleme lösen. Wenn ich eine strategische Entscheidung brauche, was wollen wir nutzen oder was wollen wir nicht, welche juristischen Aspekte sind zu betrachten, habe ich entweder meinen Produkt Owner oder ich habe meine Datenschutzexpertise sozusagen mit im Team. So dass ich tatsächlich so einen Moderator zwar mache, aber ich brauche nicht so diese, wie im klassischen Scrum, so einen, der so richtig auf die Prozesse achtet und so weiter. Da würde ich sagen, das ist gar nicht so zwingend erforderlich bei so einem Team. Weil es meistens eben nicht so ein ganz streng agiles Projekt ist, wo ein Team konstant über eine ganze Zeit zusammenarbeitet, sondern, wie gesagt, das sind häufig dann sehr versetzte Arbeiten. Man probiert einfach was aus mal und dann ist man wieder unterbrochen, dann probiert man wieder was aus. Und dann muss man das nicht ganz so strikt machen. Also das würde ich jetzt so sagen, wenn ich an meine Projekte denke.

EB01\_Eckhard Hauenherm: 59 - 59 (0)

Und wie macht ihr das dann mit dieser Scrum-Master-Rolle. Du hast gesagt, ihr habt keinen festen Scrum-Master. Wechselt der dann bei euch? Bloß damit ich das auch kurz verstehe.

**B: RR** Mit der Hand am Arm, selber organisiert.

EB02\_Rosengrün, Robin: 52 - 53 (0)

Derjenige, den ich tatsächlich nicht so sehr im Fokus sehe, wäre die Rolle des Scrum Master. Vielleicht ist das auch so, weil ich diese Rolle auch ausfülle, persönlich, kann sein, aber wir hatten in verschiedenen Projekten, die ich im Moment begleite, auch einen reinen Scrum Master und es fällt dieser Person, aus meiner Sicht, in der Realität immer wieder auf die Füße, dass sie zu wenig Fach-Know-how hat. Also rein den Prozess zu begleiten, ist schon fast, naja, ich will es nicht zu sehr abwerten, aber so einen Touch von Sekretärinnen-Arbeit, von Hilfsarbeit. Das ist jetzt zu hart ausgedrückt. Ich finde jetzt aber kein mittleres Wort dafür, aber es entwertet auch so ein bisschen diese Person. Die Rolle des Scrum Masters ist für mich so ein bisschen skurril manchmal.

EB05\_Rohrbeck, Christian: 59 - 59 (0)

Scrum Master -> Nein. Also ist erst mal in dieser Form nicht vorgesehen. Dass Aspekte des Scrum Masters, wenn man den unter der Maßgabe des Enablers sieht, dass die von einzelnen in diesem Bereich wahrgenommen werden, das ist klar. Also das könnte zum Beispiel der Citizen Developer sein, dass der sagt: "Ich kann das hier versuchen, zu lösen." "Ich nehme das mit als Fragestellung usw., kann da Lösungsvorschläge machen." Das ist richtig, aber dass wir dann ein tägliches Daily haben, der ganze Fachbereich sich aufstellt. "Ich habe gestern folgendes gemacht..." "Ich mache heute dieses..." und "Das und das hindert mit." Das ist eigentlich da bei dieser Low Code/ No Code-Programmierung für den Fachbereich nicht vorgesehen. Also du musst immer wirklich dich darauf fokussieren: Wir sind hier im IT-Bereich. Wir machen eine IT-Applikation, aber wir sind nicht die IT. Wir haben eine Vorgabe und wir erarbeiten die innerhalb des Fachbereichs. Und innerhalb des Fachbereichs kommt auch das Feedback und innerhalb des Fachbereichs kommen auch die Lösungen. Dass die

	<p>Abnahme dann durch dritte noch erfolgen sollte, ist eine ganz andere Sache, aber da sind wir festgezurt und insofern die Kollegen und die Kolleginnen, die da im Fachbereich arbeiten, die werden gestern ganz andere Sachen auch gemacht haben, als sich nur um dieses Projekt zu kümmern. Das ist dann eher der Entwickler, der dann an der Tastatur sitzt und sich mit der grafischen Oberfläche der No Code/ Low Code-Applikation oder Plattform beschäftigt hat.</p> <p>EB08_Fey, Peter: 45 - 45 (0)</p>	
<p>10 - Rollen und Stakeholder im Projekt\10.9 - Citizen Developer/ Fachbereichsmitarbeiter</p>	<p>Dann habe ich meinen String des Developer enablements und hier trainiere ich jetzt tatsächlich die Kundenressourcen und auch die Citizen Developer, bereite Applikationen vor.</p> <p>EB03_Fabian Gackstatter: 29 - 29 (0)</p>	<p>Der Experte sieht Citizen Developer als Teil des enablements als Teil des Projektteams.</p> <p>EB03_Fabian Gackstatter</p>
	<p>Und vom grundsätzlichen Team, die dann in dem Scrum Team sind, zur Einführung, ist immer so ein bisschen die Frage, wie man aufgestellt ist. Was meine ich damit? Es macht vielleicht auch manchmal keinen Sinn, einen Fachbereichsmitarbeiter für eine Einführung komplett aus seiner Umgebung herauszuholen, weil er dann den Bezug dazu vielleicht verliert, andererseits sagt man, wenn man agil unterwegs ist: Mitarbeiter sollen voll committed sein, das heißt, die sollen eigentlich zu 100 % in einem Scrum-Team arbeiten.</p> <p>EB04_Göttl, Simone: 31 - 31 (0)</p>	<p>Die Expertin sieht Citizen Developer in Form von Fachbereichsmitarbeitern als Teil des Projektteams.</p> <p>EB04_Göttl, Simone</p>
	<p>Und ansonsten vor allem die Leute, die es umsetzen sollen. Die Citizen, die Developer.</p> <p>EB06_Hesse, Moritz: 71 - 71 (0)</p>	<p>Der Citizen Developer wird als Teil des Projektteams gesehen. Dies kann vor allem während der Einführung förderlich sein, zum besseren Verständnis für die Governance-Anforderungen und aus welchem Grund diese so gestaltet wurden.</p> <p>EB06_Hesse, Moritz</p>
	<p>Dann gibt es innerhalb dieser Abläufe, gibt es dann unterschiedliche Rollen, zumindest PMI sieht die vor. Ich greife mal einen raus, das ist der Citizen Developer, der dann tatsächlich in der Lage ist, zu sagen: "So und so könnte diese Applikation umgesetzt werden."</p> <p>EB08_Fey, Peter: 43 - 43 (0)</p>	<p>Der Citizen Developer wird als wichtige Rolle im Projektteam wahrgenommen. Durch die Citizen Developer sollen die für das Unternehmen relevanten Use Cases mit in das Projekt eingebracht werden. Das kann vor allem bei der Anpassung der Plattform auf das Unternehmen unterstützend wirken. Es</p>

	<p>Dann ganz wichtig, die Mitarbeiter im Fachbereich. Die sind diejenigen, die die Produktionsideen bringen. Die die Ideen bringen zur Lösung. Die die Fragen aufwerfen. Die die besten Betrachter sind. Dann diejenigen, die dann halt für die Kommunikation der unterschiedlichen Fachbereiche zuständig sind. Bei so einem Gewerk vom Einkauf über Lager, Produktion, Verkauf, Kunde, Abrechnung, da sind natürlich unterschiedliche Bereiche in einem Unternehmen betroffen.</p> <p>EB08_Fey, Peter: 43 - 43 (0)</p>	<p>erfolgt ebenfalls dadurch eine erste Befähigung der Citizen Developer.</p> <p>EB08_Fey, Peter</p>
<p>10 - Rollen und Stakeholder im Projekt\10.9 - Citizen Developer/ Fachbereichsmitarbeiter\10.8.1 - Vorbehalte Citizen Developer</p>	<p>Ja, muss nicht zwangsläufig, da müssen wir das jetzt auch wieder trennen. Es gibt das eine Projekt Firma XY, die Wasserwunder GmbH, möchte jetzt eine Low Code-Lösung einführen, bei sich im Unternehmen. Das ist ja ein Projekt und dann haben wir das Projekt mittels der Low Code-Lösung XY soll der Prozess ABC umgesetzt werden. Das sind ja zwei komplett unterschiedliche Prozesse, die andersartiger nicht sein könnten. Insofern, da muss man auch noch mal unterscheiden. Ich würde aber sagen, für die Umsetzung, also nicht Einführung, sondern für die Umsetzung der Anforderungen braucht es nur den Citizen Developer. Für die Einführung der Lösung im Unternehmen braucht es ganz dringend nicht nur den Citizen Developer, sondern hier ist es ganz wichtig, dass wir diese Restriktion, Konventionen im Blick behalten, das heißt, dass diese Governancebrille aufgenommen wird, damit eben nicht diese Nachteile und Schwächen, die wir vorhin schon mal diskutiert haben, eintreten. Insofern halte ich es für essenziell, ich würde aber auch gleichzeitig behaupten, dass der Citizen Developer diese Notwendigkeit nicht sieht und es im Gegenteil als einen Knüppel zwischen den Beinen empfindet.</p> <p>EB06_Hesse, Moritz: 69 - 69 (0)</p>	
<p>11 - Wichtige Stakeholder außerhalb des Projekts\11.1 -</p>	<p>Manchmal ist das so IT-Leitung.</p> <p>EB01_Eckhard Hauenherm: 61 - 61 (0)</p>	<p>Durch den Experten werden als wichtige organisatorische Vorgesetzte außerhalb des Projekts IT-Leitung und</p>

Organisatorische  
Vorgesetzte

Bereichsleitung gesehen, da diese Ressourcen bereitstellen müssen.

EB01\_Eckhard Hauenherm

Und wenn es dann schwierig wird, muss vielleicht mal so eine Bereichsleitung oder so was mit reingeholt werden, als wichtige Stakeholder, weil, die müssen natürlich Ressourcen geben. Also die Leute, die da aus dem Business kommen, die müssen da Zeit dafür haben.

EB01\_Eckhard Hauenherm: 61 - 61 (0)

Natürlich, die IT-Leitung sollte der IT auch so bisschen Ressourcen geben, aber inhaltlich habe ich eigentlich das, was ich brauche, im Team schon vorhanden.

EB01\_Eckhard Hauenherm: 61 - 61 (0)

Ich habe das Thema endlich geschafft, in unserer IT-Strategie zu platzieren, auch dass da namentlich genannt wird. Das macht es natürlich nicht zum Selbstläufer, sondern da muss man noch viel Arbeit reinstecken. Also, was ich mir noch wünsche ist, dass das Thema wirklich auch in der IT ankommt, da als valider Lösungsweg gesehen wird.

EB02\_Rosengrün, Robin: 59 - 59 (0)

Als wichtig wird durch den Experten definiert, dass die Einführung und der Betrieb von NCLC-Plattformen ein Teil der IT-Strategie sind. Dafür ist der IT-Abteilungsleiter wichtig, da dieser einen maßgeblichen Einfluss darauf hat.

EB02\_Rosengrün, Robin

Das heißt, es hängt irgendwo unter meinem CTO, seiner Organisation, seiner Admins und der Rest vom Unternehmen hat damit nichts zu tun. Falsch. Ich habe ja gesagt, Plattform betrifft am Ende des Tages alle. Das heißt, ich brauche da auch einen Changemanagementlead auf Kundenseite, der dann das Thema Training beinhaltet.

EB03\_Fabian Gackstatter: 55 - 55 (0)

Als wichtig wird durch den Experten der CTO angesehen.

EB03\_Fabian Gackstatter

Also sinnvoll schon, auf Leitungsebene, weil meistens die Abteilungsleiter auch nur den Überblick haben, was bei Ihnen in der Abteilung alles passiert. Auf jeden Fall sinnvoll halt mit den Linienverantwortlichen, sage ich mal, die würde ich immer mit ins Boot holen. Weil die ja in der Regel auch später dann diesen

Als wichtige Verantwortliche werden durch den Experten die Abteilungsleiter und Linienverantwortlichen der Projektteammitglieder gesehen. Dies Begründet sich dadurch, dass sie während des Betriebes auch für die Umsetzung der Plattform und des Citizen Development- Konzepts zuständig sind.

EB05\_Rohrbeck, Christian

operationalen Betrieb verantworten müssen von der Software, die innerhalb eines Projektes entwickelt wird.

EB05\_Rohrbeck, Christian: 67 - 67 (0)

Und sicherlich einen Digitalisierungsbeauftragten, wenn das nicht Chefaufgabe ist, klar den nenne ich jetzt mal nicht mit. Den nenne ich jetzt mal nicht mit oder ich nenne ihn an der Stelle mal ganz kurz mit. Und ansonsten vor allem die Leute, die es umsetzen sollen. Die Citizen, die Developer.

EB06\_Hesse, Moritz: 71 - 71 (0)

Der Experte sieht als wichtigen Stakeholder entweder den Digitalisierungsbeauftragten oder direkt die Geschäftsleitung.

EB06\_Hesse, Moritz

11 - Wichtige Stakeholder außerhalb des Projekts\11.2 - Sonstige Beteiligte

Ich überlege gerade, wo vielleicht dann noch Dinge sind, wenn Sie ein Unternehmen haben, das selber keine wirkliche IT oder keine große IT hat, dann brauche ich natürlich so was wie den Provider oder so was mit drin, also ist das natürlich ein wichtiger Stakeholder, der darin eine Rolle spielt. Das ist aber nur eine andere Version von IT, die ich dann im Projektteam habe.

EB01\_Eckhard Hauenherm: 61 - 61 (0)

Der Experte weist darauf hin, dass auch externe Firmen relevante Stakeholder sein können, wenn diese eine Dienstleistung erfüllen, welche thematisch oder organisatorisch durch das Projekt betroffen ist.

EB01\_Eckhard Hauenherm

Geschäftsführung, IT, Kultur beobachten und wenn ich sage, die Kultur muss beobachtet werden, dann werden sich natürlich auch die Organisationsformen ändern. Wie gesagt, der Prozess von der Anforderung zum fertigen Produkt ist ja plötzlich ein ganz anderer. Das heißt, es ist durchaus auch sinnvoll, dass irgendwie eine organisatorische Instanz mit drin ist, die diese Prozesse betrachtet und begleitet. Also einerseits, wie ändert sich denn der Prozess von der Anforderung zum Produkt, aber auch, wie finden denn die Schnittstellen statt von eurem Citizen Development gestalteten Produkt bis hin zur Integration in die Unternehmenssoftware. In die bestehende Unternehmenssoftware. Das heißt, dass man hier so eine organisatorische Brille mit aufhat. Entweder in Form vom Geschäftsführer, den ich schon genannt hatte, oder wenn es ein großes Unternehmen ist, was solche Instanzen

Der Experte nennt die Rolle eines Prozessverantwortlichen als wichtigen Stakeholder außerhalb des Projekts. Dieser erfüllt die Aufgabe der organisatorischen Planung und der Prozessbegleitung. Die betrifft alle Prozesse, die etabliert oder verändert werden müssen im Zusammenhang mit dem Einführungsprojekt.

EB06\_Hesse, Moritz

	<p>dediziert bereithält, dann halt auch, dass man die ins Boot holt. EB06_Hesse, Moritz: 71 - 71 (0)</p>	
	<p>Beides, also die Tester sollen sowohl in dem Fachbereich sein, ich grenze das jetzt mal auf den Fachbereich ab. Also die sollen ja genau gucken: "Haben wir das jetzt gut gemacht oder nicht?" Also sie sollen ja selber ihre Arbeit bewerten und ihre Arbeit da auch beurteilen. Und dann soll es natürlich innerhalb des Unternehmens soll es Leute geben, die in der Lage sind, diese Applikation zu testen. Wenn ich jetzt innerhalb der IT-Abteilung meines Unternehmens niemand mehr habe, der frei ist zum Testen, dann muss ich mir am Markt jemand suchen, der das testet. Aber das muss passieren, weil, das ist bei einer IT-Abteilung so. Das ist, wie wenn ich ein Auto baue. Also es kann auch sein, dass ich ein Auto per Hand baue und sage: "Ich lass es auf die Straße los." Aber getestet werden muss es, sonst darf es nicht raus. EB08_Fey, Peter: 57 - 57 (0)</p>	<p>Der Experte sieht die Rolle eines Testers als wichtigen Stakeholder außerhalb des Projektes, welche die entwickelten Applikationen überprüfen. Dadurch kann ein Feedback erfolgen für die Qualität der Plattform und der Umgang mit dieser. EB08_Fey, Peter</p>
<p>11 - Wichtige Stakeholder außerhalb des Projekts\11.3 - Geschäftsleitung</p>	<p>Natürlich sollte so was wie die Geschäftsführung informiert darüber sein, weil wie gesagt, das ist durchaus eine Änderung der Arbeitsweise im Unternehmen.  EB01_Eckhard Hauenherm: 63 - 63 (0)</p>	<p>Der Experte sieht die Geschäftsleitung als wichtigen Stakeholder, da es sich bei dieser Einführung durchaus um eine Änderung der Arbeitsweise im Unternehmen handelt. EB01_Eckhard Hauenherm</p>
	<p>Ich habe das Thema endlich geschafft, in unserer IT-Strategie zu platzieren, auch dass da namentlich genannt wird. Das macht es natürlich nicht zum Selbstläufer, sondern da muss man noch viel Arbeit reinstecken. Also, was ich mir noch wünsche ist, dass das Thema wirklich auch in der IT ankommt, da als valider Lösungsweg gesehen wird. EB02_Rosengrün, Robin: 59 - 59 (0)</p>	<p>Der Experte hebt hervor, dass es wichtig ist, das Thema in der Digitalisierungsstrategie zu platzieren. Diese wird auch durch die Geschäftsleitung festgelegt. EB02_Rosengrün, Robin</p>
	<p>Geschäftsführung: Je nachdem, ob es ein Vorhaben ist, dass ich No Code/ Low Code ganz neu einführe, um ein Digitalisierungsvorhaben umzusetzen, dann ist natürlich die Geschäftsführung, einfach als wesentlicher Projekttrieb, durchaus gefragt. Wenn das jetzt aber heißt, ich</p>	<p>Der Experte sieht die Geschäftsführung als wichtiger Projekttrieb. EB06_Hesse, Moritz</p>

möchte von Low Code-Lösung A auf Low Code-Lösung B und insgesamt sind wir schon sehr agil und Low Codig unterwegs, dann ist dieses Erfordernis natürlich nicht so sehr gegeben.

EB06\_Hesse, Moritz: 71 - 71 (0)

11 - Wichtige Stakeholder außerhalb des Projekts\11.4 - Abteilungen

Natürlich muss ich eventuell, das fällt mir gerade ein, das wäre vielleicht noch so ein Stakeholder, wenn man so an Personalabteilungen denkt. Ich gebe natürlich Leuten Verantwortungsbereiche, also manche Unternehmen haben ja so Personalrollenbeschreibungen, die sehr strikt sind und da könnte das auch ein Punkt sein.

EB01\_Eckhard Hauenherm: 63 - 63 (0)

Natürlich die Leute aus dem Fachbereich, weil, die Leute, die ich im Team habe, machen ja die Applikation nicht für sich, sondern für den gesamten Fachbereich. Also bauen sich einen neuen Prozess, mit dem dann alle arbeiten. Na, da muss natürlich eine Information sein. Da würde ich schon eher in die Breite gehen, was die Information angeht.

EB01\_Eckhard Hauenherm: 63 - 63 (0)

Das sind ja IT-Einführungsprojekte, das heißt, es ist einerseits wichtig, so eine Compliance Abteilung für das Thema Datenschutz mit einzubeziehen, die Informationssicherheit, was auch zulässig ist

EB04\_Görtl, Simone: 45 - 45 (0)

Was mir jetzt noch so als Nebenstakeholder oder Stakeholder mit einfällt, sind, ganz wichtig auch vorab, die Themen Datenschutz, IT-Security und es gibt je nach Unternehmensgröße die verschiedensten Abteilungen. Es gibt zum Beispiel Unternehmen, die haben so eine Art operational resilience dabei. Das bedeutet, was passiert, wenn dieses Systems dann auf einmal ausfällt. Die sind da die SLA's. In welcher Zeit muss wer

Der Experte sieht als wichtigen Stakeholder außerhalb des Projekts die Personalabteilung, die involviert sein sollte. Der Bereich NCLC betrifft oft viele Bereiche in Unternehmen und deren Mitarbeiter. Das kann Auswirkungen auf Verträge von Mitarbeitern oder andere personalorganisatorische Details haben. Als Beispiel werden durch den Experten Personalbeschreibungen genannt.

Fachbereichsmitarbeiter werden ebenfalls als wichtige Stakeholder genannt. Dies begründet sich in der Tatsache, dass diese später entwickelten Applikationen den Fachbereichsmitarbeitern als Unterstützung dienen sollen.

EB01\_Eckhard Hauenherm

Die Expertin nennt die Compliance Abteilung als wichtigen Stakeholder, der mit eingebunden werden sollte, im Fall, dass diese nicht bereits Teil des Projekts sind.

EB04\_Görtl, Simone

Der Experte sieht als wichtige Stakeholder Abteilungen, welche sich mit den Themen Datenschutz und IT-Security beschäftigen.

EB05\_Rohrbeck, Christian



welche Reparatur durchgeführt haben. Also diese Gedanken, so ein Target Operation Modell aufzusetzen. Wie das später im Betrieb aussieht.

EB05\_Rohrbeck, Christian: 67 - 67 (0)

IT, je nachdem wie tief ich die Low Code-Lösung in meine IT integrieren möchte. Also möchte ich, wie ich vorhin gesagt habe, ein SAP-System, Daten sollen rausfallen, am Ende wieder zurückfallen. Ist selbstverständlich, die IT als wichtigen Projektpartner mit ins Boot zu holen, sonst hängen wir die ab. Nicht technisch, sondern kulturell/ organisatorisch, die haben dann einfach keinen Bock mehr. Ich würde mal behaupten die verdrehen so wieso häufig die Augen bei Low Code Lösungen, weil, am Ende ist es viel Wildwuchs, der dann irgendwie in der IT zusammengefeigt werden muss.

EB06\_Hesse, Moritz: 71 - 71 (0)

Der Experte hebt die Notwendigkeit hervor, dass die Abteilung auf einer kulturell/ organisatorischen Ebene mit eingebunden werden sollte und über den aktuellen Stand informiert wird.

Dadurch soll auch innerhalb der IT-Abteilung vermieden werden, dass dort eine Ablehnung der Plattform auftritt.

EB06\_Hesse, Moritz

11 - Wichtige Stakeholder außerhalb des Projekts\11.5 - Mitarbeitervertretung

Also ein Betriebsrat, zum Beispiel, muss auch involviert sein. Wir haben Kunden, wo der Betriebsrat eben so weit weg davon ist, dass der sagt: "Alles was als Applikation im Unternehmen ausgerollt oder entwickelt wird, muss durch einen Betriebsrat genehmigt werden." Was für einen klassischen IT-Bereich natürlich ok ist. Was aber das Citizen Development krass ausbremsen würde, weil, wenn jetzt jeder anfängt, Applikationen zu bauen und jede einzelne Applikation vom Betriebsrat genehmigt werden muss, ist dein Betriebsrat auf einmal der totale Flaschenhals, weil, dafür sind die ja nicht ausgelegt, auf einmal von hunderttausend Leuten hunderttausend Applikationen zu sichten und zu genehmigen, zum Beispiel.

EB03\_Fabian Gackstatter: 21 - 21 (0)

Ich muss meinen Betriebsrat mit abholen, der in den meisten Projekten klassischerweise nicht beteiligt ist, aber dadurch, dass jetzt meine Plattform prinzipiell jeden Mitarbeiter betrifft, muss der Betriebsrat halt involviert sein, muss die Entscheidungen auch mitgehen, muss sich auch darauf einstellen, was heißt

Der Experte hebt hervor, dass die Mitarbeitervertretung mit eingebunden werden muss. Dies kann sich darin begründen, dass beim Rollout von neuen Applikationen im regulären Prozess eine Genehmigung durch die Mitarbeitervertretung nötig ist, um sicherzustellen, dass keine Mitarbeiterrechte verletzt werden. Dies ist für die hohe Anzahl an Applikationen der Citizen Developer jedoch unrealistisch in der Durchsetzung. Dort muss es zu Prozessänderungen kommen und dafür sollte eine Mitarbeitervertretung involviert sein und die Entscheidungen, die getroffen werden, akzeptieren und vertreten.

EB03\_Fabian Gackstatter

das denn im Sinne von: Da kommen jetzt Änderungen auf uns zu. Sachen, die wir vielleicht in der klassischen IT-Entwicklung gemacht haben, funktionieren jetzt mit einer No Code/ Low Code-Plattform ganz anders, wie das Beispiel was ich vorhin gebracht habe mit Unternehmen, die sagen: "Jede Applikation, die ausgerollt wird, muss durch den Betriebsrat genehmigt werden." Ist für eine No Code/ Low Code-Plattform mit Citizen Developern ein Todesstoß an der Stelle, wenn ein Betriebsrat da nicht sauber abgeholt wird, weil, das kann nicht funktionieren.  
 EB03\_Fabian Gackstatter: 55 - 55 (0)

das sollte man in Konzernen nie vergessen, das ist der Betriebsrat oder die Personalversammlung. Das heißt, all das beeinflusst ja Mitarbeiter und da hat der Betriebsrat auch ein Mitsprachrecht und auch solche Tools müssen in so eine Konzern- oder einer Betriebsvereinbarung aufgenommen werden. Das heißt, diese Personen oder Personengruppen mit zu involvieren von Tag eins, ist unheimlich wichtig, weil, am Ende können das die Show Stopper sein, dass ein Tool nicht ausgerollt wird. Habe ich auch schon erlebt. Es war bei einer MS Teams- Einführung bei einem Kunden hat sich der Rollout zwei Wochen verzögert, weil es keine Freigabe des Betriebsrats gab. Und das ist, glaube ich, etwas, was außerhalb dessen ist, was man vielleicht im Framework explizit sieht. Es ist vielleicht ein Stakeholder, aber ein Stakeholder, der halt nicht als Stakeholder bezeichnet werden sollte, sondern eine sehr wichtige Rolle einnimmt.  
 EB04\_Görtl, Simone: 45 - 45 (0)

Die Expertin hebt hervor, dass es notwendig ist, falls vorhanden die Mitarbeitervertretung mit einzubeziehen. Entweder direkt im Projekt oder als wichtiger Stakeholder. Sie verweist darauf, dass es zu in ihrer Praxiserfahrung dazu gekommen ist, dass sich ein Projekt verzögert hat, weil der Betriebsrat die Freigabe dafür nicht erteilt hat.  
 EB04\_Görtl, Simone

11 - Wichtige Stakeholder außerhalb des Projekts\11.6 - Alle Mitarbeiter im Unternehmen

Ja, natürlich. Also, informiert halten würde ich so viele Leute wie möglich.  
 EB01\_Eckhard Hauenherm: 63 - 63 (0)

Der Experte sieht es als wichtig an, dass im besten Fall möglichst alle Mitarbeiter über das Projekt informiert werden.  
 EB01\_Eckhard Hauenherm

Also grundsätzlich ist es immer die Frage, wie man es dann aufsetzt und wer in so einem Einführungsteam ist. Am Ende geht es ja darum, dass die Mitarbeiter, je nachdem fünftausend,

Der Experte sieht es als sehr wichtig an, dass möglichst alle Bereiche des Unternehmens informiert werden über die Plattform. Zielstellung ist, dass möglichst viele Mitarbeiter die

	<p>zehntausend, vielleicht auch nur fünfhundert Mitarbeiter, die das Tool nutzen können, das heißt, es ist ja auch in gewisser Weise ein Stakeholderkreis und den muss ich ja auch bespielen oder die auf die Reise mitnehmen. Was dann passiert, wenn ich die nicht mit auf die Reise nehme, dann sage ich an einem Tag, ich schalte jetzt den Schalter um und das Tool, es gibt die No Code/ Low Code-Plattform, aber keiner davon weiß oder was man damit machen kann und ich finde einfach, das ist ja schon ein Veränderungsprozess, ein Change, das heißt, es sind auch die Fachbereiche bzw. auch sämtliche Einheiten des Unternehmens in gewisser Form mitzunehmen. Das kann über Leuchttürme da rein passieren, aber auch über grundsätzliche Kommunikationsmaßnahmen vorab und das ist, glaube ich, ein Stakeholder, den man häufig unterschätzt, weil, am Ende müssen die Mitarbeiter es ja auch nutzen.</p> <p>EB04_Göttl, Simone: 49 - 49 (0)</p>	<p>Applikationen, die durch die Citizen Developer entstehen, nutzen können und somit Mehrwert generieren. Dies kann durch Multiplikatoren erfolgen oder auf anderen Wegen.</p> <p>EB04_Göttl, Simone</p>
<p>11 - Wichtige Stakeholder außerhalb des Projekts\11.7 - Support Abteilungen</p>	<p>Die muss ich auf alle Fälle mit reinnehmen und auch meine Supportstruktur muss mit eingebunden werden, weil jetzt eine Plattform auszurollen, heißt halt nicht nur: "Wir rollen ein neues Office Paket aus." Das heißt: "Hier sind die zehn neuen FAQs, die die Supportorganisation beinhalten muss." Sondern ich habe hier auf einmal eine riesige Entwicklungsgruppe sitzen, die sich dem Thema IT auseinandersetzt, die ganz andere Fragen haben werden als der klassische Anwendungsnutzer. Also, der anruft und sagt: "Hier mein Excel spinnt. Könnt ihr mir das mal neu installieren?" Du wirst auf einmal auch User haben, die dann Entwicklungsfragen stellen. Wie baue ich eine Canvas-App? Wie baue ich eine Model Driven-App? Wie komme ich an diesen Connector ran? Das heißt, entweder baue ich eine komplett separate Supportorganisation auf, die sich dann um diese Themen auch dediziert kümmern kann. Oder ich bohre entsprechend meine bestehende Supportorganisation so weit auf, dass die diesen zusätzlichen Demand auch handeln können. Und wie</p>	<p>Der Experte hebt es, als sehr wichtig hervor, die zugrundeliegenden Support-Organisationen einzubeziehen. Dies wird nötig, da auf eine große Entwicklergruppe, welche nicht aus klassischen IT-Strukturen kommt und aus diesem Grund einen veränderten Supportbedarf hat. Dies kann dazu führen, dass sich die Struktur der Support-Organisation ebenfalls ändert. Entweder durch eine eigene Support-Organisation für die Entwickler oder durch zusätzliche Ressourcen für die bestehende.</p> <p>EB03_Fabian Gackstatter</p>

gesagt, dafür haben wir ja in der Implementierung einen komplett eigenen Stream, der nichts andere macht, als eine Supportorganisationen aufzuschauen und zu integrieren, die ich ja so in einem klassischen Entwicklungsprojekt nicht habe.

EB03\_Fabian Gackstatter: 55 - 55 (0)

Und das größte Problem, was die halt haben, ist, dass das Projektteam dann weg ist, wenn das Projekt dann abgeschlossen wird, aber die das restliche Leben lang mit den Ergebnissen zu tun haben werden. Also hier sehe ich die größten Hindernisse und die größten Chancen mit für so ein Projekt, dass man frühzeitig an die herantritt, die damit später involviert sind.

EB05\_Rohrbeck, Christian: 67 - 67 (0)

Das sollte man auf jeden Fall in der Projektphase schon anfangen und alle die, die dann später involviert sind, sei es ein User Help Desk, diese Software geht kaputt und irgendjemand möchte jemand qualifiziert anrufen, der ihm dann hilft. Dann muss dieses User Help Desk informiert werden. Also es ist von Unternehmen zu Unternehmen und der Größe unterschiedlich und dem, was man macht. Ganz wichtig, aus meiner Sicht, in der Projektphase schonmal sich Gedanken machen, wie es später in der Business as Usual-Phase aussieht. Wer dann für Störungen verantwortlich ist? Wie lange die Reparaturfristen sind? Ist das vertraglich abgesichert? Dafür die richtigen Stakeholder herauszufinden, das ist schon ganz wichtig.

EB05\_Rohrbeck, Christian: 67 - 67 (0)

Der Experte hebt die Notwendigkeit hervor, in einem möglichst frühen Stadium Support-Organisationen einzubinden. Aufgrund der weitreichenden und neuen Plattform müssen die während des laufenden Betriebs die Entwickler unterstützen, wenn zum Beispiel die technischen Berater aus dem Projektteam nicht mehr im Unternehmen sind, weil es Fremddienstleister waren. In diesem Zusammenhang sollten ebenfalls Prozessevaluierungen vorgenommen werden, für Störsituationen und wie mit diesen umgegangen wird.

EB05\_Rohrbeck, Christian

12 - Skalierung Pilotprojekt\12.1 - Sequenzielles Vorgehen

Schrittweise.

**I: RS** Heißt Abteilung für Abteilung für Abteilung, oder?

Der Experte beschreibt ein sequenzielles Vorgehen bei der Skalierung der Plattform im Unternehmen. Bei diesem Vorgehen werden weitere Abteilungen nacheinander mit der Plattform befähigt und die Erfahrung aus den vorhergehenden Abteilungen geteilt werden. Er

**B: EH** Tatsächlich, also mal gucken, weitere Anforderungen sammeln. Weitere Anforderungen sammeln, weitere Möglichkeiten sammeln, wo gibt es Bedarf oder wo gibt es Möglichkeiten, das, was wir jetzt gelernt haben, weiter zu treiben, Effizienzgewinne herauszuholen und solche Dinge, wo gibt es Wünsche? Aber tatsächlich schrittweise. Ich würde das, glaube ich, tatsächlich immer so machen. Zieht sich zwar vielleicht ein bisschen. Also die Alternative wäre, man kann sich so zwei Vorgehensmodelle überlegen. Entweder eine schrittweise Einführung. Ich gehe dann immer weiter, bis ich das so einigermaßen flächendeckend habe, dann wird es wahrscheinlich auch so ein bisschen zum Selbstläufer. Also irgendwann können die Leute sich das auch selber beibringen, weil, so schwierig sind die Plattformen dann auch nicht. Ich habe zwei Leute, die haben das schonmal gemacht, die gucken sich mit einem Dritten das an und zeigen dem dann mal, wie das geht und dann kann der das auch, so ungefähr.

EB01\_Eckhard Hauenherm: 65 - 67 (0)

Dass, ja, dass es eben dazu führt, dass eben alle an einem Strang ziehen. Auf der anderen Seite würde ich von der Seite der agilen Methoden, kann ich mir vorstellen, dass man durchaus auch in Sprints arbeitet, also wenn man zum Beispiel darüber spricht, dass man in einer iterativen Vorgehensweise einzelne Fachbereiche in dieses Projekt hinzuholt und die darin schult. Das wäre eine Sache, wo man iterativ vorgehen könnte, einfach damit man die Gelegenheit hat, aus den Erfahrungen des einen Onboarding des Fachbereiches für den nächsten zu lernen. Ich würde bei den agilen Methoden auch Wert darauf legen, schnell Erfolge sichtbar zu machen, dass man also nicht mit einem beispielsweise Big Bang arbeitet und alles passiert im Kämmerchen bis das ganze Thema leicht geht, sondern man muss eben zwischendrin auch die Erfolge sichtbar machen. Also immer ein Stückchen mehr,

merkt an, dass es zeitlich etwas länger dauern kann, allerdings kann die Skalierung ab einem bestimmten Punkt mit immer geringerem Aufwand betrieben werden.

EB01\_Eckhard Hauenherm

Der Experte sieht einen Vorteil darin, in einer agilen Weise die Skalierung zu gestalten. Dabei wird in Sprints je eine Abteilung in das Einführungsprojekt hinzugezogen und die Mitarbeiter befähigt. Vorteil dieser Vorgehensweise ist, dass ein ständiger Lernprozess etabliert wird und Erfahrungen, welche bei der Skalierung in eine Abteilung gesammelt werden, wiederum für die nächste weiterverarbeitet werden können. Es sollte nicht passieren, dass die Plattform zu einem spezifischen Zeitpunkt einfach eingeführt wird, sondern es sollte stetig ein Projektfortschritt kommuniziert werden.

EB07\_Janke, Tino

ja, Inkrement sozusagen, ein Stückchen mehr zeigen, was man geschaffen hat, um die Projektmitglieder einfach am Ball zu behalten und auch den Stakeholdern zu zeigen, dass man hier konsequent Mehrwert schafft für die Organisation.  
EB07\_Janke, Tino: 35 - 35 (0)

12 - Skalierung Pilotprojekt\12.2 - Paralleles Vorgehen

eine zweite Vorgehensweise, mit einem Piloten anzufangen und zu gucken, wenn der funktioniert, dann stellen wir es flächendeckend erst mal allen Abteilungen zur Verfügung. Das wird man aber nach meinem Dafürhalten tatsächlich nicht so betreuen können, dass die Leute wirklich anfangen damit zu arbeiten. Es sei denn, es gibt wirklich Schmerzen in den Abteilungen, wo die sagen: "Die können wir nur damit lösen." Dass die sozusagen freiwillig darin sind. Sonst wird das nach meiner Erfahrung eher so enden, wenn ich sofort in die Breite gehe: "Ja hier hat sich schon wieder mal einer ein neues Tool überlegt, aber eigentlich ich komme mit meinen Excel-Dateien eigentlich ganz gut klar. Warum soll ich mich jetzt damit beschäftigen."

EB01\_Eckhard Hauenherm: 67 - 67 (0)

Das ist eine gute Frage, große Herausforderung. Wir versuchen nicht durch Manpower zu skalieren, sondern durch Automatisierung. Also, wie das Beispiel, ich muss nicht von Hand Lizenzen vergeben, sondern richte dafür feste Prozesse ein. Also für die Governancethemen richte ich feste Prozesse ein, die mir erlauben, das effizienter zu machen, dass ich trotzdem mit, ich glaube, wir haben gerade 150 % Governance, also 1,5 Personen die, die Governance machen, dass die das sowohl für 100 Menschen geschafft haben, aber durch weitere Professionalisierung auch für 2000 aktive Maker schaffen. Und genauso im Enabling, dass ich mir Strukturen suche, die ich anfangs für einen geringen Personenanteil, auch mit einem hohen manuellen Anteil, dann eben gemacht hab, aber dass wir auch Strukturen haben, die sich sofort an alle richten. Ein Beispiel dafür ist bei uns zum Beispiel, dass wir

Der Experte beschreibt, zusätzlich zu einem sequenziellen Vorgehen, ebenfalls ein paralleles Vorgehen. In diesem Modell würde es zu einem unternehmensweiten Roll-Out der Plattform kommen. Der Experte hebt jedoch hervor, dass es schwierig sein kann, alle Abteilungen mit den erforderlichen Ressourcen zu betreuen. Er rät jedoch von diesem Vorgehen ab, es sei denn, eine Lösung über NCLC-Plattform ist die einzige Lösung für eine spezifische Problemstellung. Im Falle der schlechten Betreuung kann es dazu führen, dass die Plattform abgelehnt und auf alte Methodiken zurückgegriffen wird.  
EB01\_Eckhard Hauenherm

Der Experte sieht es als vorteilhaft, die Skalierung über Automatisierung von Prozessen zu betreiben, statt über die Erhöhung von personellen Ressourcen. Als Beispiel führt er an, dass ein in Person gehaltener Workshop mit Trainer vor Ort durch einen Video-Self-Service-Kurs ersetzt wurde. Dadurch ist es möglich, mehrere Abteilungen zu befähigen, ohne einen erhöhten Personalaufwand zu haben. Es wird hervorgehoben, dass ein gewisser erhöhter personeller Ressourcenbedarf mit steigender Skalierung nötig wird.  
EB02\_Rosengrün, Robin

nicht mehr monatlich feste Kurse für PowerApps/ PowerAutomate geben, sondern dass ich einen Videoselfservice-Kurs an irgendeinem Punkt gemacht habe, den die Leute dann selber durchklicken können, das ich dann nicht sagen muss, „Melde dich zum nächsten Kurs an“, sondern „Hier ist der Selfservice Kurs, kannst starten“ und ja, das nimmt natürlich Arbeit ab und ja, dem Kurs ist es egal, ob er 1-mal im Monat oder 1000-mal im Monat geklickt wird. Dass man halt als manueller Trainer oder als persönlicher Trainer nicht mehr schaffen kann. So versuchen wir unsere Strukturen weiter zu professionalisieren. Natürlich schon personell ein Stück weit zu wachsen, aber nicht annähernd mit der Anzahl der Nutzer.

EB02\_Rosengrün, Robin: 55 - 55 (0)

dann würde ich zu so einem skalierten Framework tendieren, wie SAFe, oder man könnte auch, wenn es mehrere Teams sind, an ein Scrum of Scrum denken oder ein Nexus.

EB04\_Göttl, Simone: 41 - 41 (0)

Die Expertin sieht für die Skalierung einen Wechsel in der Vorgehensweise als sinnvoll an. Als Beispiel nennt sie das SAFe Framework oder die Methodik Scrum-of-Scrums.

EB04\_Göttl, Simone

Ja, in der Regel ist es so, man ist erstmal isoliert mit seiner Entwicklung. So ein Projekt ist ja abseits des Daily Business und ab einem gewissen Punkt hat das System, das Testentwicklungssystem, eine Reife erhalten, in dem man einen acceptance test durchführen kann. Spätestens an diesem Punkt macht es Sinn, da so einen Prototyp vorzustellen und auch bestimmte Leute einzuladen, daran teilzuhaben, daran teilhaben zu lassen, in dem man denen zum Beispiel einen Testaccount verschafft, damit die das dann weitergeben, diese Information über die Systemeinführung. Wichtig ist hier erst mal, natürlich ein Produkt zu schaffen, was den Vorgaben entspricht und dass die Leute, die in dieser Entwicklung dabei sind, natürlich immer abgeholt werden just in time. Das ist klar, aber wenn es jetzt um die unternehmensweite Einführung geht, man in dem Moment, wo man ein vorzeigbares, sage ich mal, Deliverable hat. Also muss nicht das Gesamtprodukt sein. Diejenigen, die

Der Experte hebt die Bedeutung von frühzeitiger Kommunikation mit den Stakeholdern und Anwendern hervor. Es sollten in Vorbereitung auf die Skalierung bereits frühzeitig erste Ergebnisse gezeigt werden. Dies sollte nicht nur auf einer Vortragsebene erfolgen, sondern die Nutzer sollten ebenfalls die Möglichkeiten haben, erste eigene Erfahrungen mit der Plattform zu sammeln. Ist diese Vorbereitung erfolgt, kann eine entsprechende Skalierung in die vorbereiteten Abteilungen erfolgen.

EB05\_Rohrbeck, Christian

es später betrifft, schon frühzeitig abholt und denen anhand eines Beispiels oder eines Testaccount oder so zeigt, was das bewirken wird. Meine Erfahrung ist hier, dass reine Worte oft links rein, rechts rausgehen. Also ohne visuelle Unterstützung, im Idealfall durch einen Account oder Testaccount oder halt durch eine PowerPoint etwas aufbereitet, die abholt und dann mit teilhaben lässt an der Entwicklung, soweit das möglich ist, durch eigene Gedanken so, und dann, wenn die Leute, die Kollegen, mental abgeholt sind, dann kann man das auch unternehmensweit einführen.

EB05\_Rohrbeck, Christian: 69 - 69 (0)

12 - Skalierung Pilotprojekt\12.3 - Sequenziell-Paralleles Modell

Also, Sie haben ja erst mal grundsätzlich gesagt, wir versuchen gerne, mit den Kunden zusammen die Eignung des Produkts zu testen oder zu validieren, indem wir es in einem fokussierten Bereich, in einem bestimmten Department, in einer bestimmten Abteilung zunächst erst mal einführen und dort vereinzelte Prozesse testen und wenn der Kunde damit zufrieden ist. Das ist ja so eine Kundenentscheidung. Wir möchten natürlich gerne, dass der Kunde das hoch skaliert und in allen Bereichen anwendet als Anbieter und der Kunde überträgt dann im Grunde sein Wissen, also das ist in der Vergangenheit immer so gewesen, dass man sich dann weitere Abteilungen sucht, wo es gut passen würde und dann entsprechend den Transfer auf diese Abteilungen vornimmt. Wenn das nicht sowieso als eine große Digitalisierungschtaufgabe kam, dass es ganz weit eingeführt werden soll, ist das so der Fall, dass man zunächst einmal beginnt, Prozesse und die Eignung zu testen im kleinen Rahmen und das dann halt ausrollt auf weitere Prozesse, auch mit unterschiedlichen Verantwortungsrollen. Manchmal ist das der Fall, dass man sagt, einer betreut alle Prozesse für verschiedene Abteilungen oder man sagt, man gibt die Verantwortung in die einzelnen Abteilungen. Da gibt's verschiedene Varianten und Lösungen, was durchaus auch immer ein Ansatz bei uns ist, um diesen

Der Experte erläutert für die Skalierung, dass zu Beginn von einer Pilotabteilung heraus skaliert werden sollte. Im weiteren Verlauf der Skalierung sollte erst schrittweise vorgegangen und in den jeweiligen Abteilungen vereinzelte Prozesse evaluiert werden, die sich für die Umsetzung auf der Plattform eignen und diese dann auf die Plattform übertragen. Ist die Skalierung in eine Abteilung erfolgt, kann die mit einer weiteren fortgeführt werden. Dabei wird auf ein Train-the-Trainer-Konzept angewendet. Dies bedeutet, dass Multiplikatoren ausgebildet werden, in den Abteilungen, welche den initialen Wissenstransfer vom Plattformbetreiber weitergeben an die Abteilungen, in denen die Plattform neu implementiert wird. Dadurch können Skalierungseffekte entstehen, welche ein schnellere Skalierung ermöglichen.

EB06\_Hesse, Moritz



Wissenstransfer, also gar nicht mal die Skalierung der Lösung oder die Diffusion in die verschiedenen Bereiche, sondern mehr so der Wissenstransfer, dass wir so ein Train-The-Trainer-Konzept versuchen herbeizuführen. Das heißt, dass wir sagen, wir versuchen nicht nur die Leute zu befähigen, es selber umzusetzen, sondern darüber hinaus die auch noch zu befähigen, andere Leute zu befähigen. Einfach um da quasi die Abhängigkeit zu verringern, in dem wir quasi sagen, Train-The-Trainer und der soll dann bitte auch die Leute befähigen. Dass wir so einen Multiplikations- und Skalierungseffekt dadurch erzielen.

EB06\_Hesse, Moritz: 83 - 83 (0)

Und wenn die Strukturen dann da sind, um das Thema ausweiten zu können, würde ich dann an die Skalierung gehen, also ich würde mir die nächste Fachabteilung nehmen oder je nachdem auch mehrere, aber nicht alle. Ich würde mir sozusagen die nächsten Fachabteilungen nehmen, die in dieses ganze Projekt oder über dieses Projekt begleiten, die mit den Themen vertraut machen und sie schlussendlich onboarden, also sie in die Lage versetzen, dass sie selber die Low Code/ No Code-Plattform nutzen können. Und da auch wieder der Rückschluss zu dem agilen Arbeiten, also aus dieser ersten Iteration lernen und die nächste Iteration mit der nächsten Fachabteilung dann besser machen und das wäre dann so eine Art Inkubator, wo man sagen kann, man lässt die Fachabteilung nacheinander dieses Programm durchlaufen. Danach sind die Fachabteilungen in der Lage, selber diese Plattformen zu nutzen. Sie wissen, welche Governance, also welche Regeln, sie einhalten müssen, welche Verantwortlichkeiten es für das Thema gibt, sie wissen, wohin Sie sich mit Fragen wenden müssen, nämlich an die Community. Genau. So also, das wäre dann dieses Thema Skalierung und das, glaube ich, über allem so drüber schwebt, wäre eine Phase, die könnte man dann Innovationen nennen, dass man also sagt „Ok, ich hab jetzt die Organisationseinheiten onboarded, sie

Der Experte sieht beim Thema Skalierung eine sequenzielles bis teilweise paralleles Vorgehen als positiv an. Es ist wichtig, dass die Plattform bei einer parallelen Vorgehensweise nicht sofort im ganzen Unternehmen implementiert wird, sondern in vereinzelt Abteilungen. Durch dieses sequenzielle Vorgehen ist es möglich eine agile Vorgehensweise in Iterationen durchzuführen und aus den Erfahrungen und dem Feedback, das beim Implementieren gesammelt wird, Schlussfolgerungen für die nächsten Abteilungen zu ziehen. Teil der Skalierung in den einzelnen Abteilungen, sollte eine umfassende Befähigung auf die Plattform sein, sowie der Aufbau von Wissen über die im Unternehmen erstellten Governance, IT-Security und Datenschutz Rahmenbedingungen.

Ein Prozess, der während der Skalierung, nachdem die ersten Abteilungen erfolgreich eine Implementierung durchlaufen haben, beginnen sollte, ist ein kontinuierlicher Verbesserungsprozess. In diesem Prozess ist es wichtig, Veränderungen an der Plattform weiterzugeben und zu evaluieren, wie dies das weitere Skalierungsgeschehen beeinflusst und wenn nötig, Anpassungen an den Prozessen und Rahmenbedingungen vorzunehmen.

EB07\_Janke, Tino

nutzen die Plattform, es kommen Ergebnisse raus und jetzt schauen wir, wie wir das Ganze eben kontinuierlich verbessern können.“ Also wir wollen uns nicht auf dem eigenen Stand ausruhen, das wird auch nicht funktionieren, weil die Plattform wächst, also die Low Code/ No Code-Plattform wird wachsen. Die Funktionalitäten werden sich ändern und die Skills der Fachabteilungen werden wachsen und das ist einfach etwas was, für das man sich ständig mit diesem Thema beschäftigen muss und seine einmal getroffenen, zum Beispiel Leitplanken, auch immer wieder hinterfragen sollte. Das wäre dann so dieses Thema Innovation, wäre dann so, dass das letzte beziehungsweise eigentlich auch wieder das Thema, wo dann die Iteration die nächste Iteration startet.

EB07\_Janke, Tino: 43 - 43 (0)

12 - Skalierung Pilotprojekt\12.4 - Vorbereitende Maßnahmen

Und auch in der Einführung, wenn wir da schon kleine Tools entwickeln, die vielleicht präsentieren und mal zeigen und "Guck mal, so geht das und so einfach war das." Und solche Sachen, das würde ich schon so sagen.

EB01\_Eckhard Hauenherm: 63 - 63 (0)

Eine vorbereitende Maßnahme, die der Experte für die Skalierung sieht, ist das Vorzeigen von Zwischenergebnissen in der Organisation.

EB01\_Eckhard Hauenherm

Da kommt eben unser Center of Excellence ins Spiel. Dass wir sagen, wir schaffen jetzt hier die saubere Grundlage zum einen von den Prozessen her, damit vom Prozess her klar ist, ein User möchte eine Applikation bauen, das sind die Schritte, die er tun muss. Und das funktioniert mit zehn Usern, das funktioniert mit hunderttausend Usern. Dass der Prozess sauber etabliert ist und zum anderen aber auch vom Thema Plattform enablement. Dass die richtigen Umgebungen zur Verfügung gestellt werden, dass das Thema Lizenzmanagement adressiert ist, dass das Thema Monitoring und Reaction sauber definiert ist, wenn ein User das und das macht, was er nicht tun sollte, was passiert dann? Geht dann erst mal eine Mail nur an den Admin raus oder schreitet die Plattform selber ein und unterbindet gewissermaßen Aktivitäten. Die Grundlagen müssen

Der Experte vertritt die Ansicht, dass über eine dedizierte Einheit sich mit der Skalierung beschäftigt wird. Die Aufgabe dieser Einheit ist, die nötigen Prozesse zu definieren und zu entwickeln. Dies wird als Grundlage der Skalierung gesehen, dass Vorgehensweisen erstellt werden für die verschiedenen Prozesse und Situationen, die während des Betriebs passieren. Hierbei sollten Prozesse erstellt werden für verschiedene Szenarien, welche Lizenzmanagement und Monitoring der Plattform betreffen. Es sollten Richtlinien erstellt werden, wie mit unberechtigten Datenzugriffen und mit Sicherheitsrisiken umgegangen wird.

Sind diese Grundlagen hergestellt, sieht der Experte wenig Probleme für die Skalierung.

EB03\_Fabian Gackstatter

halt sauber definiert werden und dann skaliert so ein Center of Excellence auch sauber oder eine Plattformeinführung. Und deswegen ist halt wichtig und dafür haben wir dedizierte Angebote, die den Kunden eben genau dabei unterstützen, diese Grundlage zu schaffen, damit Skalierung eben hinterher funktioniert.

EB03\_Fabian Gackstatter: 57 - 57 (0)

Also grundsätzlich ist es immer die Frage, wie man es dann aufsetzt und wer in so einem Einführungsteam ist. Am Ende geht es ja darum, dass die Mitarbeiter, je nachdem fünftausend, zehntausend, vielleicht auch nur fünfhundert Mitarbeiter, die das Tool nutzen können, das heißt, es ist ja auch in gewisser Weise ein Stakeholderkreis und den muss ich ja auch bespielen oder die auf die Reise mitnehmen. Was dann passiert, wenn ich die nicht mit auf die Reise nehme, dann sage ich an einem Tag, ich schalte jetzt den Schalter um und das Tool, es gibt die No Code/ Low Code-Plattform, aber keiner davon weiß oder was man damit machen kann und ich finde einfach, das ist ja schon ein Veränderungsprozess, ein Change, das heißt, es sind auch die Fachbereiche bzw. auch sämtliche Einheiten des Unternehmens in gewisser Form mitzunehmen. Das kann über Leuchttürme da rein passieren, aber auch über grundsätzliche Kommunikationsmaßnahmen vorab und das ist, glaube ich, ein Stakeholder, den man häufig unterschätzt, weil, am Ende müssen die Mitarbeiter es ja auch nutzen.

EB04\_Göttl, Simone: 49 - 49 (0)

Ja, in der Regel ist es so, man ist erstmal isoliert mit seiner Entwicklung. So ein Projekt ist ja abseits des Daily Business und ab einem gewissen Punkt hat das System, das Testentwicklungssystem, eine Reife erhalten, in dem man einen acceptance test durchführen kann. Spätestens an diesem Punkt macht es Sinn, da so einen Prototyp vorzustellen und auch bestimmte Leute einzuladen, daran teilzuhaben, daran teilhaben zu lassen, in dem man denen zum Beispiel einen

Die Expertin sieht als wichtige vorbereitende Maßnahme der Skalierung eine erfolgreiche Kommunikation über die Veränderungen, welche durch die Plattform im Unternehmen ausgelöst werden. Die Abteilungen sollten in die entsprechenden Kommunikationskanäle über den Fortschritt des Einführungsprojekts eingebunden werden und sollten Zugang zu Informationen über die Plattform erhalten. Eine weitere Möglichkeit der Kommunikation kann über Multiplikatoren erfolgen, welche vorher auf die Plattform befähigt wurden und die Rahmenbedingungen für den Umgang mit der Plattform kennen.

EB04\_Göttl, Simone

Der Experte weist daraufhin, dass eine frühzeitige Kommunikation zu den späteren Nutzern der Plattform aufgebaut werden sollte. Zu diesem Zwecke sollten frühzeitig Ergebnisse präsentiert und die entsprechenden Personengruppen dazu eingeladen werden, diese zu testen. Dies beugt ebenfalls der Problematik von Widerständen vor. Es kann sein, dass Nutzer zu Beginn den Mehrwert der Plattform nicht erkennen, aufgrund der mangelnden Erfahrung

Testaccount verschafft, damit die das dann weitergeben, diese Information über die Systemeinführung. Wichtig ist hier erst mal, natürlich ein Produkt zu schaffen, was den Vorgaben entspricht und dass die Leute, die in dieser Entwicklung dabei sind, natürlich immer abgeholt werden just in time. Das ist klar, aber wenn es jetzt um die unternehmensweite Einführung geht, man in dem Moment, wo man ein vorzeigbares, sage ich mal, Deliverable hat. Also muss nicht das Gesamtprodukt sein. Diejenigen, die es später betrifft, schon frühzeitig abholt und denen anhand eines Beispiels oder eines Testaccount oder so zeigt, was das bewirken wird

EB05\_Rohrbeck, Christian: 69 - 69 (0)

Ja, das ist ein schwieriges Thema, weil also die Erfahrung ist, dass oft der Horizont von Fachkollegen eingeschränkt ist. Ich möchte niemanden zu nahe treten, nicht falsch interpretieren meine Worte, aber ich merke halt, man ist oft das links und rechts und diejenigen, die ein neues System oder später mit einem neuen System arbeiten werden, sollten wirklich frühzeitig da abgeholt werden. Und man wird auch mit relativer Sicherheit da Widerstände spüren. Das bedeutet, hier ist aus meiner Sicht, steter Tropfen höhlt den Stein, die beste Strategie. Man würde sagen, dass man in einem gewissen Zyklus die Änderung vorstellt und auch die Kollegen vom Fachbereich mit einbindet, soweit das geht. So dass die auch aktiv da mitarbeiten können.

EB05\_Rohrbeck, Christian: 71 - 71 (0)

Da gibt's verschiedene Varianten und Lösungen, was durchaus auch immer ein Ansatz bei uns ist, um diesen Wissenstransfer, also gar nicht mal die Skalierung der Lösung oder die Diffusion in die verschiedenen Bereiche, sondern mehr so der Wissenstransfer, dass wir so ein Train-The-Trainer-Konzept versuchen herbeizuführen. Das heißt, dass wir

mit der Plattform. Zur Vermeidung dieser Problematik ist eine erfolgreiche Kommunikation wichtig und das Aufzeigen von Mehrwert.

EB05\_Rohrbeck, Christian

Der Experte hebt als vorbereitende Maßnahme für die Skalierung die Ausbildung von Multiplikatoren hervor. Dies ermöglicht in der aktiven Skalierungsphase, dass die Mitarbeiter weitere Mitarbeiter befähigen können für die Plattform und somit nicht auf den direkten Wissenstransfer durch den Plattformanbieter angewiesen sind.

sagen, wir versuchen nicht nur die Leute zu befähigen, es selber umzusetzen, sondern darüber hinaus die auch noch zu befähigen, andere Leute zu befähigen. Einfach um da quasi die Abhängigkeit zu verringern, in dem wir quasi sagen, Train-The-Trainer und der soll dann bitte auch die Leute befähigen. Dass wir so einen Multiplikations- und Skalierungseffekt dadurch erzielen.

EB06\_Hesse, Moritz: 83 - 83 (0)

Ja genau, also typischerweise muss man ja schnell Erfolge zeigen, damit man einfach immer mehr Leute auf seine Seite holt und von dem Projekt begeistert. Das heißt, im besten Fall steht am Ende des Prototypings ein praxisrelevanter Use Case für das Unternehmen, wo man mal Applikationen gebaut hat, mit dieser Applikation, die ein echtes Problem im Unternehmen löst, kann man dann in andere Abteilungen gehen und sagen:

„Guck mal, was wir hier gemacht haben, das Ganze, innerhalb von beispielsweise einer Woche umgesetzt, erspart uns irgendwie den Aufwand von sonst 2 Monaten.“ Also, da muss man dann so ein bisschen eben die Benefits darstellen und das ganze Thema, ja, mehr oder weniger unter die Leute bringen. Je nach Unternehmensgröße muss man dann halt gucken, ob das schon genügt, ob man damit sozusagen weitere Abteilungen dazu animieren kann, bei dem Thema auch dabei sein zu wollen. Ansonsten muss man halt gucken, dass man vielleicht noch 1 – 2 weitere Use Cases umsetzt, damit man so eine Art Blumenstrauß an Lösungen hat, auch vielleicht, um die Vielseitigkeit zu zeigen. Genau, und wichtig ist aus meiner Sicht eben immer, dass man das Ergebnis teilt, dass man den Erfolg feiert, einfach um Lust in anderen Unternehmensteilen zu schaffen.

EB07\_Janke, Tino: 43 - 43 (0)

Sobald man ausreichend Interesse und Bewusstsein für die Plattform gesammelt hat in den anderen Unternehmensteilen, kommt man, glaube ich dahin, dass man die Strukturen aufsetzen muss, also wir

EB06\_Hesse, Moritz

Der Experte sieht als wichtige Vorbereitende Maßnahme die Notwendigkeit, eine entsprechende Motivation für die Plattform zu schaffen. Dies kann dadurch geschafft werden, dass am Ende der Prototypen-Phase eine unternehmensrelevanter Anwendungsfall vorhanden ist, der den Mehrwert der Plattform widerspiegelt. Abhängig von der Unternehmensgröße kann es erforderlich sein, dass mehrere Anwendungsfälle erstellt werden müssen. Eine weitere vorbereitende Maßnahme ist die erfolgreiche Implementierung der Rahmenbedingungen für die Nutzung der Plattform. Es sollte ebenfalls bereits begonnen werden, die Befähigung von Mitarbeitern zu planen und welche Tools dafür genutzt werden sollen. Für diese Befähigungsmaßnahmen sollten entsprechende Administrationsrollen eingeplant und mit entsprechenden personellen Ressourcen besetzt werden.

EB07\_Janke, Tino

bewegen uns jetzt weg von 1 bis 3 Einzelfalllösungen, sondern wir wollen jetzt das Thema organisationsweit ausrollen und da muss ich mir dann Gedanken machen, wie stell ich denn die Governance zum Beispiel auf für diese Plattform? Also was sind die Leitplanken, in denen ich die Fachabteilungen berechnen, etwas umzusetzen? Was sind denn die Leitplanken, wo die IT-Abteilung immer noch unter die Arme greift, weil, es wird nicht ohne einander gehen, weil da beide Abteilungen müssen sozusagen zusammenarbeiten? Ich muss mir Gedanken machen über eine Community. Wie ich das Wissen in der Organisation verteile und gegebenenfalls ist das auch der Punkt, wo feste Rollen für dieses Thema Low Code/ No Code beziehungsweise Citizen Development im Unternehmen ins Spiel kommen. Also, beim Thema Community, wer managt die Community? Beim Thema Governance, wer arbeitet das aus, stimmt das mit den einzelnen Abteilungen ab, wenn die IT-Abteilung die Governance vorgibt, kann es sein, dass die Fachabteilung so ein bisschen hinten runterfällt? Wenn die Fachabteilung Governance vorgibt, wird ziemlich sicher sein, dass das mit der IT-Abteilung nicht zu hundert Prozent abgestimmt ist. Im besten Fall findet man jemanden, der weder in der einen noch in der anderen Welt zu Hause ist, sondern der sich wirklich in Zukunft um dieses Thema Citizen Development Low Code/ No code kümmern wird. Das wäre also so eine erste feste Rolle, die man im Unternehmen dann finden würde.

EB07\_Janke, Tino: 43 - 43 (0)

Ja, ich sag mal, von der Einzelapplikation, Fachabteilung, das ist ja genau die Sache, die mit diesem Ideationworkshop, zumindest initial, angedacht werden soll. Das heißt, wer sind die Stakeholder? Wer sind die Beteiligten? Wo sind die Touchpoints? Wie ist die Customer Journey? Oder die Product Journey? Je nachdem, welche Perspektive man einnehmen kann. Dass es die Userstories dafür gibt, dass es die Personas dafür geben sollte, die man aufzeichnet. Das

Der Experte sieht für Vorbereitung der Skalierung als wichtigen Bestandteil die Etablierung von Befähigungsworkshops an.

Ein weiterer Aspekt, den der Experte als wichtig ansieht, ist die Klärung, welche Stakeholder in den Abteilungen es gibt? Welche relevanten Anwendungsfälle der Abteilungen existieren? Diese und weitere relevante Faktoren sollten im Vorfeld evaluiert werden.

sind alles, ich will nicht sagen Banalitäten aus dem Bereich des Service Designs, aber es sind alles auf jeden Fall die Pflichtaufgaben, die man erfüllen muss.

EB08\_Fey, Peter: 49 - 49 (0)

Und wenn ich dann erkenne, in der Anlage dieser Applikation auch schon im minimal viable product. Das soll eigentlich im Endeffekt da und dazu führen, verwendet werden, dass wir uns auch austauschen, mit dem Einkauf, mit dem Lager, mit der Produktion, dann sind diese Dinge zumindest vorzubereiten. Und diese Skalierung findet dann dadurch statt, dass man innerhalb des Entwurfs dieses Programms zumindest diese Schnittstellen mit bedenkt. Wenn jetzt dann, sagen wir mal, klar, eine andere Abteilung dazu kommt und sagt: "Wir haben es erst abgelehnt und wollten es gar nicht haben, aber jetzt sehen wir den Vorteil. Wir wollen da auch rein." Dann muss man halt schauen, wie man dann über einen entsprechenden Workshop, über den Austausch, den man dann noch moderieren kann, wie man dann von einer gegebenen Lösung zu einer anderen Lösung kommen kann. Und das ist wie in der jetzigen IT auch, also das sind dann Entwicklungsaufgaben für den Citizen Developer, der dann an der Tastatur sitzt und sagt: "Das und das kann ich umsetzen."

EB08\_Fey, Peter: 49 - 49 (0)

Es sollte ebenfalls eine Analyse der relevanten Schnittstellen erfolgen, die es mit anderen Abteilungen wie Einkauf, Lager oder Produktion gibt.

Der Experte hebt hervor, dass es wichtig ist, für die Skalierung die relevanten Peripherie-Abteilungen mit zu bedenken und Schnittstellen zu diesen zu bilden. Dadurch kann eine Skalierung vorbereitet werden und wenn neue Abteilungen hinzukommen, dann sollte ein Befähigungskonzept entwickelt sein, wie die entsprechenden Abteilungen angebunden und befähigt werden.

EB08\_Fey, Peter

13 - Befähigung Mitarbeiter\13.1 - Befähigung durch Schulung/ Workshop

Also, typischerweise machen wir das auf zwei Ebenen, würde ich jetzt mal grob sagen. Es gibt so ein bisschen so mehr so Einführungsveranstaltungen. Wir stellen vor, was im Piloten so gelaufen ist. Was wir für tolle Apps gebaut haben und so weiter vielleicht. Um erst mal Leute dafür zu begeistern, so ein bisschen Projektmarketing zu machen und wir planen und bauen natürlich so eine Art Schulungskonzept auf, in Form von begleitenden Schulungen. Also manchmal auch so, dass wir sagen: "Ok, ihr müsst die Schulung machen, dann kriegt ihr Zugriff auf die Plattform." Also dass

Der Experte sieht als eine Befähigungsform für Mitarbeiter einen Einführungsworkshop oder Einstiegsschulung als förderlich an. Während dieses Einstiegsworkshops wird eine erste Befähigung auf die Plattform vorgenommen. Eine weitere Zielstellung dieser Schulung ist es, die Mitarbeiter auf die vorhandenen Unternehmensrichtlinien für Governance, IT-Security, Umgang mit Daten und weitere zu befähigen. Diese Schulung wird als Voraussetzung konzipiert, um mit der Plattform arbeiten zu können. Dadurch soll einer falscher Umgang mit der Plattform oder eine

man wirklich sagt: Ein bisschen Know-how muss da sein. Wie gesagt, man kann schon bei der Mächtigkeit der Plattform relativ schnell an Stellen kommen, wo man auch wirklich Unsinn machen kann. Deswegen kann man das eigentlich ganz gut so machen, dass man sagt: „Ok wir, machen eine Schulung und Bestandteil der Schulung ist eben, zum Beispiel, auch so ein bisschen die Unternehmensrichtlinien, Governance, wie gehen wir damit? Warum dürft ihr bestimmte Dinge nicht machen und warum sind die abgeschaltet? Auch wenn ihr die woanders mal gelesen habt.“ Dass man das versucht zu begründen. Ich bin da immer eigentlich ein Freund von einer relativ offenen Kommunikation. Das ist nicht in allen Unternehmen so, da wird einfach abgeschaltet, dann hofft man, das keiner merkt, dass es auch etwas anderen gibt. Würde ich aber tatsächlich nur eine Schulungsbegleitung machen. Wie das dann konkret aussieht, ob man das applikationsbasiert, übergreifend macht, sei mal dahingestellt. Das hängt dann sehr vom Einzelfall im Unternehmen ab. Also ohne Schulung würde ich es nicht machen.

EB01\_Eckhard Hauenherm: 69 - 69 (0)

Ich würde trotzdem immer noch Schulungen für komplexere oder für so einen Gesamtüberblick immer noch anbieten. Die kann man dann tatsächlich, also wenn das Unternehmen keine Personalabteilung hat oder keine eigenen Trainer hat, dann sollen sie sich halt Trainer kaufen. Ich würde das immer, das ist jetzt meinem Geschäft auch so ein bisschen mit geschuldet, ich würde es immer mit einem Live-Training machen, um einfach besser auf Fragen von Teilnehmern und so eingehen zu können. Es gibt natürlich ganz viele Video-Anleitungen und so was und Blogs und so weiter, wo man sich vieles zusammensuchen kann. Aber schöner ist natürlich, wenn man so was beispielhaft mit einem Live-Training und mit Fragestellungen und so weiter

Missachtung der Unternehmensrichtlinien vermieden werden.

Die Form der Schulung sollte nicht als Videokurs erfolgen, sondern mit einem Trainer, der als Ansprechpartner für Fragen vorhanden ist. Dies schließt nicht aus, dass es kein Webmeeting sein kann.

EB01\_Eckhard Hauenherm



durchführen kann. Das sollte man einfach als kontinuierliches Schulungsangebot, also die meisten Unternehmen haben ja so was wie einen Schulungskatalog, da kann sowas dann tatsächlich durchaus drin sein. Und wie gesagt, das kann gebunden sein an die Zulassung auf die Plattform oder nicht. Dass man einfach sagt: „Ok, wenn du das machen willst, dann mach die Schulung und dann lassen wir dich zu und dann darfst du Power Automate benutzen.“

EB01\_Eckhard Hauenherm: 71 - 71 (0)

Dafür haben wir ebendiesen dedizierten Stream, der sagt: "Wir schlaun jetzt mal Entwickler auf." Das können interne Entwickler sein. Das können aber auch Citizen Developer sein, die dann als Multiplikatoren funktionieren. Also idealerweise identifizieren wir im Vorfeld die Champions beim Kunden. Was wären denn unsere Vorreiter? Klassischerweise die Leute, die die besten Excelsheets bauen, wo die komplexesten Formeln drin sind oder die Leute, die PowerPoints bauen und Animationen und allen Features, die wirklich die Tools selber ausreizen. Das sind die Champions, die wir initial identifizieren möchten, wo wir sagen: "Ok, wir geben euch jetzt mal ein Bootcamp für Power Platform, wie baut ihr denn Applikationen. Und ihr tragt dieses Wissen dann in eure Teams weiter. Dass ihr dann entsprechend die zweite Generation, dritte Generation an Citizen Developern ausbilden könnt." Und auf der anderen Seite auch als Champions in der Community agiert, die da sagen: "Wir stellen für euch die entsprechenden Informationen zusammen. Wir hosten alle zwei Monate einen Hackathon, wo wir gemeinsam mit den restlichen Mitarbeitern Applikationen bauen, die dann auch solche Walk-in-Clinics zum Beispiel betreiben, dass man sagt: "Jeden Donnerstag zwischen 12 und 14 Uhr ist Sprechstunde. Alle Citizen Developer können sich in ein Teammeeting einwählen und die Champions sind da, um Rede und Antwort zu stehen." Also wir arbeiten da viel mit Multiplikatoren, weil, du wirst nie

Der Experte sieht es als sinnvoll an, initial eine intensive Schulung mit ausgewählten Entwicklern und Citizen Developern durchzuführen. Inhalt der Schulung ist eine umfangliche Befähigung auf die Plattform. Diese ausgewählten Citizen Developer und Entwickler sollen im weiteren Verlauf als Multiplikatoren fungieren und weitere Befähigungsangebote betreiben, unterstützen und halten. Für die Identifikation der Citizen Developer sollten Mitarbeiter gesucht werden, welche vorhandene Tools wie Excelsheets bereits tiefgreifend nutzen können oder weitreichendes technisches Verständnis haben. Diese Schulungen können auch einen Workshopcharakter haben. Der Experte benennt dies mit Hackathon. Bei diesem Workshop stellt der Experte eine Plattform, wo Mitarbeiter mit Problemstellungen hineinkommen können und gemeinsam mit Kollegen, die auch zu diesem Workshop erschienen sind, eine Lösung zu entwickeln, dabei steht der Experte beratend zur Verfügung. Die Lösungsentwicklung soll aber durch die Mitarbeiter erfolgen.

EB03\_Fabian Gackstatter

ein Enduser-Training für ein ganzes Unternehmen organisieren können. Ist auch nicht erforderlich, weil nicht jeder User wird zum Citizen Developer werden an der Stelle. Und, wie gesagt, wir versuchen da den sauberen Grundstein zu legen mit den Champions. Die zu befähigen, damit sie es entsprechend weitertragen können und auch als Vorbilder agieren können.

EB03\_Fabian Gackstatter: 59 - 59 (0)

Ein Hackathon ist für uns, dass wir sagen: "Wir organisieren jetzt mal ein - wie beschreibe ich das - wie einen Flashmob." Also ich stelle den zur Verfügung. Wir kommen mit einem Team, fünf, sechs Leute, mit Entwicklern, mit SME's und geben dann quasi ein offenes Mikrofon für die Leute, die sagen: "Ok ich habe da eine Idee, die würde ich gerne umsetzen. Können wir das heute zusammen machen?" Und dann werden entsprechende Teams geformt, je nachdem, wie viele Leute zur Verfügung stehen, um dann eben diese Ideen innerhalb von einem Tag umzusetzen und zu einem Prototyp-Level zu bringen. Also, wenn jetzt ein User sagt: "Ich hätte hier eine Idee für einen internen Prozess, ich schaff das aber alleine nicht." Dann wird eben ein Team geformt, die dann gemeinsam anfangen, diese Applikation zu designen, zu implementieren, die dann durch unsere Techniker und SME's unterstützt wird im Laufe des Tages und am Ende des Tages gibt es dann eine Abschlusspräsentation, wo dann jedes Team entsprechend seinen Prototyp vorstellt, was sie denn heute im Laufe des Tages gebaut haben. Das ist halt eine super Gelegenheit, um zum einen Wissen zu vermitteln, aber auch um einfach Netzwerke zu bilden, dass sich Leute mal austauschen können, dass Ideen ausgetauscht werden, so diese gegenseitige Befruchtung. Also das Team A, die hatten da eine super Idee, die könnten wir in unserer Abteilung auch gebrauchen. Da hat noch nie einer dran gedacht. Ich bau das jetzt auch und dafür vernetze ich mich mal mit den Jungs aus Team A, die das da jetzt im Hackathon

schonmal vorgebaut haben. Also das ist so der Sinn von einem Hackathon, um Wissen zu vermitteln und auch die Community voranzubringen, um zu zeigen, ihr seid da nicht alleine als Citizen Developer, da gibt es unheimlich viele Leute, die das auch gerne machen. Und das ist halt auch so eine KPI, wo ich sehen kann: Wie viele Leute melden sich denn an für so einen Hackathon? Sind das Zehn? Sind das nur meine zehn Champions, die ich vielleicht habe oder kommen da zweihundert Leute, die im Zweifelsfall nur zugucken wollen. Die mal sehen: "Hier ich könnte mir das gut vorstellen. Kann ich in so einem Team einfach mal zwei bis drei Stunden zugucken, wie baut ihr denn eine App?"

EB03\_Fabian Gackstatter: 69 - 69 (0)

Gut, man kann ja überlegen, also manche dieser Hersteller bieten ja auch so Zertifizierungsplattformen an. Das ist ja auch noch mal so ein ganz nettes goody, dass man als Mitarbeiter quasi so eine Zertifizierung, also wie beim Scrum Master, gibt es eine Zertifizierung, aber für diese Plattformen auch. Das ist noch ein guter Punkt.

EB04\_Göttl, Simone: 53 - 53 (0)

Und sonst, wie gesagt, Communitys, ganz klassische Trainings und was man sich auch anschauen sollte, das klingt jetzt wieder aber YouTube- Videos, also es gibt ein paar Koryphäen für einzelne Plattformen und wenn die erzählen, was möglich ist, dann muss ich persönlich sagen, da bin ich mit einem 5-Minuten-Video schneller, als mich einmal selbst durchzuprobieren und das auch natürlich zu kommunizieren und sagen, das sind so die best of links.

EB04\_Göttl, Simone: 53 - 53 (0)

Sobald das der Fall ist, ist auch die Akzeptanz groß und man lernt dann auch Learning by doing, in dem Fall, und natürlich klassischerweise würde man noch eine Schulung geben, bevor es dann losgeht mit dem neuen System. Aus meiner Erfahrung ist es so, wenn man jetzt sagt,

Die Expertin sieht klassische Trainings als Befähigungsmöglichkeit als förderlich an, gegebenenfalls mit einem Zertifizierungsprozess im Anschluss.

EB04\_Göttl, Simone

Der Experte sieht den Bedarf für eine initiale Schulung als gegeben an, weist jedoch darauf hin, dass der Großteil der Befähigung der Mitarbeiter in der Abteilung beim Selbstlernen passieren sollte. Dies wird erforderlich, weil aus seiner Erfahrung eine einfache Schulung nicht reicht, um alle Fragen zu

ich plane jetzt für die gesamte Abteilung einen halben Tag Schulung ein, dann sind die fit. Das funktioniert nicht, da sind Ängste dabei bei den Leuten. Können die das leisten? Wie sieht der Prozess aus? Was passiert, wenn? Das sind alles Fragestellungen, die dann so hochkommen. Die kann man in der Regel nicht beantworten mit so einer Frontalbesprechung. Es macht mehr Sinn, wenn möglich, die vor Ort mit einzubeziehen und diese Dinge im täglichen Rahmen, auf kleiner Basis zu klären und dadurch eine Akzeptanz zu schaffen. Auf jeden Fall: Stichwort hier ist, mit einbeziehen und nicht nebenher entwickeln und dann sagen: "Tam tam, die sind fertig. Jetzt wird alles gut. Seht her." Das wird in der Regel nicht funktionieren, da ist zu viel Widerstand da.

EB05\_Rohrbeck, Christian: 71 - 71 (0)

Wir haben einerseits ein Format, das nennt sich smapCademy. Das ist die Möglichkeit, sehr gezielt mit Hilfe von geschulten Trainern, das Wissen zur Anwendung gezielt erlernen. Mit Kursen, mit Prüfungen etc. pp. Das heißt, das ist dieser erste initiale Wissenstransfer von uns als Anbieter, wie kann ich dieses Produkt überhaupt benutzen auf den Anwender.

EB06\_Hesse, Moritz: 79 - 79 (0)

Das heißt, die Menschen müssen, wie bei den IT-Programmen auch, nicht befähigt werden, mit Kobold umzugehen oder mit C++ oder was auch immer. Sondern die müssen befähigt werden, mit der Applikation umzugehen, die hieraus erstellt worden ist. Das muss man tun über übliche Schulungen, das sind halt dann natürlich auch die Touchpoints, die man bedenken muss. Wo kommt eigentlich ein Stakeholder mit dieser Applikation in Berührung. Was muss er wissen? Wie muss er sich verhalten? Was müssen wir ihm auch bieten, anbieten an Eingabemöglichkeiten? Und dass man das dann eh schnell ändern kann, ist halt einer der Vorteile von No Code/ Low Code, aber grundsätzlich, die müssen befähigt

beantworten. Der Mitarbeiter muss nicht nur lernen, wie er die Plattform bedient, sondern wie die Prozesse funktionieren und wie er sich in bestimmten Szenarien verhalten soll.

EB05\_Rohrbeck, Christian

Der Experte sieht ein Schulungsformat als nützlich an. Bei diesem Format gibt es einen Trainer, der für Fragen bereitsteht. Er weist darauf hin, dass es ein weitreichendes Schulungsangebot geben sollte, zu verschiedenen Themen, eventuell mit Prüfung.

EB06\_Hesse, Moritz

Der Experte hält Schulungen für sinnvoll und weist darauf hin, dass die Mitarbeiter befähigt werden müssen, mit der Applikation umzugehen und nicht mit der Plattform. Diese Schulungen sollten sich damit beschäftigen, dieses Wissen zu vermitteln sowie welche Rahmenbedingungen und Prozesse im Kontext mit den Applikationen existieren.

EB08\_Fey, Peter

werden, mit der Applikation umzugehen und nicht mit der Plattform.

EB08\_Fey, Peter: 51 - 51 (0)

Jetzt nehme ich mal wieder diese Chef-Haltung ein, die ich eben angedeutet hatte. Wir stellen kluge Leute ein, damit sie uns sagen, was wir zu tun haben. Jetzt gehe ich genau diesen Weg mal. Das heißt, innerhalb der Anlage dieser Applikation kommen halt auch Fragestellungen auf. Wie kann ich das denn lernen? Wie kann ich mich denn an die Wege erinnern? Dazu müssten innerhalb des Teams die klugen Leuten, die ja für die Fachbereiche schon arbeiten, müssten Lösungen geschaffen werden. Ob man dann sagt: "Mensch, die Applikation, die ist jetzt schon so umfangreich geworden", das es ja sein kann. Vom Einkauf bis zur Abrechnung ist ja schon relativ viel, was man da alles bedenken muss. Das ist jetzt zu viel geworden, das muss jetzt richtig beschult werden, sonst verlieren wir die Anwender. Dann ist das halt eben so, dann muss eine richtige Schulung durchgeführt werden. Wenn aber gesagt wird: "Mensch, hier hilft jetzt einfach ein Sheet", dann geht das auch, dass man einfach so eine Art Spickzettel hat und sagt: "Ok, wenn ich das und das machen will, dann muss ich so und so vorgehen. Alles andere wird mir dann in der Applikation erklärt." Das kann man ja auch dann durchführen, dass man dann das sensitiv macht. Wo die Maus sich befindet, dass man Erklärungstexte einblendet. Das ist eine Sache, die dann durch das Team besprochen wird, das das dann auch realisiert.

EB08\_Fey, Peter: 53 - 53 (0)

13 - Befähigung Mitarbeiter\13.2 - Befähigung durch Community

Dafür haben wir ebendiesen dedizierten Stream, der sagt: "Wir schlauen jetzt mal Entwickler auf." Das können interne Entwickler sein. Das können aber auch Citizen Developer sein, die dann als Multiplikatoren funktionieren. Also idealerweise identifizieren wir im Vorfeld die Champions beim Kunden. Was wären denn unsere Vorreiter? Klassischerweise die Leute, die die besten Excelsheets bauen, wo die komplexesten Formeln drin sind

Der Experte sieht das Bilden von Communitys als sinnvoll an. Diese Communitys bestehen aus den Nutzern der Plattform. Dies ist im Problemfall die erste Anlaufstelle für den Nutzer, wo er eine Lösung finden kann. Das soll dazu beitragen, dass nicht zuerst die Support-Organisation angerufen wird oder die interne IT-Abteilung und somit die Aufgabenlast im Zusammenhang mit der NCLC-Plattform zu groß wird.

oder die Leute, die PowerPoints bauen und Animationen und allen Features, die wirklich die Tools selber ausreizen. Das sind die Champions, die wir initial identifizieren möchten, wo wir sagen: "Ok, wir geben euch jetzt mal ein Bootcamp für Power Platform, wie baut ihr denn Applikationen. Und ihr tragt dieses Wissen dann in eure Teams weiter. Dass ihr dann entsprechend die zweite Generation, dritte Generation an Citizen Developern ausbilden könnt." Und auf der anderen Seite auch als Champions in der Community agiert, die da sagen: "Wir stellen für euch die entsprechenden Informationen zusammen. Wir hosten alle zwei Monate einen Hackathon, wo wir gemeinsam mit den restlichen Mitarbeitern Applikationen bauen, die dann auch solche Walk-in-Clinics zum Beispiel betreiben, dass man sagt: "Jeden Donnerstag zwischen 12 und 14 Uhr ist Sprechstunde. Alle Citizen Developer können sich in ein Teammeeting einwählen und die Champions sind da, um Rede und Antwort zu stehen." Also wir arbeiten da viel mit Multiplikatoren, weil, du wirst nie ein Enduser-Training für ein ganzes Unternehmen organisieren können. Ist auch nicht erforderlich, weil nicht jeder User wird zum Citizen Developer werden an der Stelle. Und, wie gesagt, wir versuchen da den sauberen Grundstein zu legen mit den Champions. Die zu befähigen, damit sie es entsprechend weitertragen können und auch als Vorbilder agieren können.

EB03\_Fabian Gackstatter: 59 - 59 (0)

Die Communitys sind insofern wichtig, weil sie die erste Anlaufstelle für die User sein sollten. Dass eben ein User, der eine Frage hat oder Probleme hat, nicht als erstes zum Telefon greift und den internen Support anruft, die ihm im Zweifelsfall sowieso nicht helfen können, sondern dass es wirklich interne Communitys gibt oder externe Communitys auch, in denen dann Fragen beantwortet werden können, von einem Citizen Developer zum anderen. Wenn es um dedizierte Implementierung geht, wenn es einfach um Best Practices geht. Oder

Diese Communitys können intern im Unternehmen etabliert werden oder extern zum Beispiel in sozialen Netzen. Hier können verschiedene Fragestellungen platziert werden, vom Einsteigerlevel bis zu Fragen für eine Lösungsentwicklung.

EB03\_Fabian Gackstatter

auch neue Leute, die fragen: "Wie fange ich denn am besten an? Was wären denn für mich die besten Einstiegs- punkte? Was sind denn so ein paar Ein- steigertipps, zum Beispiel?" Und die sind in der Regel selbst organisiert. Das heißt, die werden initial zur Verfügung gestellt von einer Plattform her. Die In- halte kommen dann aber durch die User selber und können im Zweifelsfall auch durch die interne IT unterstützt werden. Das heißt, dass die zum Beispiel als Mo- deratoren agieren oder auch als Champi- ons. Die sagen, als Teil meines Develop- perjobs baue ich auch ab und zu einfach mal No Code/ Low Code-Applikationen und stelle die zur Verfügung. Packe die auch in einen entsprechenden Applikati- onskatalog und hoste dann auch mal eine Session, wo ich zeige, wie habe ich denn diese Applikation tatsächlich ge- baut, um neuen Citizen Developern oder auch bestehenden Citizen Developern einfach neue Techniken beizubringen.  
EB03\_Fabian Gackstatter: 61 - 61 (0)

Aber was wir auch festgestellt haben, was sehr wichtig für eine Einführung ist, das Stichwort, was man unter Communi- ties festmacht. Das heißt, zum Beispiel, eine Kollaborationsplattform wie Slack oder Teams oder so zu nutzen und dort eine Community für diese No Code/ Low Code-Plattform zu bilden. Das heißt, dort werden einerseits Informationen bereit- gestellt, was ich schon kann, vielleicht auch erste Trainings angeboten. Ande- rerseits schafft man es darüber auch dann, wenn Kolleginnen und Kollegen Herausforderungen haben in der Nut- zung: "Ich habe mal eine Frage?" Dann sind dort ein paar hundert, die man ein- fach mit einem Schlag fragen kann und die darauf antworten können. Ganz ein- fach, wie in einer WhatsApp-Gruppe/ - Chat am Ende, und was man natürlich auch machen kann, wenn da sich ein paar aktiv in der Community anmelden. Also es sollten nicht alle sein, aber man sollte sich dafür aktiv anmelden. Man schafft es natürlich dort dann auch viel- leicht, die nächste Testgruppe für ein- zelne Features zu finden und denen

Die Expertin hebt bei der Befähigung von Mitarbeitern das Werkzeug der Communitys hervor. Diese Communitys sind als Befähigungsplattform gestaltet, in der viele verschiedene Nutzer der Plattform zusammenarbeiten. Dies dient zur Informationsbereitstellung, dem Anbieten von Trainings sowie dem Support bei der Lösungsentwicklung. Hat also ein User ein Problem mit sei- ner Anwendung, kann er in die Com- munity hinein sein Problem beschrei- ben und dann erhält er unter Umstän- den Lösungsideen durch andere Mit- glieder Community.  
EB04\_Göttl, Simone

vorab schon Sachen als nächste Pilotgruppe bereitzustellen. Also, wenn ich vielleicht eine Bot-Funktion noch nicht aktiviert habe, eine AI-Funktion noch nicht, die das dann mitgehen zu lassen. Das ist ein wichtiger Punkt, der auch sehr gut funktioniert und der, aus eigener Erfahrung ich schon festgestellt habe, dass das ziemlich cool ist.

EB04\_Görtl, Simone: 51 - 51 (0)

Und sonst, wie gesagt, Communitys, ganz klassische Trainings und was man sich auch anschauen sollte, das klingt jetzt wieder aber YouTube- Videos, also es gibt ein paar Koryphäen für einzelne Plattformen und wenn die erzählen, was möglich ist, dann muss ich persönlich sagen, da bin ich mit einem 5-Minuten-Video schneller, als mich einmal selbst durchzuprobieren und das auch natürlich zu kommunizieren und sagen, das sind so die best of links.

EB04\_Görtl, Simone: 53 - 53 (0)

Und das Zweite, wie können sich die Anwender untereinander helfen. Da gibt's ja auch diese verschiedenen Community of Interest, Community of Experts oder so was. Ich habe sie gerade nicht mehr alle parat, aber genau in diesen Bereich gehen wir auch rein, mit einer zweiten Initiative, und zwar bauen wir gerade auf eine Community für smapOne, wo sich quasi die Anwender, natürlich insbesondere intensive Anwender, also vor allem Leute, die das sehr, sehr intensiv nutzen und die das sehr leidenschaftlich nutzen, versuchen wir die untereinander zu vernetzen. Und durchaus auch für einen Wissenstransfer untereinander. Also gar nicht mal nur für die Anwender in einem Unternehmen, in einer Organisationseinheit, sondern tatsächlich über Grenzen hinaus. Das sich zum Beispiel Unternehmen der gleichen Branche untereinander vernetzen können. Hey, ihr habt eine tolle Lösung gebaut, könnt ihr uns mal einen Tipp geben, wie ihr das gelöst habt?

EB06\_Hesse, Moritz: 79 - 79 (0)

Der Experte hebt die Bedeutung von Communities als wichtigen Bestandteil für die Befähigung hervor. Diese Communitys können aus verschiedentlichen Gruppen zusammengesetzt sein. Zum Beispiel es kann eine Community mit Entwicklern geben, aber auch eine Community mit interessierten Stakeholdern. Durch diese Communitys soll erreicht werden, dass ein Wissenstransfer zwischen den Nutzern entsteht.

EB06\_Hesse, Moritz



13 - Befähigung  
Mitarbeiter\13.3 -  
Befähigung durch  
Multiplikatoren

Man kann verschiedene Dinge machen. Also man kann tatsächlich intern, wie ich das eben schon mal kurz gesagt habe, so eine Art Multiplikatoren aufbauen. Also manchmal kommt man in der Plattform auch ohne explizite Schulung aus, also nach der Einführung, wenn genug Leute im Unternehmen schon damit gearbeitet haben, kann sich das Know-how tatsächlich auch schon so verbreiten, da muss man nicht unbedingt zwingend eine Schulung machen. Ich würde trotzdem immer noch Schulungen für komplexere oder für so einen Gesamtüberblick immer noch anbieten.

Der Experte sieht Multiplikatoren als eine Maßnahme zu Befähigung von Nutzern an. Diese Multiplikatoren definiert er als erfahrene Nutzer, die bereits ein hohes Maß an Wissen und Erfahrung für die Plattform besitzen.

EB01\_Eckhard Hauenherm

EB01\_Eckhard Hauenherm: 71 - 71 (0)

Dafür haben wir ebendiesen dedizierten Stream, der sagt: "Wir schlauen jetzt mal Entwickler auf." Das können interne Entwickler sein. Das können aber auch Citizen Developer sein, die dann als Multiplikatoren funktionieren. Also idealerweise identifizieren wir im Vorfeld die Champions beim Kunden. Was wären denn unsere Vorreiter? Klassischerweise die Leute, die die besten Excelsheets bauen, wo die komplexesten Formeln drin sind oder die Leute, die PowerPoints bauen und Animationen und allen Features, die wirklich die Tools selber ausreizen. Das sind die Champions, die wir initial identifizieren möchten, wo wir sagen: "Ok, wir geben euch jetzt mal ein Bootcamp für Power Platform, wie baut ihr denn Applikationen. Und ihr tragt dieses Wissen dann in eure Teams weiter. Dass ihr dann entsprechend die zweite Generation, dritte Generation an Citizen Developern ausbilden könnt." Und auf der anderen Seite auch als Champions in der Community agiert, die da sagen: "Wir stellen für euch die entsprechenden Informationen zusammen. Wir hosten alle zwei Monate einen Hackathon, wo wir gemeinsam mit den restlichen Mitarbeitern Applikationen bauen, die dann auch solche Walk-in-Clinics zum Beispiel betreiben, dass man sagt: "Jeden Donnerstag zwischen 12 und 14 Uhr ist Sprechstunde. Alle Citizen Developer können

Der Experte hebt hervor, dass die Ausbildung von Multiplikatoren ein wichtiger Bestandteil von der Befähigungsstrategie sein sollte. Für die Identifikation der Multiplikatoren sollten Mitarbeiter gesucht werden, welche vorhandene Tools wie Excelsheets bereits tiefgreifend nutzen können, oder weitreichendes technisches Verständnis haben.

Die Multiplikatoren können bei Workshops, Schulungen und Communitys Aufbau sowie Betrieb unterstützen. Dies ermöglicht eine teilweise Unabhängigkeit von den Plattformherstellern und der IT-Abteilung. Der Experte macht den Erfolg von Multiplikatoren unter anderem davon abhängig, ob eine entsprechende Kultur im Unternehmen herrscht, bei dem sich Mitarbeiter gegenseitig helfen und aktiv an der Plattform mitwirken wollen oder eher als Konsumenten die IT-Abteilung in die Verantwortung rücken. Der Nachteil, wenn die IT-Abteilung alleine diese Aufgaben übernimmt, ist die Gefahr, dass diese erneut zu einem Bottleneck wird.

EB03\_Fabian Gackstatter

sich in ein Teammeeting einwählen und die Champions sind da, um Rede und Antwort zu stehen." Also wir arbeiten da viel mit Multiplikatoren, weil, du wirst nie ein Enduser-Training für ein ganzes Unternehmen organisieren können. Ist auch nicht erforderlich, weil nicht jeder User wird zum Citizen Developer werden an der Stelle. Und, wie gesagt, wir versuchen da den sauberen Grundstein zu legen mit den Champions. Die zu befähigen, damit sie es entsprechend weitertragen können und auch als Vorbilder agieren können.

EB03\_Fabian Gackstatter: 59 - 59 (0)

Das kommt auf die Unternehmensgröße und die Struktur darauf an, würde ich sagen, ob ich tatsächlich genügend Leute habe in meiner internen IT oder andersherum, habe ich tatsächlich genügend Champions, dass sich das Thema quasi selbst organisiert? Dass die User sich untereinander selber helfen, oder ist es von der Kultur eher so, es wird alles von der internen IT vorangetriebene. Die agieren als die Ansprechpartner und die User konsumieren nur an dieser Geschichte. Das ist so ein bisschen Unternehmenskultur, was dahintersteckt. Wir sehen beides. Also manche Unternehmen, wo die interne IT am Anfang das Forum zur Verfügung stellt und dann zehn Champions nominiert und die schmeißen den Laden dann und holen entsprechend die User ran. Wir haben aber auch Kunden, wo wirklich ein zentrales IT-Forum existiert. Die dann wie ein verlängerter Arm funktionieren, die dann Citizen Developern Fragen beantworten. Ist natürlich nicht ideal, weil, dadurch wird die IT schnell wieder zum Flaschenhals und wenn die Fragen nicht rechtzeitig beantwortet werden können, sinkt natürlich die Motivation auf Seiten der Citizen Developer. Also wir sehen in den Unternehmen, wo wirklich da eine starke Community ist, die sich selber mit Wissen versorgt, da ist die Adaption von der Plattform auch wesentlich höher, weil den Leuten halt geholfen wird, weil da viel eher ein Austausch stattfindet als

	<p>jetzt bei den Unternehmen, wo es klassisch durch die interne IT verwaltet wird. EB03_Fabian Gackstatter: 63 - 63 (0)</p>	
	<p>In jeder Abteilung gibt es, manche nennen es Champions, manche nennen es Leuchttürme, ein oder zwei oder auch drei, je nachdem wie groß die Abteilung oder der Bereich ist, Leuchttürme identifizieren, die auch die frohe Botschaft vorantragen und auch wissen, was diese Tools können und dort sozusagen Botschafter sind, das ist glaube ich eins. EB04_Göttl, Simone: 51 - 51 (0)</p>	<p>Die Expertin sieht Multiplikatoren als positiven Faktor, um eine Motivation für Plattform zu erzeugen und für den Transfer von Wissen. EB04_Göttl, Simone</p>
	<p>Ein Coach, der die Entwicklung die ersten Wochen und Monate über hin begleitet, bis so viel Know-how aufgebaut ist, dass man das selber machen kann. Und zwar nicht in kurzen knappen Schulungen mit Frontalbeschallung, sondern in einer entspannten Art und Weise nebenher. Der sich da nebenbei mit reinsetzt. Kostet mehr, ist klar, aber an der Stelle absolut wertvoll, weil, mit einer Schulung ist es nicht getan. EB05_Rohrbeck, Christian: 31 - 31 (0)</p>	<p>Der Experte sieht einen Multiplikator als wichtig an, der vor allem in der Einführungszeit und dem Beginn des aktiven Betriebs als Berater zur Verfügung steht und Wissen weitergibt. Er hebt hervor, dass Schulungen allein in Frontalbeschallung nicht allein den gewünschten Erfolg herbeiführen. EB05_Rohrbeck, Christian</p>
	<p>Da gibt's verschiedene Varianten und Lösungen, was durchaus auch immer ein Ansatz bei uns ist, um diesen Wissenstransfer, also gar nicht mal die Skalierung der Lösung oder die Diffusion in die verschiedenen Bereiche, sondern mehr so der Wissenstransfer, dass wir so ein Train-The-Trainer-Konzept versuchen herbeizuführen. Das heißt, dass wir sagen, wir versuchen nicht nur die Leute zu befähigen, es selber umzusetzen, sondern darüber hinaus die auch noch zu befähigen, andere Leute zu befähigen. Einfach um da quasi die Abhängigkeit zu verringern, in dem wir quasi sagen, Train-The-Trainer und der soll dann bitte auch die Leute befähigen. Dass wir so einen Multiplikations- und Skalierungseffekt dadurch erzielen. EB06_Hesse, Moritz: 83 - 83 (0)</p>	<p>Der Experte sieht die Rolle von Multiplikatoren für den Wissenstransfer als sehr wichtig an. Zu diesem Zweck werden gezielt Mitarbeiter ausgebildet, welche danach selbst weitere Citizen Developer und Entwickler ausbilden können. Dadurch wird die Abhängigkeit zum Plattformhersteller verringert. EB06_Hesse, Moritz</p>
<p>13 - Befähigung Mitarbeiter\13.4 - Befähigung durch selbst lernen</p>	<p>Was ich gar nicht so schlimm fand, also ich bin eher der Typ, der sagt, ich stehe auch eher auf der Seite der Citizen</p>	<p>Durch den Experten wird darauf hingewiesen, dass die unterschiedlichen Arten existieren, wie Personen befähigt</p>

Developer, also dass man es offen verwenden kann, dass man das aktiv den Leuten anbietet, schmackhaft macht. Dass man da nicht einen Riegel vorschreibt wie: „Du musst aber erst Kurs XY gemacht haben, bevor du darauf entwickeln darfst.“ Es gibt Leute, die lernen so, es gibt Leute, die lernen so, manche wollen gerne einen internen Kurs machen. Manche gehen über YouTube Videos, also so habe ich es auch gelernt. So biete ich es jetzt auch mittlerweile selber an, also es gibt einfach Leute, die schaffen es selber, sich da so schnell einzuarbeiten und auf einen gewissen Stand zu kommen, denen muss man dann nur noch ein bisschen die Leitplanken zeigen und die können dann selber damit arbeiten, andere brauchen halt ein bisschen mehr Unterstützung, müssen da ein bisschen mehr hingeführt werden. Also von daher denke ich, ist es wichtig, dass es wenig, also dass es keine Restriktionen gibt, das Ding zu nutzen an sich, natürlich Governance Restriktionen, was man machen darf, was man nicht machen darf, aber dass es erst mal zugänglich für alle verfügbar ist und dann eher über so eine Experimentierphase geht, wo die Leute das möchte dann einfach selber ausprobieren können.

EB02\_Rosengrün, Robin: 35 - 35 (0)

Und genauso im Enabling, dass ich mir Strukturen suche, die ich anfangs für einen geringen Personenanteil, auch mit einem hohen manuellen Anteil, dann eben gemacht hab, aber dass wir auch Strukturen haben, die sich sofort an alle richten. Ein Beispiel dafür ist bei uns zum Beispiel, dass wir nicht mehr monatlich feste Kurse für PowerApps/ PowerAutomate geben, sondern dass ich einen Videoselfservice-Kurs an irgendeinem Punkt gemacht habe, den die Leute dann selber durchklicken können, das ich dann nicht sagen muss, „Melde dich zum nächsten Kurs an“, sondern „Hier ist der Selfservice Kurs, kannst starten“ und ja, das nimmt natürlich Arbeit ab und ja, dem Kurs ist es egal, ob er 1-mal im Monat oder 1000-mal im Monat geklickt wird. Dass man halt als manueller

werden können und dass für jedes Individuum eine andere Art der Befähigung besser oder schlechter funktioniert. Der Experte favorisiert aus seiner eigenen Perspektive die Anwendung von Selbstbefähigung. Der Experte nennt seine Erfahrung mit YouTube Videos, über die er sich unter anderem selbst befähigt hat. Mitarbeiter, die für diese Befähigungsform geeignet sind, sollten jedoch in irgendeiner Form die Rahmenbedingungen des Unternehmens gezeigt werden. Diese Art der Selbstbefähigung und zeitunabhängigen Lernmethoden ermöglicht es den Mitarbeitern, sich das Wissen anzueignen zu einem der Zeitpunkt, der für sie gut ist und den sie freiwillig gewählt haben. Dies kann zur Motivation beitragen. Durch diese Befähigungsmethode können große Personengruppen befähigt werden, mit einem mittleren Aufwand.

EB02\_Rosengrün, Robin

Trainer oder als persönlicher Trainer nicht mehr schaffen kann. So versuchen wir unsere Strukturen weiter zu professionalisieren. Natürlich schon personell ein Stück weit zu wachsen, aber nicht annähernd mit der Anzahl der Nutzer.

EB02\_Rosengrün, Robin: 55 - 55 (0)

Dafür haben wir ebendiesen dedizierten Stream, der sagt: "Wir schlauen jetzt mal Entwickler auf." Das können interne Entwickler sein. Das können aber auch Citizen Developer sein, die dann als Multiplikatoren funktionieren. Also idealerweise identifizieren wir im Vorfeld die Champions beim Kunden. Was wären denn unsere Vorreiter? Klassischerweise die Leute, die die besten Excelsheets bauen, wo die komplexesten Formeln drin sind oder die Leute, die PowerPoints bauen und Animationen und allen Features, die wirklich die Tools selber ausreizen. Das sind die Champions, die wir initial identifizieren möchten, wo wir sagen: "Ok, wir geben euch jetzt mal ein Bootcamp für Power Platform, wie baut ihr denn Applikationen. Und ihr tragt dieses Wissen dann in eure Teams weiter. Dass ihr dann entsprechend die zweite Generation, dritte Generation an Citizen Developern ausbilden könnt." Und auf der anderen Seite auch als Champions in der Community agiert, die da sagen: "Wir stellen für euch die entsprechenden Informationen zusammen. Wir hosten alle zwei Monate einen Hackathon, wo wir gemeinsam mit den restlichen Mitarbeitern Applikationen bauen, die dann auch solche Walk-in-Clinics zum Beispiel betreiben, dass man sagt: "Jeden Donnerstag zwischen 12 und 14 Uhr ist Sprechstunde. Alle Citizen Developer können sich in ein Teammeeting einwählen und die Champions sind da, um Rede und Antwort zu stehen." Also wir arbeiten da viel mit Multiplikatoren, weil, du wirst nie ein Enduser-Training für ein ganzes Unternehmen organisieren können. Ist auch nicht erforderlich, weil nicht jeder User wird zum Citizen Developer werden an der Stelle. Und, wie gesagt, wir versuchen da den sauberen Grundstein zu legen mit den Champions. Die zu

Der Experte sieht ein gewisses Maß an Selbstbefähigung als wichtig an. Dies wird auf der einen Seite dadurch realisiert, dass die Mitarbeiter in die entsprechenden Workshops wie einen Hackathon kommen und dort lernen, sowie durch das aktive Mitarbeiten in einer Community. Diese Selbstbefähigung geht in der Beschreibung des Experten eng mit anderen Befähigungsmethoden einher.

EB03\_Fabian Gackstatter

befähigen, damit sie es entsprechend weitertragen können und auch als Vorbilder agieren können.

EB03\_Fabian Gackstatter: 59 - 59 (0)

Also es gibt verschiedene Formate. Es gibt dann so Center of Excellence, wo man so was verankern kann. Das heißt, einerseits die Governance verankert, also Prozesse, was darf ich, was darf ich nicht? Was ist erlaubt? Wie erfolgen die Prozesse für Freigaben? Welche Apps darf ich überhaupt programmieren? Da findet man häufig eine gewisse Expertise, dass man sagt: "Man hat so Leuchtturmmitarbeiter, die da natürlich auch helfen können." Nichtsdestotrotz würde ich bei einem Rollout immer die Mitarbeiter mitnehmen. Das heißt, deshalb sage ich auch bewusst mitnehmen, für mich ist das nicht eine einmalige Schulung, sondern für mich ist das eine fortwährende Enablement, Befähigung, was durch Trainings passieren kann, durch einzelne Learningnuggets, YouTube-Videos, aber auch vielleicht durch Communitys. Und deshalb ist es nicht nur Aufgabe des Center of Excellence, sondern muss man einfach auch schauen, wer am Ende auch der Product Owner ist und dass der sein Produkt natürlich durch so einen Konzern und Haus trägt.

EB04\_Görtl, Simone: 19 - 19 (0)

Das andere ist natürlich, ganz klar, Intranetseiten zu haben und darüber zu kommunizieren. Also Intranet ist jetzt vielleicht das eine in der Unternehmenskommunikation. Es gibt ja auch viele Unternehmen, die dann auch wie eine Art eigene interne App haben für die Unternehmenskommunikation, das da zu verkünden.

EB04\_Görtl, Simone: 51 - 51 (0)

Und sonst, wie gesagt, Communitys, ganz klassische Trainings und was man sich auch anschauen sollte, das klingt jetzt wieder aber YouTube-Videos, also es gibt ein paar Koryphäen für einzelne Plattformen und wenn die erzählen, was möglich ist, dann muss ich persönlich sagen, da bin ich mit einem 5-Minuten-

Die Expertin hebt die Wichtigkeit hervor das Mitarbeiter motiviert werden sollten, an Befähigungsmaßnahmen teilzunehmen. Eine der Befähigungsmaßnahmen, die durch die Expertin beschrieben wird, ist die Selbstbefähigung durch das Teilnehmen an Videokursen oder durch das aktive Mitarbeiten in Communitys. Die Befähigung über Videos hat wesentliche Vorteile gegenüber dem Trial-und-Error Verfahren, bei dem Mitarbeiter selbst versuchen, eine Lösung zu erarbeiten. Durch solche Videos kann schnell Wissen vermittelt werden durch Experten, welches benötigt wird, zu dem Zeitpunkt, an dem es benötigt wird. Ein weiterer Faktor der Selbstbefähigung ist das Lesen von themenbezogenen Beiträgen, zum Beispiel im unternehmensinternen Intranet.

EB04\_Görtl, Simone

Video schneller, als mich einmal selbst durchzuprobieren und das auch natürlich zu kommunizieren und sagen, das sind so die best of links.

EB04\_Göttl, Simone: 53 - 53 (0)

Sobald das der Fall ist, ist auch die Akzeptanz groß und man lernt dann auch Learning by doing, in dem Fall, und natürlich klassischerweise würde man noch eine Schulung geben, bevor es dann losgeht mit dem neuen System. Aus meiner Erfahrung ist es so, wenn man jetzt sagt, ich plane jetzt für die gesamte Abteilung einen halben Tag Schulung ein, dann sind die fit. Das funktioniert nicht, da sind Ängste dabei bei den Leuten. Können die das leisten? Wie sieht der Prozess aus? Was passiert, wenn? Das sind alles Fragestellungen, die dann so hochkommen. Die kann man in der Regel nicht beantworten mit so einer Frontalbeschallung. Es macht mehr Sinn, wenn möglich, die vor Ort mit einzubeziehen und diese Dinge im täglichen Rahmen, auf kleiner Basis zu klären und dadurch eine Akzeptanz zu schaffen. Auf jeden Fall: Stichwort hier ist, mit einbeziehen und nicht nebenher entwickeln und dann sagen: "Tam tam, die sind fertig. Jetzt wird alles gut. Seht her." Das wird in der Regel nicht funktionieren, da ist zu viel Widerstand da.

EB05\_Rohrbeck, Christian: 71 - 71 (0)

Wir haben aktuell noch ein eher klassisches Modell, das befindet sich gerade auch selber in der Transformation. Wir haben momentan noch den Fall, dass wir sehr klassische Präsenzvorlesungen virtuell halten, an denen die Leute teilnehmen, die eher kleine Übungsaspekte live beinhalten. Wo dann die Übungen jeder selber zu Hause vornehmen kann und wir wollen das gerade, das fußt so ein bisschen auf meinen Erfahrungen als wissenschaftlicher Mitarbeiter, wo ich mich mit E-Learning und so was beschäftigt habe. Das bauen wir gerade um in ein Flip-Classroommodell, wo genau das umgedreht wird. Wo die Vorlesungen Asynchron als Video-on-Demand-

Der Experte weist darauf hin, das eine Schulung allein nicht als Befähigung reicht. Es sollte den Mitarbeitern die Möglichkeit gegeben werden zu lernen, während sie die Plattform anwenden und einen Ansprechpartner vor Ort zu haben, der Helfen kann, wenn Probleme auftreten.

EB05\_Rohrbeck, Christian

Der Experte sieht es als positiven Faktor für die Selbstbefähigung an, wenn Mitarbeiter sich selbst über Videokurse befähigen können, in einem selbst gewählten Tempo, zu einer selbst gewählten Zeit. Übungen werden dann in Gruppenveranstaltungen durchgeführt, bei denen über die Lösungswege gesprochen wird.

EB06\_Hesse, Moritz

Vorlesungen bereitgestellt werden. Das heißt das theoretische Wissen soll sich jeder selber aneignen in dem eigens gewählten Tempo, zu der eigens gewählten Uhrzeit, an dem selbst gewählten Tag. Und dann gemeinsame Treffpunkte, also Übungseinheiten, wo dann vertiefend im Gemeinsamen darauf eingegangen wird, wie die Lösungen entsprechend gestaltet werden können.

EB06\_Hesse, Moritz: 81 - 81 (0)

14 - Qualitätssicherung Projekt  
14.1 - Qualitätssicherung durch emotionale Aspekte

**B: EH** Gute Frage. Motivation, Motivation, Motivation. Wie mache ich das am besten. Ich überlege gerade. Eigentlich kann man das am besten durch klare Ziele. Man sollte sozusagen nicht in die Situation kommen, wo man nicht weiß, wo man jetzt eigentlich als nächstes hin will. Das ist so der Punkt. Also Motivation aufrechterhalten: Ok, das haben wir jetzt geschafft. Das ist schön. Das ist toll. Jetzt nehmen wir uns das nächste vor, damit das nicht auseinanderläuft. Sonst besteht tatsächlich die Gefahr, dass in so einem Projekt irgendwas erreicht wurde, und dann sagen die Leute: "Mein Teil ist getan. Ich gehe jetzt weg. Ich brauche jetzt nicht mehr." So ungefähr, das ist schon tatsächlich ein Punkt. Das heißt, man muss so ein bisschen dafür sorgen, dass man auch vielleicht auch aus dem geschäftlichen Umfeld einen Nachschub an Anforderungen kriegt. Was könnte man vielleicht machen? Wo könnte man hingehen? Es ist natürlich zu einem gewissen Grad naturgegeben, dass man nach einer ersten Einführungsphase in den meisten Abteilungen erst mal so ein Stadium hat, wo man sagt: „Ok, lass uns das jetzt erst mal eine Weile ausprobieren. Wir wollen jetzt nicht noch mehr Applikationen machen, sondern wir wollen einfach jetzt mal so ein bisschen ausprobieren.“ Das würde ich auch immer so zulassen, dass man sagt: "Ok, dann treffen wir uns in einem Monat wieder, wenn ihr das ausprobiert habt. Guckt mal, wo haben wir Optimierungspotenzial. Sind vielleicht neue Dinge aufgetaucht, die wir angehen können. Und dann machen wir wieder mal eine kleine Iteration, wo wir

Der Experte hebt als Kriterium für eine Qualitätssicherung im Projekt das Thema Motivation hervor. Zu diesem Zweck sollten Erfolge gefeiert werden und neue Aufgaben sollten bearbeitet werden. Dafür ist es nötig, dass regelmäßig neue Aufgabenanforderungen entstehen. Es besteht die Gefahr bei fehlender Motivation, dass Mitarbeiter nach Abschließen einer Aufgabe das Projekt verlassen. Dadurch würde Wissen aus dem Projekt abfließen. Der Experte verweist darauf, dass es Phasen geben kann, wo wenig passiert und dass dort probiert wird, was sind Funktionen oder neue Funktionen. Diese Phasen sollten allerdings unterbrochen sein durch Phasen aktiver Entwicklung, um das Thema nicht auslaufen zu lassen.

EB01\_Eckhard Hauenherm



uns mit der nächsten Applikation beschäftigen können." Solche Sachen, oder: "Wir treffen uns regelmäßig mit der IT und gucken mal, was hat sich so an Aktualisierungen der Plattformen ergeben, die uns neue Dinge gibt." Oder manchmal gehe ich auch einfach ran an das Unternehmen: "Hier, das hat Microsoft gerade gemeldet" oder: "Das können wir jetzt auch machen. Wäre das nicht was für euch? Das passt doch ganz gut, da haben wir doch so lange dran rum gekämpft und das nie geschafft. Aber jetzt soll das angeblich gehen." So was passiert mir auch immer mal. Dass ich sage: "Ok, das geht jetzt viel einfacher. Das können wir jetzt so machen." Das ist aber immer so eine schrittweise Motivation tatsächlich. Also so ein Projekt ist tatsächlich naturgegeben zum Teil so, dass man zwischendurch so Phasen hat, wo tatsächlich wenig passiert. Man muss ja auch einfach eine Möglichkeit haben, das einfach mal laufen zu lassen und zu probieren. Funktioniert das, ist das für uns sinnvoll. Ich kann ja nicht immer wieder neu entwickeln, entwickeln und entwickeln oder so was.

EB01\_Eckhard Hauenherm: 73 - 73 (0)

Jetzt geht es um den Scope und die Qualität dessen, was man erstellt hat. Und das ist gerade bei einer Einführung eine ganz spannende Sache. Also man sollte realistisch davon ausgehen, dass die Qualität auf gar keinen Fall 100 % hat, wenn man ein Softwaresystem in die Produktion einführt. Sondern es wird immer irgendwas geben, was man nicht bedacht hat, weil es einfach von der Komplexität oder von der Zeit her oder den fehlenden Ressourcen nicht anders machbar war. Es wird immer Dinge geben, die als Workaround starten. Wo man sagt: "Wir haben das jetzt gerade so hinbekommen. Hoffentlich klappt alles." Diese Dinge erfordern bei der tatsächlichen Produktionseinführung viel Aufmerksamkeit und das muss man vorab einplanen. Also, ich kann jetzt nur so ideale Sachen geben. In der Regel ist so ein

Ein Teil der Qualitätssicherung ist die Akzeptanz, dass zu Beginn es zu Fehlern kommt. Dies kann aufgrund verschiedener Faktoren entstehen, wie einem zu engen Zeitplan, fehlende Erfahrung mit der Plattform oder fehlende Ressourcen. Zu diesem Zwecke sollte vor allem in der Anfangszeit ein höherer Fokus auf das Monitoring der Anwendungen angewandt werden. Es sollte berücksichtigt werden, dass nach einer intensiven Einführungsphase das Projektteam erschöpft ist und diesem sollte Raum gegeben werden, sich zu erholen.

EB05\_Rohrbeck, Christian

Projektteam ja schon durch, wenn es gerade in der Projektendphase viel gearbeitet hat. Die Leute sind erschöpft, ausgelaugt und wenn dann direkt die Einführung kommt, was oft der Fall ist, dann wirds schwierig. Dann hat man ein Team, was schon so ein bisschen durch ist. Man sollte ganz offen damit umgehen, sagen: "Wir haben jetzt ein neues System. Wir erwarten hier Kinderkrankheiten. Das bedeutet für alle, ihr müsst jetzt Überstunden schieben. Ihr müsst diese Abläufe, diese neuen Automatismen, nachkontrollieren. Funktioniert das so? Ist alles dran gedacht und ihr steht zumindest bereit, für den Fall, dass was sein kann."

EB05\_Rohrbeck, Christian: 73 - 73 (0)

14 - Qualitätssicherung Projekt  
14.2 - Qualitätssicherung durch Monitoring

Ist jetzt auch unabhängig von Power Platform-Projekten, würde ich sagen. Wir haben regelmäßige Statuschecks. Wir haben auch die entsprechenden Quality Gates, wo wir sagen: "Ok, das und das sind jetzt eigentlich die Punkte, die wir adressiert haben sollen."

EB03\_Fabian Gackstatter: 65 - 65 (0)

Gerade so in der Anfangsphase, wo wir viel Wissen mitbringen und das auch an den Kunden übergeben, dass das auch wirklich verinnerlicht wird. Dass der Kunde auch in der Lage ist, dieses Wissen dann zu transportieren und für sich umzusetzen. Und dafür muss man regelmäßige Statuschecks auch einfach machen und auch mal Rückfragen stellen. Nach dem Motto: "Zeigt mir mal, was ihr jetzt mitgenommen habt. Wie habt ihr das jetzt implementiert?" Dass wir auch konstantes Review machen auf die Themen, die dann vom Kunden implementiert werden. Um zu sehen, folgen die auch wirklich den Best Practices oder hat man hier tatsächlich noch irgendwas übersehen. So ein Beispiel wäre eine Environmentstrategie zum Beispiel. Wie muss ich denn jetzt meine Environments aufsetzen. Da haben wir verschiedene Szenarien, die wir mit dem Kunden durchspielen. Der Kunde nimmt das dann mit, kommt mit einem eigenen

Der Experte empfiehlt zur Qualitätssicherung, Qualitätschecks durchzuführen. Diese sollten einen Soll-Ist-Abgleich durchführen für einen spezifischen Zeitpunkt.

Es sollten Rückkopplungsprozesse implementiert werden, welche überprüfen, ob das vermittelte Wissen richtig aufgenommen wurde und ob dieses Wissen mit der entsprechenden Qualität angewendet wird. Hier kann überprüft werden, ob Governancerichtlinien eingehalten werden, Sicherheitsrichtlinien bekannt sind oder Datenschutzregeln mit in die Applikation eingebunden werden.

EB03\_Fabian Gackstatter

Konzept wieder und wir führen dann ein entsprechendes Review durch. Zum einen auf der konzeptionellen Ebene und dann, wenn es auch tatsächlich implementiert ist, dass wir auch da noch mal drauf schauen. Sind wirklich die richtigen Drähte miteinander verdrahtet. Sind da noch irgendwo Schlupflöcher offen, die dann wieder zu Problemen führen könnten.

EB03\_Fabian Gackstatter: 65 - 65 (0)

Jetzt geht es um den Scope und die Qualität dessen, was man erstellt hat. Und das ist gerade bei einer Einführung eine ganz spannende Sache. Also man sollte realistisch davon ausgehen, dass die Qualität auf gar keinen Fall 100 % hat, wenn man ein Softwaresystem in die Produktion einführt. Sondern es wird immer irgendwas geben, was man nicht bedacht hat, weil es einfach von der Komplexität oder von der Zeit her oder den fehlenden Ressourcen nicht anders machbar war. Es wird immer Dinge geben, die als Workaround starten. Wo man sagt: "Wir haben das jetzt gerade so hinbekommen. Hoffentlich klappt alles." Diese Dinge erfordern bei der tatsächlichen Produktionseinführung viel Aufmerksamkeit und das muss man vorab einplanen. Also, ich kann jetzt nur so ideale Sachen geben. In der Regel ist so ein Projektteam ja schon durch, wenn es gerade in der Projektendphase viel gearbeitet hat. Die Leute sind erschöpft, ausgelaugt und wenn dann direkt die Einführung kommt, was oft der Fall ist, dann wirds schwierig. Dann hat man ein Team, was schon so ein bisschen durch ist. Man sollte ganz offen damit umgehen, sagen: "Wir haben jetzt ein neues System. Wir erwarten hier Kinderkrankheiten. Das bedeutet für alle, ihr müsst jetzt Überstunden schieben. Ihr müsst diese Abläufe, diese neuen Automatismen, nachkontrollieren. Funktioniert das so? Ist alles dran gedacht und ihr steht zumindest bereit, für den Fall, dass was sein kann."

EB05\_Rohrbeck, Christian: 73 - 73 (0)

Der Experte hebt hervor, dass ein höheres Maß an Monitoring nötig ist, vor allem in der Anfangsphase des Projekts, da hier verschiedene Fehler auftreten können, aufgrund von mangelnder Erfahrung mit der Plattform oder durch projektbedingte Faktoren.

EB05\_Rohrbeck, Christian

	<p>Ich weiß gar nicht, ob die No Code/ Low Code-Plattformen das mitbringen. Aber das ist dann eine Sache, die ist ein bisschen schwierig nachzuweisen. Meines Erachtens ist es so, hier haben wir Anwender in diesem Bereich, in den Fachbereichen, und die sind dann am ehesten in der Lage zu beurteilen, ob sie selber eine gute und sie befriedigende Arbeit geleistet haben oder nicht. Und wir brauchen auf jeden Fall, das hatte ich ja eben schon ausgeführt, jemand, der mit Verstand auf die Sicherheit dieser Applikation schaut und dann einen Test, den man durchführt, denn ungetestete Software, egal aus welcher Feder sie stammt, auf einem Markt zu haben, geht nicht. Also es kann sein, dass man eine fehlerhafte Software am Markt hat und dass sie nachgearbeitet werden muss. Da muss dieser Fehler sozusagen akzeptiert worden sein, aber die Software selber muss immer getestet werden und sie muss ein Testat bekommen: "Ja, darf raus." Das dürfte aber nicht durch die Fachabteilung passieren, da müsste meines Erachtens, bevor sie dann aufschlägt im Außenuniversum, im Metaverse, müsste dann tatsächlich ein Test durchgeführt werden.</p> <p>EB08_Fey, Peter: 55 - 55 (0)</p>	<p>Der Experte vertritt die Auffassung, dass zur Qualitätssicherung eine Überprüfung der Funktionalität durch die Anwender erfolgen sollte. Im Weiteren sollte außerhalb der Fachabteilung als zusätzliche Instanz eine Überprüfung erfolgen, ob Governance-, Sicherheits- und Datenschutzrichtlinien eingehalten wurden.</p> <p>EB08_Fey, Peter</p>
<p>14 - Qualitätssicherung Projekt\14.3 - Qualitätssicherung durch Feedback</p>	<p>Regelmäßige Feedbackrunden mit dem Kunden: Ist das jetzt auch verstanden? Sind die entsprechenden Arbeitspakete auch implementiert? Haben wir unsere Checkliste, die wir am Anfang setzen, haben wir die tatsächlich abgearbeitet? Sehen wir, dass der Kunde jetzt in der Lage ist, diese Plattform selber zu betreiben? Oder sehen wir, die haben zwar in den Workshops mit dringesessen aber sich nur, wie soll ich sagen, bespaßen lassen, aber es nicht wirklich verinnerlicht, worauf es tatsächlich ankommt.</p> <p>EB03_Fabian Gackstatter: 65 - 65 (0)</p>	<p>Der Experte hebt die Notwendigkeit von regelmäßigem Feedback hervor. Dabei sollte überprüft werden, ob die Plattform richtig implementiert wurde, können die Anwender die Plattform nutzen, kann das Unternehmen die Plattform so selbst betreiben, werden Projektziele und Zwischenziele erreicht.</p> <p>EB03_Fabian Gackstatter</p>
<p>15 - Erfolgsmessung Projekt\15.1 - Quantitative Größen</p>	<p>Also was wir monitoren auf hoher Flugebene ist natürlich die Anzahl der Personen, die die Plattform nutzen. Anzahl der aktiv genutzten Anwendungen lässt sich durch Governance ja auch einfach ermitteln. Also das sind Daten, die erfasst</p>	<p>Der Experte sieht als quantitative Methoden der Erfolgsmessung die Anzahl der aktiven Nutzer der Plattform und die Anzahl aktiv genutzter Anwendungen.</p> <p>EB02_Rosengrün, Robin</p>

werden von der Power Plattform jetzt im Speziellen, aber wahrscheinlich auch von allen anderen.

EB02\_Rosengrün, Robin: 67 - 67 (0)

Ja, es gibt verschiedene KPI's, die wir nutzen. Zum einen, wie viele Applikationen werden denn überhaupt gebaut? Werden die dann tatsächlich auch genutzt? Weil, wir sehen auch bei manchen Kunden, da werden am Anfang da tausende Applikationen gebaut, die dann nach drei Monaten noch als Zombies rum agieren, die aber keinen produktiven Einsatz haben. Wir messen zum einen, wie viel Applikationen gibt es denn? Mit wie vielen Usern sind die geteilt? Wie aktiv werden die genutzt? Wie viele Aktionen werden generiert darüber? Dann in den Communitys selber, wie viele Mitglieder hat denn meine Community? Wie viele Postings gibts denn dann? Wie sieht eine Antwortfrequenz aus, wenn eine Frage gestellt wird? Kommt da innerhalb von einer Woche eine Antwort, kommen da zehn Antworten zurück? Also, es gibt schon verschiedene Indikatoren, die wir auch aus der Plattform selber auslesen können. Direkt, bevor wir irgendwelche User mal befragen müssen, inwieweit ist denn die Plattform angekommen und natürlich auch das Aufkommen von Supporttickets, um zu sehen, funktioniert auch ein Supporthandling sauber. Rufen die Leute tatsächlich alle beim Helpdesk an oder organisieren sie sich wirklich in der Community, die dafür zur Verfügung gestellt wurde? Wie viele Leute partizipieren an einem Hackathon, wenn wir ihn zur Verfügung stellen. Oder in einer Walk-in-Clinic, wie viele Leute kommen tatsächlich und haben Fragen oder haben Lösungen? Und dann später, wenn die Plattform etwas gereift ist bei einem Kunden. Was wir normalerweise auch etablieren, ist ein Applikationskatalog, in dem dann Applikationen, die von Usern gebaut werden, auch anderen Usern zur Verfügung gestellt werden, also ein App-Store und auch ein Store für einzelne Komponenten. Und auch daran kann ich sehen, wie ist denn die Adaptation von meiner Plattform? Habe

Der Experte sieht als Mittel der quantitativen Messung des Erfolgs verschiedenen Messwerte als sinnvoll an.

- Anzahl der erstellten Anwendungen
- Anzahl genutzten Anwendungen
- Anzahl ungenutzten Anwendungen
- Anzahl Nutzer, die bestimmte Anwendungen nutzen
- Anzahl Nutzungsaufrufe der Anwendung
- Anzahl Mitglieder Communitys
- Anzahl Beiträge in den Communitys
- Antwortfrequenz auf Beiträge in Communitys
- Anzahl Supporttickets
- Anzahl Teilnehmer in Workshops
- Anzahl Teilnehmer in freiwilligen Workshops

Für den Fall, dass ein Anwendungskatalog, analog zu einem Appstore, eingeführt wurde:

- Anzahl Anwendungen im Anwendungskatalog
- Messung der Frequenz des Bereitstellens von Anwendungen

EB03\_Fabian Gackstatter

ich da nach sechs Monaten, sind da zwei Applikationen in meinem App-Store oder sind da zweihundert drin, die die Leute gebaut haben und auch die einzelnen Komponenten. Wie viele sind das? Wie ist die Qualität von den Komponenten? Es gibt schon relativ viele KPI's, die man abfragen kann. Eben nicht nur, wie viele User haben denn eine Lizenz bekommen, zum Beispiel.

EB03\_Fabian Gackstatter: 67 - 67 (0)

Wann ist ein Projekt denn erfolgreich? Das ist ja eigentlich im Sinne so: in Time, in Budget, in Scope, hat man ja immer so schön gesagt, im klassischen. Zu Projektabschluss, kann man sagen, hat man all das erreicht, was man in seinem Backlog drin hatte, zu einem gewissen definierten Zeitpunkt, dann ist das abgeschlossen worden und dann bin ich auch, das muss ganz klar sagen, in dem Kostenpart drin.

EB04\_Görtl, Simone: 57 - 57 (0)

Das ist aber eher das Projekt, aber wenn ich jetzt messen möchte, war ich erfolgreich, diese No Code/ Low Code-Plattform im Unternehmen bereitzustellen. Also wird sie danach genutzt, dann habe ich natürlich unterschiedliche Messkriterien, also ich kann mir anschauen, wie viele Apps und Anwendungen werden angefangen, wie viele werden entwickelt und wie viele werden Live geschaltet? Wie ist grundsätzlich die Nutzungsrate

EB04\_Görtl, Simone: 57 - 57 (0)

Als Erfolgsmessung für das Projekt sieht die Expertein die Zielerreichung als wichtigen Faktor der Messung. Dies bedeutet, ist das Projekt in Scope, in Time, in Budget.

Für die Messung der Plattform selbst sieht sie verschiedene Messkriterien:

- Anzahl der Anwendungen, die begonnen werden zu entwickeln
- Anzahl der Anwendungen werden produktiv genutzt
- Nutzungsrate Anwendungen und Plattform

EB04\_Görtl, Simone

Also natürlich, messbare Größen sind natürlich immer Zeit und Budget. Das ist am einfachsten. Das ist auch das, was in der Regel die Stakeholder abfragen, weil das wirklich einfache Größen sind.

EB05\_Rohrbeck, Christian: 73 - 73 (0)

Der Experte sieht als Messgrößen Zeit und Budget.

EB05\_Rohrbeck, Christian

Das ist technisch, also ich sage das jetzt mal nur aus einer rein technischen Perspektive. Technisch messe ich den Erfolg, also ich spreche jetzt mal nur von Integrationsprojekten, ich rede jetzt nicht von reinen, obwohl man kann das jetzt auch mal ohne Integration, solche Projekte betreue ich hier im Unternehmen

Der Experte sieht als Erfolgsmessung vor allem technische Erfolgskriterien. Werden die richtigen Daten verarbeitet, werden die richtigen Daten weiterverarbeitet.

EB06\_Hesse, Moritz

halt weniger. Ich komme immer dann ins Spiel, wenn es darum geht, irgendwie ein Umsystem anzuschließen. Und ich bewerte das als einen Erfolg, wenn der Kunde sagt: Ja, die Daten aus dem Fremdsystem kommen genauso, wie sie sollen, bei mir in meiner Low Code-Lösung an. Ich kann die Lösung effektiv und effizient bedienen. Also insbesondere, dass sie mir hilft, meinen Arbeitsprozess einfach zu bewerkstelligen und dass die Daten dann am Ende wieder sauber ins Systems kommen, und den Erfolg messe ich dann, wenn der Datenfluss sauber gewährleistet ist. Und das ist quasi mein technischer Erfolg dann und der Erfolg für den Kunden, daran bemisst sich dann auch, ob er unser Kunde bleibt oder nicht, ist natürlich die Ersparnis beim Kunden.

EB06\_Hesse, Moritz: 85 - 85 (0)

Zum anderen kann man aber auch dahin übergehen, dass man richtige KPI's definiert, die man eben messen will und wo man dann beide Welten miteinander vergleichen kann, also zum Beispiel eine durchschnittliche App-Entwicklungsdauer könnte man jetzt festlegen und die vergleichen. Man könnte auch, das ist jetzt sehr organisationsbezogene, je nachdem, was man messen möchte und was Sinn macht, aber man könnte auch den gesamten Aufwand einer Applikation messen also, was hat es mich gekostet, diese Anwendungen insgesamt zu entwickeln.

EB07\_Janke, Tino: 45 - 45 (0)

Bei der funktionalen Ebene kann ich relativ schnell sagen, ja, erfüllt oder erfüllt nicht. Bei den nicht funktionalen, da muss man halt Messungen durchführen, Metriken einführen, um zu sagen, ich komme jetzt endlich an die Zeiten ran.

EB08\_Fey, Peter: 55 - 55 (0)

Als quantitative Messmethoden sieht der Experte das bilden von KPI's. Als Beispiele nennt er:

- durchschnittliche Entwicklungsdauer
- Gesamtaufwand zur Entwicklung einer Anwendung
- Kosten pro Applikation

EB07\_Janke, Tino

Der Experte sieht das bilden von Messdaten, welche für das Unternehmen wichtig sind als sinnvoll an.

EB08\_Fey, Peter

15 - Erfolgsmessung Projekt\15.2 - Qualitative Größen

Das messe ich anhand der Ziele. Also deshalb frage ich am Anfang, was wollt ihr damit erreichen. Also wenn jetzt tatsächlich als Ziel herauskommt, wir wollen damit Geld sparen, oder wir wollen damit schneller werden, dann muss ich

Der Experte sieht als qualitative Messgrößen zur Erfolgsmessung, ob Wissen im Unternehmen aufgebaut werden konnte, ob der Nutzungsumfang mit dem in der Zielstellung definierten Nutzungsumfang übereinstimmt. Wird die Plattform durch die Nutzer gerne

mir natürlich überlegen, wie viel schneller sind wir denn geworden. Ich könnte natürlich aber sagen: "Ok, wir haben eine ganze Reihe von Applikationen zum Beispiel umgesetzt." Könnte zum Beispiel Messziel sein. "Das können wir jetzt und das können wir jetzt und das können wir jetzt." Das ist natürlich in so einem Projekt in der Regel, gerade am Anfang, schwer zu messen, ob sich hier das Geld jetzt wirklich gelohnt hat. Weil, da müsste ich tatsächlich gucken, was wir durch die Automatisierung zum Beispiel an Zeiteinsparung haben und solche Sachen haben. Also, wenn ich wirklich klare Zahlen liefern will, muss ich das tatsächlich auf der Ebene messen. Bis jetzt bin ich noch nie wirklich in die Bredouille gekommen, dass der Erfolg über diese Art und Weise gemessen wurde. Also, dass ich wirklich klare Zahlen als KPI's sozusagen liefern musste, für den Erfolg des Projekts, sondern es war tatsächlich der Know-how-Aufbau im Unternehmen und der Umgang mit der Plattform und den Nutzungsgrad, ist sozusagen der Erfolgsfaktor. Also wird es tatsächlich eingesetzt? Arbeiten meine Leute damit? Haben wir die motiviert, das zu machen? Und kriegen die damit ihre Prozesse in den Griff? Und haben wir unsere Risiken dabei auch im Griff? Das ist natürlich auch ein Punkt, was ich hier eingangs ja schon sagte, das muss natürlich auch noch der Grund sein. Das sind eigentlich die Faktoren, an denen ich in dem Projekt gemessen werde.

EB01\_Eckhard Hauenherm: 75 - 75 (0)

In dem man guckt, ob die Applikation auch fertig geworden ist. Ob sie auch eingesetzt wird. Läuft der Prozess, wird der tatsächlich genutzt. Das hängt jetzt ein bisschen davon ab, in welche Richtung es geht. Aber man hat ja durchaus in den Plattformen typischerweise ein Reporting, das so ein bisschen zeigt, wie oft so was genutzt wird, wie viele Leute damit arbeiten, wie oft etwas geteilt wird und so weiter. Das kann man schon so

genutzt und werden Prozesse dadurch real verbessert? Wie wird mit Risiken umgegangen? Wie viele Fragen werden gestellt und mit welchem Inhalt?

EB01\_Eckhard Hauenherm



ein bisschen sehen, ob da kontinuierlich damit gearbeitet wird. Bei so einer Prozessautomatisierung ist es ja häufig so, das sehen sie in der Oberfläche gar nicht. Da hat irgendwer im Hintergrund einen Prozess über irgendein Datenposetry laufen lassen und der wird immer automatisch getriggert. Die Anwender merken da nichts von. Die machen nichts mehr, aber der Prozess läuft. Ist das jetzt gut oder schlecht? Da kann ich nur gucken, wie häufig wird der angestoßen? Kann ich mal ins Reporting gucken und gucken; läuft das, treten Fragen dazu auf oder funktioniert das. Am Anfang hat man typischerweise schon den Effekt, dass Leute auf einen zu kommen und sagen: "Das funktioniert noch nicht so richtig, das müssen wir noch optimieren" oder: „Da ist noch ein Punkt, an den wir noch mal ranmüssen.“ Sehr häufig habe ich das so, dass die das nach und nach sozusagen dann auch selber machen, weil; das ist ja die Idee bei der Einführung einer Plattform; dass die Anwender dann auch selber da hingehen und dann kann ich eigentlich nur gucken gehen und mal in die Read Folds gucken, ob irgendwo verwaiste Anwendungen liegen, die nie wieder genutzt wurden. Wo einer mal was gebastelt hat, was nie zu Ende gegangen ist. Da können wir vielleicht auch mal hingehen und gucken. Deswegen, das ist immer der Grund, warum ich so ein bisschen immer davon abrate, sofort in die Breite zu gehen, weil dann verlieren sie den Überblick. Dann wissen sie nicht, was passiert, dann lieber zielgerichtet mit einzelnen Leuten, mit kleinen Teams, fachbereichsspezifisch zu arbeiten, weil dann ist erstens die Motivation besser da und man kriegt auch einen besseren Überblick, was wird tatsächlich eingesetzt, was ist wirklich sinnvoll.

EB01\_Eckhard Hauenherm: 77 - 77 (0)

Da hilft es, also ich habe da immer ein offenes Ohr für die Leute, die können sich direkt an mich wenden und ich versuche schnell zu helfen und durch die Rolle von einem Enabler haben wir auch eine Community etabliert, was ich hier

Ein qualitativer Messfaktor für den Erfolg ist Feedback von den Nutzern. Wo treten Probleme auf und wird die Community genutzt? Der Experte sieht die unternehmensweite Nutzung für kritische Geschäftsanwendungen als

auch als Erfolgsfaktor sehen würde. Also jetzt nicht hyper offizielle Struktur da, aber dass man einfach eine Community aus den Leuten hat, die sich miteinander vernetzen, austauschen können, gegenseitig helfen gerade bei Problemen/ Herausforderungen, die spezifisch das Unternehmen betreffen.

EB02\_Rosengrün, Robin: 29 - 29 (0)

Aber, dann auch tatsächlich, dann gehts dann eher in die qualitativen Ziele, dass wirklich auch sich das weiterentwickelt, dass Anwendungen mit immer größerem Impact auch auf die Plattform gehoben werden. Also dass man auch wirklich dann irgendwann so sicher ist, dass man anfängt, geschäftskritischere Anwendungen darauf zu heben. Genau das lässt sich natürlich nur schwer, dann irgendwie messen. Plus, dass man halt merkt, dass man nicht die Schatten IT 2.0 aufgebaut hat dadurch, sondern dass die Governance funktioniert, also funktioniert schon dadurch, dass man sieht, man kann solche Zahlen, wie welche Anwendungen es gibt, aber auch, dass man dann halt irgendwann die Strukturen geschaffen hat, die man braucht. Dass man merkt es kommen nicht jeden Tag Tickets rein zu Themen, der Owner von der App hat das Unternehmen verlassen. Kannst du mich als Owner einsetzen, sondern dass man merkt, man hat Strukturen aufgebaut für sowas, dass das gar nicht mehr passiert ohne Awareness für solche Themen da ist.

EB02\_Rosengrün, Robin: 67 - 67 (0)

was ich natürlich auch machen kann, was ich mir definitiv mal anschauen würde, man kann das jetzt fassen unter Mitarbeiterbefragung, aber wie zufrieden sind eigentlich die Mitarbeiter mit diesem Tool, mit dieser Nutzung? Was würden Sie sich noch wünschen? Das macht man entweder unter strukturierten Interviews, in Umfragen, in, man kennt es so bisschen vom Baumarkt, wo man dann draufhaut und sagt: "Wie zufrieden waren Sie heute mit ihrem Einkauf?", das kann man auch machen.

EB04\_Görtl, Simone: 57 - 57 (0)

Erfolgsfaktor an. Weitere qualitative Messdaten sind: welche Art von Supporttickets werden durch den Help Desk aufgenommen und wird die Governance richtig angewendet.

EB02\_Rosengrün, Robin

Die Expertin sieht als Mittel der qualitativen Erfolgsmessung Nutzerbefragungen als sinnvoll an. Dadurch kann evaluiert werden wie die Zufriedenheit mit der Plattform ist und welche Featurs noch fehlen. Dies kann über strukturierte Interviews oder Umfragen passieren.

EB04\_Görtl, Simone

Wie zufrieden sind die Leute? Das merkt man dann danach, wenn man sich mal das Team und die Stakeholder hinterher anschaut. Ist es gut gelaufen, sind alle glücklich? Es ist halt keine wirklich messbare Größe.

EB05\_Rohrbeck, Christian: 75 - 75 (0)

Also da muss irgendein Effekt irgendwie sichtbar sein. Das muss schneller gehen. Das muss leichter gehen. Medienbrüche sind plötzlich weg. Wir haben ja ganz viele Kunden, die nach wie vor mit Papier und Klemmbrett arbeiten und Stift. Und die sagen: Herr Hesse, der Nutzer tippt das hier ein, dann fotografiert er das hier ab, dann landet das per Mail bei der Frau Schulze. Die Frau Schulze nimmt sich das, tippt das dann in die Excel-Tabelle, die Excel-Tabelle wird irgendwo abgelegt. Sie glauben gar nicht, wie oft das passiert, dass die Frau Schulze einen Zahlendreher drin hat und plötzlich gehen unsere Prozesse nicht mehr. Ja, das heißt, der Erfolg bemisst sich ja durchaus daran, wie die Effekte beim Kunden wahrnehmbar sind. Wow! Ich habe früher was an die IT gegeben. Nach 4 Wochen hatte ich die Antwort: Können wir umsetzen. Ist Ende Herbst da. Jetzt habe ich eine Idee. Ich klick mich rein in die Plattform und 15 Minuten später habe ich eine Applikation, mit der kann ich meine Daten hier draußen im Feld erfassen und wenn ich es abgeklickt habe, landet die Excel-Tabelle bei der Frau Schulze und die muss nichts mehr umändern. Also das ist dann die wesentliche Erfolgsmessung.

EB06\_Hesse, Moritz: 85 - 85 (0)

Also ich glaube, Erfolg kann man in verschiedenen Richtungen messen. Zum einen den gefühlten Erfolg, dass man also über zum Beispiel Retrospektiven herausfindet, was denken die Mitarbeiter dazu? Was ist der große Vorteil der in Führungsstrichen Neuen Welt?

EB07\_Janke, Tino: 45 - 45 (0)

Ja, also das ist der Punkt, ich habe ja, sozusagen während der Entwicklung,

Als Mittel der qualitativen Erfolgsmessung wird eine Abfrage angestrebt über die Zufriedenheit der Nutzer.

EB05\_Rohrbeck, Christian

Als Mittel der qualitativen Erfolgsmessung wird eine Abfrage von Zufriedenheitsfaktoren gesehen. Dabei sollte evaluiert werden, ob Prozesse verbessert oder beschleunigt wurden, passieren weniger Fehler durch die Applikation und ist der Mehrwert für Nutzer erkennbar.

EB06\_Hesse, Moritz

Der Experte weist darauf hin, dass über das Tool der Retrospektive eine qualitative Evaluierung von Erfolgszielen möglich ist.

EB07\_Janke, Tino

Der Experte hebt hervor als qualitative Methodik, Evaluationsmethodiken zu

habe ich ja ganz viele Veränderungen. Ich habe eine Basis, auf der ich aufsetze, dann fange ich an zu entwickeln. Lerne dazu, ändere es wieder ab, baue das immer ein. Dadurch ist die Evaluation, das ist der Vergleich des Ergebnisses mit dem angestrebten Ziel, diese Evaluation ist dann zumindest nicht auf einem Status, der mal vorher in der Historie war, einfach durchzuführen. Weil eben moving Target, das verändert sich eben auch und auf einmal habe ich eine ganz andere Anforderung, in der sechsten/siebten Woche, als ich zu Anfang hatte. Ist bei der IT-Abteilung üblicherweise nicht anders. Auch die machen ja viele, viele Dinge wieder neu. Der Vergleich, bin ich erst mal mit der Funktionalität zufrieden, die kommt dann, wie im Review im agilen Bereich, der kommt dann durch den Kundenkontakt. Das heißt: "The proof pudding is in the eating", also hier muss dann tatsächlich der Pudding gegessen werden und so ist es hier auch. Das heißt, es wird angewandt. Dadurch kommt erst mal heraus, habe ich das gut gemacht oder nicht.

EB08\_Fey, Peter: 55 - 55 (0)

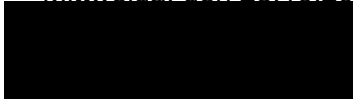
nutzen. Dabei sollte festgestellt werden, wurde das Ziel erreicht, wie verändern sich die Anwendungsfelder. Sind diese noch genauso wie ursprünglich geplant oder konnten zusätzliche Einsatzfelder aufgezeigt werden. Wie zufrieden ist die IT-Abteilung? Der Experte weist darauf hin, dass der reale Erfolg erst durch die Anwendung über einen längeren Zeitpunkt evaluiert werden kann.

EB08\_Fey, Peter

## Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.  
Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.  
Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 28.10.2022



Rocco Schenk