HS PF

Hochschule Pforzheim

Master Information Systems

Potenzialanalyse bei der Softwareentwicklung mit Low Code Plattformen in der öffentlichen Verwaltung

Eine Analyse am Beispiel der Low Code Plattform A12 im Kontext der OZG Umsetzung

Master-Thesis

Erstprüfer	Prof. Dr. Peter Weiß
Zweitprüfer	Prof. Dr. Frank Morelli
vorgelegt von	Jonas Baur
Matrikelnummer	317212
Semester	3
Abgabetermin	31.01.2023

Erklärung

Ich bestätige, dass ich die vorliegende Arbeit eigenständig verfasst habe, keine anderen Quellen oder Hilfsmittel als die angegebenen verwendet und alle übernommenen Passagen kenntlich gemacht habe.

Pforzheim, den 31.01.2023

Inhaltsverzeichnis

Abbildungsverzeichnis vi

Tabellenverzeichnis vii

Abkürzungsverzeichnis viii

1 Einleitung 1
1.1 Problemstellung 2
1.2 Ziel der Arbeit 3
1.3 Aufbau der Arbeit 3
1.4 Methodik 4

2 Grundlagen 6
2.1 Digitalisierung in der öffentlichen Verwaltung 6
2.1.1 Definition Digitalisierung 6
2.1.2 Definition und Abgrenzung der öffentlichen Verwaltung 7
2.1.3 Stand der Digitalisierung in der öffentlichen Verwaltung 8
2.1.4 Strukturelle Herausforderungen 9
2.1.5 Technische Herausforderungen 13
2.1.6 Aktuelle Lösungsansätze 14
2.2 Softwareentwicklung 23
2.2.1 Definition Softwareentwicklung 23
2.2.2 Phasenorientierte Vorgehensweise 24
2.2.3 Agile Vorgehensweise 25
2.2.4 Erkenntnisse 26
2.3 Low Code Plattform 26
2.3.1 Definition und Abgrenzung: Low Code Plattform 26
2.3.2 Architektur 27
2.3.3 Kategorien 30
2.3.4 Merkmale 32
2.3.5 Anbieter 34
2.3.6 Vorteile 38
2.3.7 Nachteile 39
2.4 Überblick der Grundlagen 40

3 Die Low Code Plattform A12 42
3.1 Einführung 42
3.2 Anwendungsgebiet 43
3.3 Modellierungskonzept 43

3.4 Modellierungstools 45
3.5 Vorteile 45
3.6 Vorgehensweise bei der Softwareentwicklung 46
3.7 Architektur 47
3.8 Komponenten 48
3.9 Betriebsmöglichkeiten 49
3.10 Zusammenfassung 49

4 Empirische Untersuchung 50
4.1 Methodenauswahl und -beschreibung: Experteninterviews 50
4.2 Expertenauswahl und Darstellung der Stichprobe 51
4.3 Vorbereitung 52
4.4 Leitfaden ausarbeiten 53
4.5 Auswertungsmethode Qualitative Inhaltsanalyse 56

5 Ergebnisse 58
5.1 Begriffserklärung und Vorteile von Low Code Plattformen 58
5.1.1 Allgemein 58
5.1.2 A12 62
5.2 Einsatz von Low Code 63
5.2.1 Voraussetzungen für den erfolgreichen Einsatz von Low Code Plattformen 63
5.2.2 Vorrausetzung für den erfolgreichen Einsatz von A12 65
5.2.3 Voraussetzungen für die Nutzung von A12 66
5.3 Expertenmeinung zu Low Code und welche Probleme es löst 66
5.3.1 Software bzw. Anwendungen, bei denen Low Code Plattformen besonders geeignet sind 66
5.3.2 Herausforderungen bei der Softwareentwicklung mit Low Code Plattformen in der öffentlichen Verwaltung 66
5.3.3 Optimierung der IT-Sicherheit durch Reduzierung der Schatten-IT 67
5.3.4 Nachteile von Low Code Plattformen in der Softwareentwicklung 69
5.3.5 Probleme, die A12 bei der Softwareentwicklung in der öffentlichen Verwaltung löst 70
5.3.6 Herausforderungen bei der Softwareentwicklung mit A12 71
5.3.7 Verbesserung der IT-Sicherheit durch Verwendung von A12 71
5.3.8 Nachteile von A12 72
5.4 Zukunftsperspektive 73
5.4.1 Möglichkeiten das Einsatzgebiet von Low Code Plattformen in der öffentlichen Verwaltung zu erweitern? 73
5.4.2 Zukünftige mögliche Anwendungsgebiete für den Einsatz von Low Code Plattformen in der öffentlichen Verwaltung 73

5.4.3 Weitere Einsatzmöglichkeiten mit A12 die bisher nicht angewandt werden ... 74
5.4.4 Wünschenswerte Verbesserungen in A12 ... 74

6 Fazit und Ausblick ... 75
6.1 Ausblick ... 79
6.2 Limitierung ... 81

Literaturverzeichnis ... 82

Anhang ... 94

Abbildungsverzeichnis

Abbildung 1: Digital Economy and Society Index - Digital Public Services (European Commission 2022, S. 4) 9
Abbildung 2: Ebenen der Low Code Plattform (Eigene Darstellung in Anlehnung an Sahay et al. 2020, S. 2) 27
Abbildung 3: Hauptkomponenten der Low Code Plattform (Sahay et al. 2020, S. 2) 29
Abbildung 4: Low-Code-Entwicklungsplattformen für professionelle Entwickler, Q2 2021 (Bratincevic und Koplowitz 2021, S. 3) 35
Abbildung 5: Softwareentwicklungsprozess mit A12 (Mehmanesh et al. 2022, S. 33) 46
Abbildung 6: A12 Architektur – Anwenderseite (Mehmanesh et al. 2022, S. 35) 47
Abbildung 7: A12 Architektur – Serverseite (Mehmanesh et al. 2022, S. 36) 48

Tabellenverzeichnis

Tabelle 1: Übersicht der Experten 52

Abkürzungsverzeichnis

API	Application Programming Interface
eID	Europäische digitale Identität
GUI	Graphical User Interface
PaaS	Platform as a Service
REST	Representational State Transfer
UI-Formen	User Interface
UX	User Experience

1 Einleitung

“A tragedy is unfolding”. Mit diesem Satz beschrieb Javier H. Valdés, Co-Direktor der New Yorker Taxi-Arbeiter-Allianz, die damalige Covid-19-Situation in New York City (Correal und Jacobs 2020). 730 Tote durch Covid-19 verzeichnete die Stadt am 14. April und sorgte bei den verantwortlichen Politikern für schnelles Handeln (City of New York 2020). Um nachvollziehen zu können, wie sich Covid-19 in New York City ausbreitet und wie den Bedürftigen effizient geholfen werden kann, richtete die Stadt innerhalb von drei Tagen das COVID-19 Engagement Portal ein, damit Informationen über Personen mit COVID-19 und weitere Personen, die mit den Erkrankten in Kontakt standen, gesammelt werden konnten (PR Newswire 2020).

Doch nicht nur in den USA ist der Wunsch nach digitalen Services vom Staat, die innerhalb kurzer Zeit den Bürgern zur Verfügung stehen, hoch. Laut einer repräsentativen Umfrage des Digitalverbands Bitkom fordern 86% der Deutschen von ihrer Stadtverwaltung, die Digitalisierung mit mehr Nachdruck zu verfolgen (bitkom 2021). Die Politik reagiert auf die Forderung der Bürger[1], indem sie das Onlinezugangsgesetz (OZG) verabschiedete. Bund, Länder und Kommunen sind dazu verpflichtet, bis Ende 2022 ihre Verwaltungsdienstleistungen über Verwaltungsportale auch digital anzubieten (Bundesministerium des Innern und für Heimat 2022). Laut Bitkom sind bis Oktober 2021 von insgesamt 575 Verwaltungsleistungen bisher 314 gegenwärtig aktiv in Bearbeitung. Davon befinden sich 115 in der Planungs- und 199 in der Umsetzungsphase. Bisher sind 73 Einzelleistungen für die Bürger verfügbar (bitkom 2021). Dabei stellt sich die Frage, warum in der öffentlichen Verwaltung, die Umsetzung von digitalen Verwaltungsdienstleistungen so langwierig ist und gleichzeitig die Stadt New York City innerhalb von drei Tagen ein COVID-19 Engagement Portal einrichten konnte. Möglich war die schnelle Umsetzung durch eine bisher unkonventionelle Art der Softwareentwicklung, bei der wenig, bis keine Programmierkenntnisse vorausgesetzt werden. Mitarbeiter der Stadt New York kannten die Anforderungen an ein solches Portal, da sie tagtäglich mit den Herausforderungen der Covid-19 Pandemie konfrontiert waren. Der Vorteil gegenüber der bisherigen Softwareentwicklung bestand darin, dass direkt mit der Umsetzung des Portals begonnen werden konnte, da die

[1] Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers verzichtet. Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

Mitarbeiter nicht auf die Expertise der Softwareentwickler angewiesen sind. Inwieweit dieses Potenzial ebenfalls für die Softwareentwicklung in der deutschen öffentlichen Verwaltung genutzt werden kann, soll in der folgenden wissenschaftlichen Arbeit analysiert werden.

1.1 Problemstellung

Bürger und Sachbearbeiter in öffentlichen Verwaltungen nutzen privat intuitiv zu bedienende digitale Anwendungen und erwarten diese Benutzerfreundlichkeit auch bei Verwaltungsdienstleistungen. Im privaten Alltag ist Amazon das Kaufhaus, Spotify der Plattenladen und Netflix die Videothek. Diese Erwartungshaltung in Bezug auf Anwenderfreundlichkeit wird ebenfalls vom Staat erwartet. Hierfür sind digitale Verwaltungsdienstleistungen notwendig, was dazu führt, dass viele Anwendungen entwickelt oder verbessert werden müssen. Darüber hinaus unterliegen die gesetzlichen Anforderungen in der öffentlichen Verwaltung einem stetigen Wandel, was eine hohe Anpassungsfähigkeit erfordert. Es reicht daher nicht nur aus, bestehende Software zu nutzen. Je nach Anwendungsgebiet sind Weiterentwicklung und Wartung der Software notwendig (Frank et al. 2021, S. 3).

Ein Grund für die Verzögerung des aktuellen Onlinezugangsgesetzes (OZG) und generellen Digitalisierungsambitionen der öffentlichen Verwaltung ist, dass die Softwareentwicklung von langen, starren und umfangreichen Entwicklungsprozessen geprägt ist. Folglich ist die Softwarebereitstellung zeit-, kosten- und ressourcenintensiv (techconsult 2021, S. 4).

Für das Jahr 2022 bestimmt der Branchenverband Bitkom 96.000 offene Stellen für IT-Fachkräfte (bitkom 2022). Dieses Problem wird sich laut der Boston Consulting Group in den kommenden Jahren weiter zuspitzen. Sie erwarten in ihrem aktuellen Report, über die Zukunft von Jobs, dass bis zum Jahr 2030 rund 1,1 Millionen Talente im Bereich IT und Mathematik fehlen werden (Strack et al. 2021, S. 8). Für die öffentliche Verwaltung wird es ohne eine grundsätzlich neue Herangehensweise in der Softwareentwicklung schwierig, die selbst gesteckten Ziele zu erreichen und die Digitalisierung zu meistern. Zusätzlich können Softwareentwickler, welche die erforderlichen Programmierkenntnisse mitbringen, nicht mit der Nachfrage aus den Fachabteilungen mithalten (Garms 2021, S. 41).

Im Zusammenhang mit den zuvor erwähnten Problemen, werden in der Literatur häufig Low Code Plattformen als potenzielle Lösung genannt, da dadurch Software schneller und einfacher entwickelt werden kann. Ob Low Code Plattformen in der öffentlichen Verwaltung tatsächlich die Lösung für

die aufgezeigten Probleme sein können, wird in der folgenden wissenschaftlichen Arbeit behandelt. Um dieser Problemstellung nachzugehen, werden Projekte, in denen die Low Code Plattform A12 eingesetzt wird, genauer betrachtet.

1.2 Ziel der Arbeit

Das Ziel der Forschung ist es das Potenzial der Softwareentwicklung mit Low Code Plattformen in der öffentlichen Verwaltung zu beleuchten. Hierfür wird eine empirische Untersuchung in Form von Experteninterviews durchgeführt, um Einblicke in die Softwareentwicklung mit Low Code Plattformen zu erhalten. Mithilfe der Interviews gilt es aufzuzeigen, welche Vorteile durch den Einsatz von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung entstehen. Daraus ergibt sich die Frage, welche Voraussetzungen hierfür erfüllt sein müssen. Eine weitere wichtige Erkenntnis, die mithilfe der Experteninterviews herausgearbeitet werden soll, ist, welche Herausforderungen es aktuell bei der Softwareentwicklung mit Low Code Plattformen in der öffentlichen Verwaltung gibt. Mithilfe der identifizierten Herausforderungen gilt es anschließend zu untersuchen, wie die Softwareentwicklung mit der Low Code Plattform A12 verbessert werden kann.

1.3 Aufbau der Arbeit

Zunächst gilt es dem Leser die Grundlagen zu den Themen Digitalisierung in der öffentlichen Verwaltung, Softwareentwicklung und Low Code Plattformen zu vermitteln. Bezüglich der Digitalisierung in der öffentlichen Verwaltung wird zum einen die aktuelle Situation aufgezeigt, um im Anschluss auf die damit einhergehenden Herausforderungen einzugehen. Ansätze, wie diese Herausforderungen zu bewältigen sind, werden anschließend betrachtet. Bei der Softwareentwicklung werden chronologisch die verschiedenen Vorgehensweisen aufgezeigt. Des Weiteren ist das Verständnis des Prinzips der Low Code Plattform entscheidend. Dabei sind verschiedene Aspekte wie eine Abgrenzung des Begriffs, Vorteile, Arten und Aufbau relevant.

Die Low Code Plattform A12 ist ein weiterer wichtiger Bestandteil der wissenschaftlichen Arbeit, da beispielhaft das Potenzial von Low Code Plattformen bei der Softwareentwicklung analysiert wird. Daher wird A12 in einem eigenen Kapitel beleuchtet und der Bezug zu Anwendungsgebiet, Modeling-Konzept und Architektur genauer betrachtet.

Im vierten Kapitel wird das methodische Vorgehen beschrieben. Dabei handelt es sich um eine empirische Untersuchung in Form von Experteninterviews, die im Anschluss mit der qualitativen Inhaltsanalyse ausgewertet wird. Bei den Experten handelt es sich um Projektbeteiligte aus verschiedenen Fachbereichen, die in Projekten arbeiten, in denen A12 als Low Code Plattform zum Einsatz kommt.

Im darauffolgenden Abschnitt werden die eingeholten Informationen aus der empirischen Untersuchung strukturiert und so ausgewertet, dass allgemeine und mgm-spezifische Erkenntnisse über den Einsatz von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung gewonnen werden können.

Abschließend werden die wichtigsten Erkenntnisse der Thesis kompakt und anschaulich als Fazit wiedergegeben und ein Ausblick für weitere, mögliche Forschungsfelder erörtert.

1.4 Methodik

Als Methodik der wissenschaftlichen Arbeit werden Experteninterviews als empirische Untersuchung verwendet, da dadurch die Möglichkeit besteht, aktuelle, interne Projektinformationen über die Softwareentwicklung mit Low Code Plattformen zu erhalten. Zum einen gilt es grundsätzlich zu überprüfen, ob die aufgestellten wissenschaftlichen Hypothesen über Low Code Plattformen von den Anwendern in der öffentlichen Verwaltung bestätigt werden. Zum anderen kann mithilfe der Befragungen das Potenzial der Softwareentwicklung mit Low Code Plattformen in der öffentlichen Verwaltung analysiert werden, um die Ziele der Arbeit zu erreichen.

Die Interviewpartner kommen aus unterschiedlichen Fachbereichen und besetzen verschiedene Positionen in diversen Projekten. Neben den Projektmanagern kommen ebenfalls Experten zu Wort, die aus dem Blickwinkel der öffentlichen Verwaltung und aus dem Blickwinkel des Auftraggebers den Softwareentwicklungsprozess begleiten. Eine weitere wichtige Betrachtungsweise ist die des Sachbearbeiters, der häufig bei großen Digitalisierungsprojekten kaum Beachtung findet. Zusätzlich kann die Befragung von Softwareentwicklern interessante Erkenntnisse darüber liefern, inwieweit das Versprechen von Low Code Plattformen in der Praxis erfüllt wird, den Programmieraufwand bei der Softwareentwicklung zu reduzieren. Weitere interessante Erkenntnisse können aus Interviews mit wissenschaftlichen Fachexperten gewonnen werden, die aus einer anderen Sicht auf die Softwareentwicklung mit Low Code Plattformen blicken. Somit können Expertisen aus unterschiedlichen Perspektiven gesammelt werden,

um die bestmöglichen Erkenntnisse zur Beantwortung der Forschungsfrage zu erhalten.

Bei den Interviews handelt es sich um keine alltägliche Situation, da sie sich von der generellen Kommunikation in Bezug auf Fremdheit, begrenzter Zeitkapazität und spezifische Rollenkonstruktion abhebt (Gläser und Laudel 2010, S. 121). Deshalb dient ein Leitfaden als Informationsquelle, um dem Interview eine klare Struktur zu geben und gleichzeitig dem Experten eine größtmögliche Entscheidungsfreiheit zu bieten. Für die Auswertung der Interviews wird die qualitative Inhaltsanalyse verwendet, da diese sich vor allem auf die Informationsgewinnung fokussiert (Bogner et al. 2014, S. 25). Sie ist in der Lage, dem ausführlichen Interviewmaterial auf systematische Art und Weise relevante Informationen zu entnehmen.

2 Grundlagen

Für die Beantwortung der Forschungsfrage ist es essenziell, dass relevante Begriffe und Themen, mit Hilfe der Literatur, genauer betrachtet werden. Hierfür setzt sich das folgende Kapitel mit der Digitalisierung in der öffentlichen Verwaltung, der Softwareentwicklung und dem Begriff Low Code Plattform auseinander.

2.1 Digitalisierung in der öffentlichen Verwaltung

Eine deutschlandweite Behördenbefragung auf allen Verwaltungsebenen kommt zu dem Ergebnis, dass die Weiterentwicklung von E-Government und Maßnahmen der Digitalisierung, die größte Herausforderung mit dem höchsten Handlungsbedarf in den nächsten fünf Jahren ist (Hertie School 2019, S. 5). Aus diesem Grund wird der aktuelle Stand und aktuelle Herausforderungen der Digitalisierung in der öffentlichen Verwaltung im folgenden Kapitel genauer betrachtet, um die Relevanz und das Potenzial der Softwareentwicklung zu ermitteln und mögliche Lösungsansätze zu entwickeln. Zusätzlich werden die aktuellen Lösungsansätze zusammengetragen und analysiert.

Im Folgenden werden die beiden Begriffe Digitalisierung und öffentliche Verwaltung genauer definiert, um ein einheitliches Verständnis dieser Konzepte innerhalb der wissenschaftlichen Arbeit zu gewährleisten. Die Definition dieser Begriffe ist relevant, da sie als Grundlage für die weitere Analyse und Diskussion innerhalb der Arbeit dienen und somit dazu beitragen, dass die Ergebnisse der Arbeit klar und verständlich sind.

2.1.1 Definition Digitalisierung

Digitalisierung ist die Überführung von analogen Daten in ein diskretes System mit, im Extremfall, nur zwei Wertzuständen (Binärsystem). Dies umfasst sowohl die Übertragung von Aufgaben vom Menschen auf den Computer als auch die Konvertierung von Daten wie Dokumenten, Bildern und Filmen in digitale Formate. Die Digitalisierung betrifft zunehmend auch die Abbildung oder Übertragung von analogen Produkten. Hierbei werden bspw. Prozesse, die früher auf Papier abgewickelt wurden, digitalisiert. Zudem werden bereits vorhandene Inhalte wie Rechnungen und Dokumente mithilfe von Formularen oder Websites elektronisch zur Verfügung gestellt (Streicher 2020, S. 2; Heuermann et al. 2018, S. 9).

Im Kontext der öffentlichen Verwaltung bedeutet dies, die Übertragung von Aufgaben, die bisher von Menschen erledigt wurden, auf den Computer und die Überführung von analog vorliegenden Daten in digitale Form. Dies kann bspw. die Digitalisierung von Prozessen sein, die früher in Papierform abgewickelt wurden, oder die Bereitstellung von bereits vorhandenen Inhalten wie Rechnungen oder Dokumente mithilfe von Formularen oder Websites in elektronischer Form.

2.1.2 Definition und Abgrenzung der öffentlichen Verwaltung

Zur öffentlichen Verwaltung gehören Bezirksregierungen und Bürgerämter, Universitäten und Gefängnisse, Ministerien und Museen, Regulierungsbehörden und Stadtwerke, die Bundesagentur für Arbeit und die Krankenkassen. Aufgrund der vielseitigen Organisations- und Handlungsformen, ist es nicht verwunderlich, dass sich der Begriff „öffentliche Verwaltung“ nicht klar definieren, sondern nur beschreiben lässt (Bogumil 2021, S. 680 f.; Vollmann 2021, S. 17). Aufgrund der erwähnten Schwierigkeit, die Verwaltung zu definieren, versucht es Vollmann mit einer negativen Begriffserklärung, bei der Verwaltung im materiellen Sinne alle staatlichen Tätigkeiten beinhaltet, die nicht Gesetzgebung, nicht Rechtsprechung und nicht Regierung sind (Vollmann 2021, S. 19). Allerdings lassen sich dabei zwei zentrale Aussagen treffen. Im Gegensatz zu anderen Organisationen, unterliegt die öffentliche Verwaltung einer demokratisch legitimierten politischen Steuerung und Kontrolle. Zusätzlich kann zwischen der unmittelbaren Verwaltung und der mittelbaren Verwaltung unterschieden werden. Zur unmittelbaren Verwaltung zählen Gebietskörperschaften der Bundes-, Landes- und Kommunalverwaltungen. Anstalten des öffentlichen Rechts wie die Bundesagentur für Arbeit, die öffentlich-rechtlichen Rundfunkanstalten, die Sozialversicherungen, die Deutsche Bundesbank und weitere kleine Organisationen gehören zur mittelbaren Verwaltung (Bogumil 2021, S. 681). Aufgrund von rechtlichen und politischen Vorgaben gestaltet, steuert und ordnet die öffentliche Verwaltung, die Entwicklung im Außenverhältnis zu Gesellschaft und Wirtschaft (Franz 2013, S. 189 f.).

Im Kontext der wissenschaftlichen Arbeit wird die öffentliche Verwaltung als Organisation betrachtet, die von politischer Steuerung und Kontrolle beeinflusst wird. Gleichzeitig fokussiert sich die wissenschaftliche Arbeit auf die unmittelbare öffentliche Verwaltung, die direkt mit der Ausführung öffentlicher Aufgaben und der Bereitstellung von Dienstleistungen für die Bevölkerung befasst ist.

2.1.3 Stand der Digitalisierung in der öffentlichen Verwaltung

Die Digitalisierung ist ein gesellschaftlich relevantes Thema, das auch in der öffentlichen Verwaltung von enormer Bedeutung ist. Digitalisierung beschreibt in Bezug auf die elektronische Datenverarbeitung, die Übertragung von Informationen aus einem analogen Format in binäre Daten (Heuermann et al. 2018, S. 9). Um das Potenzial für die Softwareentwicklung mit Low Code Plattformen in der öffentlichen Verwaltung zu analysieren, wird im Rahmen der wissenschaftlichen Arbeit eine Bestandsaufnahme der Digitalisierung in der öffentlichen Verwaltung durchgeführt, um die Relevanz von Softwareentwicklung zu ermitteln.

Das Nutzererlebnis von Dienstleistungen im privatwirtschaftlichen Bereich unterscheidet sich zunehmend von der des öffentlichen Sektors. Gerade führt Covid-19 den Bürgern vor Augen, wie groß die strukturellen Defizite sind und wie stark die Handlungs- und Zukunftsfähigkeit Deutschlands von der Modernisierung und Digitalisierung der öffentlichen Hand abhängt (Nationaler Normenkontrollrat 2021, S. 1). Die Einschätzung des Normenkontrollamts wird ebenfalls durch zahlreiche Statistiken bestätigt. Gemäß dem Index für digitale Wirtschaft und Gesellschaft (DESI*) belegt Deutschland im Bereich Digital Public Services den 19. Platz und liegt damit hinter dem EU-Durchschnitt (siehe Abbildung 1). Beim von den UN veröffentlichten E-Government Development Index wird bewertet, wie Länder Informationstechnologien nutzen, um Zugang und Integration der eigenen Bürger zu fördern (United Nations 2022b). Deutschland belegt im Jahr 2022 den 22. Platz und liegt dabei ebenfalls hinter Ländern wie Spanien oder Frankreich (United Nations 2022a).

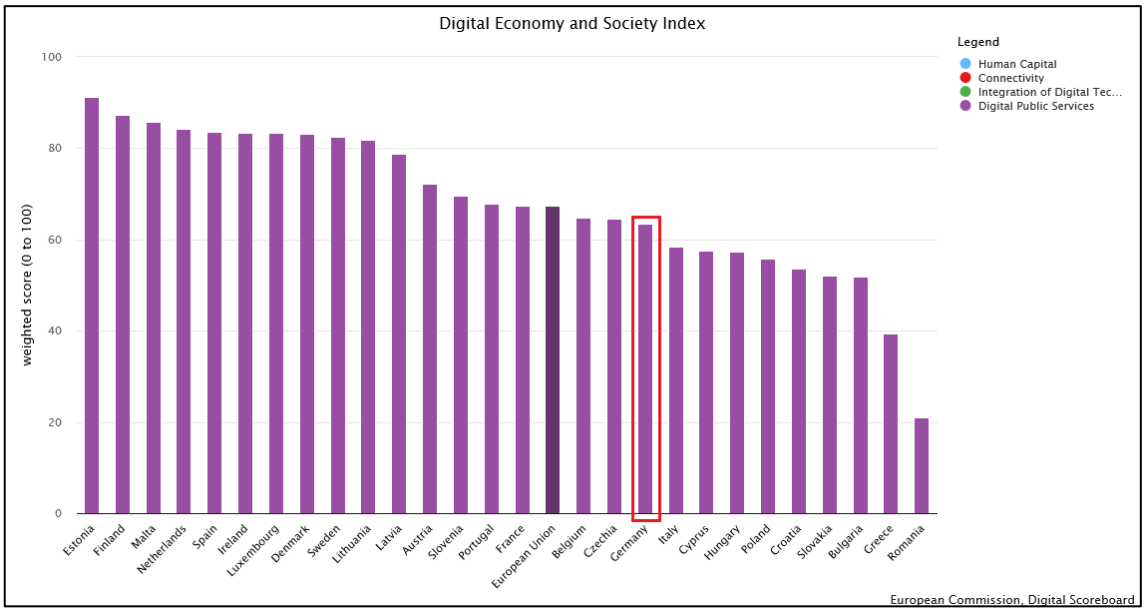

Abbildung 1: Digital Economy and Society Index - Digital Public Services (European Commission 2022, S. 4)

Jedoch lassen sich die strukturellen Defizite der öffentlichen Verwaltung nicht nur von institutioneller Seite belegen. Das Center for Digital Competitiveness zeigt in seinem Digitalreport 2021, dass lediglich vier Prozent der befragten Spitzenkräfte aus Wirtschaft, Politik und Verwaltung glauben, dass Ämter, Behörden und generell der öffentliche Dienst in Bezug auf die Nutzung digitaler Formate gut aufgestellt sind. 94 Prozent sehen einen erheblichen Nachholbedarf (European Center for Digital Competitiveness 2021, S. 12). Außerdem ziehen 88 Prozent der befragten Experten die Bilanz, dass die Covid-19-Krise bei Ämtern und Behörden Defizite aufzeigt (European Center for Digital Competitiveness 2021, S. 19).

In Anbetracht der doppelten Systemrelevanz, die eine fortschrittliche, digitale Verwaltung für Krisenbewältigung und Politikvertrauen besitzt, stellt sich die Frage, welche Möglichkeiten es gibt, die Modernisierungsfähigkeit der öffentlichen Hand substanziell zu stärken (Nationaler Normenkontrollrat 2021, S. 1). Um das Potenzial von Low Code Plattformen als Möglichkeit zur Modernisierung der öffentlichen Hand zu bestimmen, werden im Rahmen der wissenschaftlichen Arbeit die aktuellen Herausforderungen genauer betrachtet.

2.1.4 Strukturelle Herausforderungen

Im folgenden Kapitel sollen die strukturellen Herausforderungen der öffentlichen Verwaltung genauer betrachtet werden, um einen bedarfsgerechten Lösungsansatz auszuarbeiten.

Fachkräftemangel

Die fehlende Flexibilität im Besoldungssystem und die eingeschränkten Karrieremöglichkeiten, verstärken die ohnehin schon andauernde Problematik des Fachkräftemangels in der öffentlichen Verwaltung. Diese Herausforderung kann durch die Entwicklung zu einer vernetzten, digitalen, partizipativen, internationalen und kompetenten Verwaltung angegangen werden. Dabei sind flexiblere Arbeitsbedingungen, die Entwicklung von Medienkompetenzen und schnellere Kommunikation relevant. Zusätzlich muss der Umgang mit Wissen und der notwendige Wissenstransfer zwischen Verwaltungen bzw. Abteilungen entsprechend verbessert werden (Möltgen-Sicking und Winter 2018, S. 244 ff.). Um dies zu erreichen, ist eine digitalisierungsförderliche Unternehmenskultur notwendig, in der die große Mehrheit der Fach- und Führungskräfte grundsätzlich offen für digitale Innovation ist und somit der digitale Transformationsprozess aktiv unterstützt wird (Jackwerth-Rice et al. 2022, S. 104). Zusätzlich kann dem Fachkräftemangel entgegengewirkt werden, indem für die Digitalisierung bzw. Softwareentwicklung insgesamt weniger Programmierer benötigt werden und die Mitarbeiter in den Fachabteilungen in die Softwaregestaltung stärker eingebunden werden.

Altersstruktur

In den nächsten Jahren wird es in vielen Bereichen der öffentlichen Verwaltung einen umfangreichen Generationswechsel geben. Ursache hierfür sind mehrfach verfügte Sparmaßnahmen und Aufnahmestopps von Personal in den letzten 20 Jahren. Daraus resultiert eine Überalterung in vielen Bereichen des Public Sectors. Der Generationswechsel kann sich positiv auf die öffentliche Verwaltung auswirken, da junge, vollkommen anders sozialisierte und ausgebildete Personen in die Organisationen kommen werden. Sie treffen eine Generation an, die tendenziell Probleme mit der schnellen Veränderung hat, die durch die technischen Entwicklungen der letzten fünfzehn Jahre entstanden sind. Der Generationswechsel birgt die Gefahr, dass einiges an Fakten- und Prozesswissen sowie inhaltliche Erfahrungen verloren geht. Problematisch ist dies deshalb, da voraussichtlich die klassische Verfahrensabwicklung, neben der transformierten Form, noch (viele) Jahre bestehen wird. Die jungen Berufseinsteiger werden die herkömmlichen Prozesse beherrschen müssen, aber gleichzeitig intensiv an der digitalen Transformation arbeiten (Streicher 2020, S. 192 ff.). Um zu verhindern, dass Fakten- und Prozesswissen verloren geht, muss eine Möglichkeit gefunden werden, Wissen bestmöglich zusammenzutragen und zentral zu speichern. Häufig erstellen einzelne Fachabteilungen auf Grundlage von Office-Paketen IT-Produkte und

Anwendungen, die sich der Verwaltung und Dokumentation durch die IT-Abteilung entziehen (Streicher 2020, S. 202). Dadurch weiß häufig nur der Ersteller, wie das Tool aufgebaut ist bzw. bedient werden kann. Dies wird dann zu einem Problem, wenn der Ersteller nicht mehr erreichbar ist, da er bspw. in Rente geht. Außerdem müssen für eine erfolgreiche Digitalisierung der öffentlichen Verwaltung, auch all diejenigen Mitarbeiter eingebunden werden, die mit der schnellen technischen Entwicklung Probleme haben. Um dies zu erreichen, ist es erforderlich, dass Software bzw. Anwendungen auch ohne IT-Vorkenntnisse mitgestaltet werden können.

Kulturwandel

Die Digitalisierung der öffentlichen Verwaltung kann nur dann erfolgreich sein, wenn sich die Kultur der Organisation verändert. Diese Veränderungen führen dazu, dass Mitarbeiter Tätigkeiten übernehmen, die traditionell außerhalb ihrer Funktion liegen. In der Literatur wird die Idee hervorgehoben, dass durch die Digitalisierung Situationen auftreten, in denen Mitarbeiter, die nicht Teil der IT sind, die Führung bei technologieintensiven Projekten übernehmen (Vial 2019, S. 24 f.). Die Transformation der digitalen Welt entsteht nicht aus der Weiterentwicklung von Technologien, sondern auch durch Menschen in den Organisationen, die ihre Verhaltensweisen anpassen müssen. Dabei besteht die Veränderung nicht nur darin, neu gestaltete Prozesse zu implementieren. Vielmehr geht es um eine dauerhafte Veränderung der Arbeitskultur, quasi um eine kulturelle Transformation. Grundsätzlich verändert sich die Zusammenarbeit der Angestellten in der öffentlichen Verwaltung durch den Einsatz von neuen Technologien. Sie wird lebendiger, agiler und weniger leicht plan- bzw. vorhersehbar. Neue Methoden werden durch alte abgelöst und fordern eine intensive Einarbeitung nicht nur in die Vorgehensweise selbst, sondern auch in die grundlegenden technischen Möglichkeiten. Das Problem dabei ist allerdings, dass Menschen häufig an Gewohnheiten oder Traditionen und am Liebgewonnenen hängen – sie möchten nicht immer, dass sich alles verändert. Grund dafür ist, dass Veränderungen oftmals auch Unsicherheiten und Unruhe mit sich bringen. Außerdem erfordern neue Aufgaben und Anforderungen neue Talente. Dies ist häufig sehr anstrengend und dabei fragen sich viele Mitarbeiter, ob sie sich nicht schon lange genug für Dinge eingesetzt haben, die sie durch unvermeidliche Veränderungen glauben zu verlieren und wie sich die eigene tägliche Arbeit verändert (Streicher 2020, S. 196 f.). Um den dringend benötigten kulturellen Wandel der öffentlichen Verwaltung voranzutreiben, gilt es die Hürden zur Mitgestaltung so gering wie möglich zu halten. Dies geschieht, indem wenige bis keine technischen

Grundkenntnisse notwendig sind, um bei der Softwareentwicklung mitzuwirken.

Vorbehalte in der Organisation

Eine enorme Herausforderung des digitalen Wandels, wird durch die Verwaltung selbst verursacht, indem hohe Vorbehalte gegenüber der digitalen Transformation vorhanden sind. Die Skepsis gegenüber Veränderung ist gerade dort ausgeprägt, wo noch keine Transformationsstrategie oder noch kein Chief Digital Officer (CDO), als personalisiertes Zeichen der Veränderung vorhanden ist. Bei der Veränderung ist fehlende Expertise ein geringes Problem, da es sukzessiv in den Fachbereichen aufgebaut werden kann. Durch den steigenden Kompetenzstand schwindet auch die falsche Annahme, selbst nicht von der Transformation betroffen zu sein. Ein größeres Problem ist das mangelnde Vertrauen in die Expertise der Führung, denn Vertrauen bedeutet auch Zutrauen im Sinn positiver, organisatorischer und inhaltlicher Kompetenzen bei der Umsetzung. Grundsätzlich besteht ein großer Unterschied darin, eine Gruppe zu leiten, die Fachverfahren[2] abwickelt oder als Gruppenleiter die Prozesse und Verfahren zu digitalisieren. Die reine Wissensvermittlung allein reicht nicht aus, um das Vertrauen in die Veränderungen zu stärken, da die zunehmende Erkenntnis darüber, was damit einhergeht, das Vertrauen möglicherweise sogar verringern kann. Es ist ein langer Weg von einer Vertrauenskrise hin zu einem passenden Vertrauensniveau (Streicher 2020, S. 17). Eine Transformationsstrategie kann nur dann erfolgreich sein, wenn alle Mitarbeiter in die Strategie mit einbezogen werden und sich keiner zurückgelassen fühlt. Hierfür ist es wichtig, die Voraussetzungen an fachlicher Expertise zur Mitgestaltung so gering wie möglich anzusetzen, um Vertrauen in die Veränderung zu gewinnen. Wenn die Expertise der Fachexperten direkt in die Entwicklung neuer Software einfließt, kann die Gesamtzufriedenheit erhöht werden und die Akzeptanz zur Veränderung steigt.

Föderalismus

Eine Säule der föderalen Staatsstruktur in Deutschland ist das, aus der Fiskaltheorie des Föderalismus abgeleitete, Subsidiaritätsprinzip. Die Grundidee ist, dass öffentliche Aufgaben zunächst von der kleinsten Einheit, den Kreisen, kreisfreien Städte und Gemeinden, wahrgenommen werden.

[2] Fachverfahren sind Computerprogramme, die von Behörden genutzt werden, um strukturierte Prozesse, die regelmäßig anfallen, zu bearbeiten (Landesarchiv Baden-Württemberg).

Nur wenn diese die Aufgabe nicht (allein) bewältigen können, springen zuerst die Länder und dann der Bund ein. Daher ist es dem Bund, aufgrund des sogenannten Durchgriffsverbots, nicht erlaubt, direkt Aufgaben an die Kommunen und kreisfreien Städte zu übertragen (Jochimsen 2022, S. 7 f.). Daher ist es nicht verwunderlich, dass Bund, Länder und Kommunen ihre Organisation auf das (angenommene) Optimum für die Erfüllung der eigenen Aufgaben ausrichten. Die dadurch entstandenen und gewachsenen IT-Systeme sind daher nicht zueinander kompatibel. Dies führt dazu, dass gerade in den Kommunen komplizierte Fachanwendungen existieren, die sich auf spezielle kommunale Prozesse fokussieren und zusätzlich noch, je nach Behörde auf einzelne Gegebenheiten spezialisiert sind. Somit bestehen für die Erfüllung der gleichen Verwaltungsaufgaben vielfältige und lokale Standards im Arbeitsablauf (Drüke 2008, S. 12). Die spezifischen, kommunalen Fachanwendungen, welche den Föderalismus als politische Idee auszeichnet, verträgt sich nicht mit dem Bedürfnis nach Unitarisierung und Standardisierung, was für die erfolgreiche Nutzung von IT-Systeme notwendig ist (Bahrke et al. 2016, S. 91). Diese Problematik ist durch die Covid-19 Pandemie aktueller denn je, da die Probleme der spezifischen, nicht zueinander kompatiblen IT-Systeme, die eine agile und effiziente Handlungsweise verhindern verdeutlicht werden. Eine Möglichkeit dem entgegenzuwirken ist ein universelles, technisches Grundgerüst, auf der Kommunen ihre Fachanwendungen ausführen bzw. entwickeln können und somit sicherstellen, dass eine Kompatibilität gewährleistet ist.

2.1.5 Technische Herausforderungen

<u>IT-Budget</u>

Für die Digitalisierung der öffentlichen Verwaltung im Außenverhältnis zum Bürger ist eine entsprechende Ausrüstung mit Hard- und Software nötig, um die Sicherheit für IT-Infrastruktur und Datenbestand nach dem aktuellen Stand der Technik zu gewährleisten (Windoffer 2018, S. 367). Zusätzlich ist im Regelfall die klassische IT-Organisation mit ihren budgetären Rahmenbedingungen damit beschäftigt, die bestehende vielschichtige Anwendungslandschaft stabil zu betreiben und die immer anspruchsvolleren Sicherheitsanforderungen zu erfüllen. Der konkrete Wille für innovative Veränderungen ist in vielen Fällen vorhanden. Allerdings fehlt häufig die Zeit und die zusätzlichen Mittel bzw. das qualifizierte Personal (Streicher 2020, S. 201). Insgesamt lässt sich zudem ein klarer Trend erkennen, dass mehr IT-Budgets für das Tagesgeschäft und weniger für Innovationen ausgegeben werden. Seit Jahren steigen kontinuierlich die Ausgaben für Aktualisierungen, Updates und Erweiterungen. Grund hierfür sind zum einen die verkürzten

Release-Zyklen der Softwareunternehmen. Zum anderen steigen durch die Digitalisierung die Anzahl der benötigten Anwendungen und somit die notwendigen Aktualisierungen (Capgemini 2017, S. 10, 2019, S. 12). Da die kurzen Release-Zyklen und die Anzahl der benötigten Anwendungen kaum verändert werden können, muss das verbleibende Budget so effizient wie möglich eingesetzt werden. Mithilfe von modularer Softwareentwicklung kann Zeit und somit auch Kosten gespart werden, da die Entwicklung nicht jedes Mal aufs Neue von null beginnt.

Schatten IT

In der öffentlichen Verwaltung verfügt die IT über wenig konkretes Wissen bezüglich der tatsächlichen Prozesse in den betreuenden Bereichen. Daher werden Bedürfnisse zur Weiterentwicklung zu wenig oder auch gar nicht erkannt. Dies hat zur Folge, dass in vielen Fachbereichen eine Schatten-IT vorhanden ist. Hierzu gehören alle IT-Produkte und Anwendungen, die Fachabteilungen auf Grundlage von Office-Pakten eigenständig erstellt haben. Sie entziehen sich der Verwaltung und Dokumentation durch die IT-Abteilung, da kein Austausch zwischen dem Fachbereich und der IT-Abteilung bezüglich der erstellten IT-Produkte oder Anwendungen stattfindet. Zur Schatten-IT zählen unter anderem makrogesteuerte Excel-Tabellen, Textdokumente und komplexe Access-Datenbanken, die im Regelfall von Autodidakten mit hohem zeitlichem Aufwand erstellt werden. Diese nicht verwalteten Tools sind auf vielen Ebenen weit verbreitet, sie sind bestens etabliert und unverzichtbare Anwendungen, deren Ablösung schwierig und zeitaufwändig, aber im digitalen Transformationsprozess unerlässlich ist (Rentrop et al. 2011, S. 69; Streicher 2020, S. 202). Daraus ergeben sich einige Nachteile die es zu verhindern gilt: Betreuungsprobleme nach personellen Wechseln, Datenschutz- und Sicherheitsrisiken sowie extrem hohe Kosten durch Wildwuchs und Einzelbeschaffungen (Kostädt 2022, S. 296 f.). Um die Schatten-IT in der öffentlichen Verwaltung zu minimieren, gilt es administrative Hindernisse in den Fachbereichen zu beseitigen und erforderliche Technologien zur Verfügung zu stellen (Smitham 2022). Zusätzlich muss die Erstellung von IT-Produkten und Anwendungen unkompliziert möglich sein.

2.1.6 Aktuelle Lösungsansätze

Nachdem die aktuelle Situation und die Herausforderungen der Digitalisierung in der öffentlichen Verwaltung genauer betrachtet worden sind, geht es im folgenden Abschnitt um die Maßnahmen bzw.

Lösungsansätze der Bundesrepublik Deutschland zur Verbesserung der Situation.

E-Government-Gesetz

Damit die elektronische Kommunikation innerhalb der Verwaltung verbessert werden kann, ist im August 2013 das Gesetz zur Förderung der elektronischen Verwaltung (E-Government-Gesetz) verabschiedet worden. Dadurch können Bund, Länder und Kommunen einfacherer, effizientere und nutzerfreundlicherer elektronische Verwaltungsdienste anbieten.

Die Schwerpunkte des E-Government-Gesetzes sind:

- Die Verwaltung wird verpflichtet einen elektronischen Kanal zu eröffnen und zusätzlich verpflichtet sich die Bundesverwaltung einen De-Mail-Zugang zu eröffnen.
- Die Akten sollen elektronisch geführt und Grundsätze der ordnungsgemäßen Aktenführung eingehalten werden.
- Elektronische Nachweise und elektronische Bezahlung in Verwaltungsverfahren sollen leichter erbracht werden können.
- Die Publikationspflicht von elektronischen Amts- und Verkündungsblättern soll erfüllt werden.
- Analysen und Prozess müssen verpflichtend dokumentiert werden.
- Ordnung für die Verwaltung zur Bereitstellung maschinenlesbarer Datenbestände ("open data").

Neben qualifizierten elektronischen Signaturen erlaubt das Gesetz weitere Sicherheitstechnologien, um elektronische Formulare durch schriftliche Formulare zu ersetzen. Dazu gibt es zwei Technologien, die alle Funktionen schriftlich abbilden können:

1. De-Mail mit Versandmöglichkeit „absenderbestätigt“, die eine sichere Anmeldung voraussetzt.
2. Web-Anwendungen der öffentlichen Verwaltung in Kombination mit einer sicheren elektronischen Identifizierung durch die eID-Funktion des Personalausweises.

Die Rechtsverordnungsermächtigung erlaubt der Bundesregierung die rasche Anpassung an deutschland- bzw. europaweite technologische Weiterentwicklungen. Mit dem E-Government-Gesetz können weitere sichere Verfahren als Ersatz der Schriftform determiniert werden (BMI 2013, S. 2 ff.; BMI 2022b).

Digitale Verwaltung 2020

Das Regierungsprogramm „Digitale Verwaltung 2020“ vom 17.09.2014 ist ein Bestandteil der digitalen Agenda. Forderungen nach verbindlichen Standards zur flächendeckenden Digitalisierung der Verwaltung werden in dem Programm aufgegriffen. Es ist ein Anknüpfungspunkt an das bereits erwähnte E-Government-Gesetz (Bundesregierung 2017). Das Programm soll die Koordination der Umsetzung des E-Government-Gesetzes im Bundesbereich über alle Ressorts hinweg ermöglichen. Daraus ergeben sich folgende Ziele (BMI 2014b, S. 1):

- Verbindliche Standards für eine flächendeckende Digitalisierung und den Einsatz von erforderlichen innovativen, technischen und sicheren Lösungen. Es bildet ein Dach für die E-Government-Aktivitäten der Bundesverwaltung.
- Durch das Programm soll eine effiziente, elektronische Verwaltungsarbeit im Bund sowie eine medienbruch- und barrierefreie digitale Verwaltungsverfahren entstehen. Diese Arbeit zielt darauf ab, sich an den Bedürfnissen der Bürger, der Wirtschaft und nicht nur an den fachlichen Kompetenzen der Behörden zu richten.
- Die öffentliche Verwaltung soll für die Bürger elektronisch standardmäßig erreichbar sein. Durch ein Normscreening werden die verwaltungsrechtlichen Rechtsvorschriften des Bundes überprüft, um herauszufinden, welche Schriftformerfordernisse und Pflichten zum persönlichen Erscheinen potenziell wegfallen können.
- In der Verwaltung soll mittelfristig Bürokratie abgebaut und Kosteneinsparungen angestrebt werden, indem innovative technische Lösungen in der Verwaltung eingesetzt werden.

Eines der wichtigsten Ziele des Programms ist die Implementierung der elektronischen Aktenführung in der Bundesverwaltung durch den ressortübergreifenden Aktionsplan E-Akte. Dieser beinhaltet die Sammlung organisatorischer und fachlicher Aspekte sowie technischer Lösungen und Vorschläge zur Förderung einer bürgernahen und effizienteren digitalen Verwaltung. Zusätzlich sollen ergänzende, eigene IT-Kompetenzen und Prozessberatungen aufgebaut werden, damit ausgewählte Projekt in den Ressorts, besonders bei der Einführung der E-Akte, unterstützt werden können. Mit dem Projekt E-Beschaffung sollen wichtige Prozesse bei der Beschaffung des Bundes standardisiert und digitalisiert werden. Soweit möglich soll die Beschaffung der Ressorts interoperabel gestaltet werden. Dies ist aufgrund von Synergieeffekten möglich, auf Basis von bereits bestehenden standardisierten und digitalisierten Verfahren. Standardisierung

von Prozessen, Formularen und interdisziplinärer Einsatz gemeinsamer Standards sind entscheidend für eine effiziente, medienbruchfreie elektronische Abwicklung von Verwaltungsdienstleistungen. Die notwendigen Standardisierungsmaßnahmen werden von einer Bundesredaktion mit hoher Methodenkompetenz unter Einbeziehung der Ressorts koordiniert. Die von den G8-Staaten beschlossene Maßnahmen der G8-Open-Data-Charta werden in einem nationalen Aktionsplan der Bundesregierung gebündelt. Ein weiterer wichtiger Beschluss des Programms ist, entsprechend der europäischen Vorgaben für Wirtschaft und Verwaltung die Schließung der letzten Lücken bei der elektronischen Rechnung in einem durchgängig konsistenten und medienbruchfreien Prozess, von der Auftragsvergabe bis zur Bezahlung. Außerdem ist ein weiterer wichtiger Punkt, erste konkrete Schritte umzusetzen, um die relevantesten Verwaltungsdienstleistungen aller Fachbereiche und allen föderalen Ebenen für die Bürger und die Wirtschaft online anzubieten. Die Zusammenarbeit mit den Bundesländern ist ebenfalls ein wichtiger Bestandteil des Programms und wird durch den IT-Planungsrat begleitet. Dieser kümmert sich um die bereits erwähnten Inhalte im föderalen Kontext mit dem Ziel, den Bürgern und Unternehmen staatliche Dienstleistungen möglichst vollständig und lückenlos online zur Verfügung zu stellen. Für eine erfolgreiche Umsetzung ist eine Stärkung der Kompetenzen der Mitarbeiter der Bundesverwaltung in den Bereichen Datenschutz, Digitalisierung, Informationssicherheit, IT und E-Government notwendig. Dies wird ebenfalls im Regierungsprogramm vorangetrieben (BMI 2014b, S. 2 ff.; BMI 2014a, S. 3).

IT-Konsolidierung Bund

Das Projekt zur Konsolidierung der Bundes-IT läuft seit 2015 unter der Leitung des BMI. Ziel des Projekts ist es, die IT-Beschaffung zu bündeln, IT-Services aus einer Hand bereitzustellen und IT-Anwendungen in den Behörden zusammenzuführen (BWI 2019). Grund dafür sind die steigenden Herausforderungen für die IT des Bundes, da immer mehr Prozesse digital abgebildet werden, Komplexität und Datenmengen wachsen, neue Technologien integriert werden müssen und auch die Anforderungen an die Informationssicherheit und den Datenschutz zunehmen. Die steigenden Herausforderungen sind mit der aktuellen heterogenen IT-Landschaft mit geringer Standardisierung nicht zu bewältigen. Daher gibt es die IT-Konsolidierung Bund. Vereinheitlichung und Standardisierung sind diesbezüglich wesentliche Aspekte, um als Staat zukunftssichere digitale Verwaltungsleistungen anzubieten (ITZBund 2015). Zu Beginn wird die Beschaffungsbündelung priorisiert. Hierfür ist ein einheitlicher Rahmen für die Beschaffung von Informations- und Kommunikationstechnik, Hard- und

Software sowie IT-Dienstleistungen in den Bundesbehörden notwendig. Durch weniger Rechenzentren und mehr Service aus einer Hand soll bis 2022 80% des IT-Betriebs in einem Leistungsverbund gebündelt werden (BWI 2019).

OZG (Onlinezugangsgesetzt)

Das Gesetz zur Verbesserung des Onlinezugangs zu Verwaltungsleistungen verordnet Bund, Länder und Kommunen bis zum Ende des Jahres 2022 die eigenen Verwaltungsleistungen über Verwaltungsportale auch digital anzubieten und diese zu einem Verbund zu verknüpfen. Dazu gehören Leistungen wie die Beantragung von BAFöG, Geburtsurkunden oder Kindergeld (IT-Planungsrat 2020; BMI). Die Digitalisierung von Verwaltungsleistungen auf Bundes-, Länder- und kommunaler Ebene ist eine der beiden Aufgaben des OZGs. Die zweite Aufgabe ist die Vernetzung der IT-Infrastruktur, damit es für jeden Nutzer möglich ist, auf die Verwaltungsleistungen mit nur wenigen Klicks zuzugreifen. Grundsätzlich hat die Nutzerorientierung bei der OZG-Umsetzung oberste Priorität was bedeutet, dass alle Digitalisierungsprozesse nach den Bedürfnissen der Nutzer ausgerichtet sind (BMI 2017a). Insgesamt handelt es sich um 600 zu digitalisierende Verwaltungsleistungen. Diese sind im OZG-Umsetzungskatalog in Form von OZG-Leistungen in 35 Lebens- und 17 Unternehmenslagen untergliedert und 14 übergeordneten Themenfeldern wie bspw. Familie & Kind oder Unternehmensführung & -entwicklung zugewiesen. Die Ausrichtung des Umsetzungskatalogs erfolgt nicht aufgrund behördlicher Zuständigkeiten, sondern aus Sicht der Bürgerinnen und Bürger sowie Unternehmen. Der Erfolg der Digitalisierungsprogramme wird anhand der Nutzerzufriedenheit und Akzeptanz von Bürgern und Unternehmen gemessen, nicht nur daran, ob alle Verwaltungsleistungen online verfügbar sind. Hierdurch findet ein Paradigmenwechsel statt, da das OZG-Zielbild die Nutzer in den Mittelpunkt stellt (BMI 2017b).

Die OZG-Umsetzung besteht aus drei zentralen Umsetzungsaktivitäten, die in Abhängigkeit zueinanderstehen.

1. Portalverbund: Intelligente Verknüpfung der Verwaltungsportale von Bund, Länder und Kommunen.
   Ziel ist es, dass Bürger und Unternehmen, die von ihnen benötigten digitalen Verwaltungsleistungen möglichst unkompliziert bzw. komfortabel nutzen können. Durch den Portalverbund ist es für die Nutzer möglich, alle Verwaltungsleistungen in Deutschland zu finden – unabhängig davon, welches Verwaltungsportal zu Beginn aufgerufen wurde. Über den Portalverbund Online-Gateway werden die Verwaltungsportale der Länder und des Bundes technisch verbunden.

Somit ist ein Informationsaustausch untereinander möglich, damit Nutzer die Verwaltungsleistungen verbundübergreifend finden und aufrufen können (IT-Planungsrat 2020).

2. Nutzerkonten und Postfächer: Konten für Bürger und Organisationen zur Nutzung von Verwaltungsleistungen

   Für die Nutzung von digitalen Verwaltungsleistungen im Portalverbund ist eine Online-Authentifizierung notwendig. Hierfür sind Nutzerkonten für Bürger und Organisationen erforderlich, deren Erstellung der IT-Planungsrat koordiniert. Das Besondere dabei ist, dass sich die Nutzer nur einmalig registrieren und authentifizieren muss. Anschließend kann er über ein Konto alle digitalen Verwaltungsleistungen des Portalverbunds abwickeln. Zusätzlich soll das Konto so praktikabel wie möglich sein, indem persönliche Daten wie z.B. die Adresse, auf Wunsch direkt in das gewünschte Formular übertragen werden oder Bescheide in ein elektronisches Postfach zugestellt werden können (IT-Planungsrat 2020).

3. Digitalisierungsprogramme: Digitalisierung der Verwaltungsleistungen auf Bundes-, Länder- und kommunaler Ebene.

   Dieses Programm besteht aus den folgenden zwei Teilbereichen:

   a. Verwaltungsleistungen von Bund, Ländern und Kommunen digitalisieren

      Um Bürgern und Unternehmen den Zugang zu Verwaltungsdienstleistungen im Internet zu ermöglichen, müssen diese digitalisiert und für den Online-Zugang bereitgestellt werden. Diese bestehen aus insgesamt 575 OZG-Leistungsbündel von Bund, Ländern und Kommunen und sind im OZG-Umsetzungskatalog aufgeführt.

      Die Digitalisierung von Verwaltungsleistungen ist nur dann erfolgreich, wenn die Online-Services von Bürgern und Unternehmen auch tatsächlich genutzt werden. Hierfür sind Servicestandards erforderlich, die als Hilfestellung für alle Beteiligten in Bund, Ländern und Kommunen gelten, um die Nutzerfreundlichkeit sicherzustellen (BMI). Der Standard definiert 19 Qualitätsprinzipien die im Folgenden genauer beschrieben werden (BMI):

      Nutzerorientierung

      1. Nutzerbedürfnisse und -anforderungen werden erhoben, verstanden und bewertet.
      2. Digitale Angebote sollen benutzerfreundlich und verständlich gestaltet werden, um eine breite Akzeptanz bei den Nutzerinnen und Nutzern zu erreichen.
      3. Digitale Angebote sind barrierefrei konzipiert und gestaltet.
      4. Bürger sollten ihre Dokumente und Daten nur einmalig mitteilen müssen.

5. Werden Daten und Informationen der Nutzerinnen und Nutzer in digitalen Angeboten verarbeitet, werden Sicherheitsvorkehrungen zu ihrem Schutz getroffen und transparent gemacht.
6. Es werden geeignete Maßnahmen und positive Nutzererfahrungen bereitgestellt, um Bürgerinnen und Bürger sowie Unternehmen zu motivieren, den digitalen Service statt dem papierbasierten Verfahren zu nutzen.

Vorgehen
7. Solange rechtliche Vorgaben der Entwicklung einer einfachen, nutzerzentrierten und effizienten digitalen Lösung entgegenstehen, werden diese zügig adaptiert und wird dadurch die Digitaltauglichkeit des Rechts kontinuierlich erhöht.
8. Die Planung, Gestaltung, Umsetzung und laufende Verbesserung der digitalen Dienste der Verwaltung erfolgt agil, schrittweise und aus Sicht der Nutzer, um nachhaltig fortschrittlich und benutzerfreundlich zu bleiben.
9. Alle Online-Dienste können über ein zentrales Portal aufgerufen und Leistungsbeschreibungen gefunden werden, durch eine entsprechende Realisierung der digitalen Angebote (Online-Gateway Portalverbund).

Zusammenarbeit
10. Eine vertrauensvolle, Ebenen übergreifende und interdisziplinäre Zusammenarbeit, auch mit Akteuren außerhalb der Verwaltung, ist die Grundlage für nutzerorientierte und durchgängige digitale Angebote der Verwaltung unabhängig von Zuständigkeiten.
11. Zusammenschlüsse für Entwicklungsgemeinschaften werden proaktiv angestrebt.

Offenheit
12. Bei der Entwicklung und dem Betrieb digitaler Angebote sind offene Standards zu verwenden.
13. Der Quellcode für die Eigenentwicklung digitaler Angebote der Verwaltung wird als Open Source bereitgestellt, mit einer kostenfreien Lizenz, die Veränderungen erlaubt.
14. Vor der Planung und Umsetzung neuer Komponenten wird überprüft, ob bestehende digitale Angebote oder Teile davon wiederverwendet werden können.

Technischer Betrieb
15. Die Integrität, Verfügbarkeit und IT-Sicherheit der digitalen Angebote sind unverzichtbare Bestandteile bei der Entwicklung und beim ständigen Betrieb.
16. Die Interoperabilität von Komponenten wird durch gemeinsame Standards, definierte Schnittstellen und kompatible Architekturen gewährleistet.
17. Die Überprüfung des Technologie-Standards der digitalen Angebote wird regelmäßig durchgeführt und bildet die Grundlage für ihre Weiterentwicklung.

Wirkungscontrolling

18. Die Wirksamkeit der digitalen Angebote wird anonym anhand von Nutzungsintensität und Nutzerzufriedenheit erhoben.

19. Die Zufriedenheit der Nutzer mit den digitalen Angeboten wird regelmäßig überprüft, um Verbesserungen durchzuführen und zukünftige Weiterentwicklungen darauf auszurichten.

b. Digitalisierungsprogramm Föderal: Verwaltungsdigitalisierung in 14 Themenfeldern

Die Verwaltungsleistungen werden in Themenfeldern gebündelt, um überall gleich schnell und benutzerfreundlich digitalisiert werden zu können. Die Umsetzung erfolgt arbeitsteilig: Jeder Themenbereich wird gemeinsam vom Land und dem zuständigen Bundesressort geleitet. Dort arbeiten beide Parteien gemeinsam an der Entwicklung digitaler Lösungen für alle Dienstleistungen des Fachgebiets.

Diese Aufteilung der Arbeit ermöglicht eine Zeit- und Ressourcenersparnis, da nicht 16 Bundesländer unabhängig voneinander zahlreiche Online-Anwendungen entwickeln müssen. Ergebnisse aus einem Themenbereich stehen auch anderen Ländern zur Verfügung und können zur Weiterverwendung genutzt werden.

Hochpriorisierte Services werden von agilen Teams Digitalisierungslaboren entwickelt. Die Lösung ist so konzipiert, dass sie auf alle Länder übertragbar ist - nach dem sogenannten „Einer für Alle“-Prinzip (IT-Planungsrat 2020).

Rechtliche Grundlagen für die OZG-Umsetzung

§ 4 Abs. 1 OZG in Verbindung mit Art. 91c Abs. 5 GG ermöglicht es dem Bund, auf eigene Kosten IT-Komponenten wie z.B. Hard- oder Software zu beschaffen und den Ländern vorzugeben diese zu verwenden. Art. 91c Abs. 5 GG ist ein föderaler digitaler Neustart, der die Einrichtung eines verbindlichen bundesweiten Online-Portalverbunds ermöglicht. Bürger und Unternehmen erhalten die Möglichkeit, digital angebotene Verwaltungsleistungen des Bundes und der Länder über ein einheitliches Portal in Anspruch zu nehmen. Die Zusammenarbeit der Kooperationspartner ist ausdrücklich vorgesehen. Bei der Planung, der Errichtung und dem Betrieb können Bund und Länder für ihre Aufgabenerfüllung benötigten informationstechnischen Systeme zusammenwirken (BMI; Bundestag 2017).

Registermodernisierungsgesetz

Das Registermodernisierungsgesetz wurde am 05.03.2021 im Bundesrat verabschiedet und ermöglicht es Verwaltungsdaten mithilfe der sogenannten Steuer-ID sicher und datenschutzkonform der richtigen Person

zuzuordnen. Hierdurch kann die Bundesregierung das „Once-Only"-Prinzip[3] verwirklichen, da in Registern gespeicherte Angaben und Nachweise nicht immer wieder aufs Neue vorgelegt werden müssen. Somit müssen die Bürger beim Kontakt mit der Verwaltung nicht immer wieder die gleichen Daten angeben. Außerdem kann dadurch die Qualität der Registerdaten nachhaltig gesteigert werden. Mit dem Einverständnis der betroffenen Person können Angaben und Nachweise bei der Behörde, welche die benötigen Informationen bereits besitzt, angefordert werden. Das schrittweise mit der Identifikationsnummer eingeführte Datenschutzcockpit ermöglicht es dem Bürger zu überprüfen, welche ihrer Daten auf Grundalge der ID-Nummer zwischen den Behörden ausgetauscht wurden. Dadurch soll Transparenz geschaffen und Vertrauen hergestellt werden (Bundesregierung 2021; BMI 2021).

Aktueller Stand der Lösungsansätze

Bei dem Projekt IT-Konsolidierung Bund sorgen viele Irrwege und eine komplette Neuorganisation dafür, dass die Umstellung mindestens bis 2032 dauern wird. Mehrfach stellte der Bundesrechnungshof gravierende Mängel bei der Umsetzung fest und zu Beginn des Jahres 2020 musste das Projekt vollständig neu aufgestellt werden, was zur Folge hat, dass kein finalisiertes Grobkonzept vorliegt. Ein Grund für die Verzögerung ist die Personalknappheit. Beim IT-Dienstleister Informationstechnikzentrum Bund sind 15% der geplanten Stellen nicht besetzt (Rudl 2021).

Nicht viel besser sieht es bei der OZG-Umsetzung aus. Das OZG-Dashboard zeigt transparent den aktuellen Stand der OZG-Umsetzung. Stand November 2022 sind von den 575 angestrebten Verwaltungsleistungen 80 OZG-Leistungen bundesweit online (BMI 2022a). Dies zeigt klar, dass es nicht realistisch ist, alle 575 Leistungen bis Ende 2022 flächendeckend zu digitalisieren (Perplies 2022). Die Ursache hierfür ist vielfältig. Zum einen ist die Infrastruktur immer noch sehr dezentral, mit mehr als 200 unterschiedlichen Datenbanken, in denen die Daten der deutschen Behörden gespeichert werden. Es fehlen einheitliche technische Standards. Wie auch schon beim Projekt IT-Konsolidierung Bund fehlt es an Personal, um die Ziele zu erreichen. Gerade in kleinen Kommunalverwaltungen sind IT-Experten rar (Haberl 2022; INSM 2022).

[3] Das Once-Only-Prinzip zielt darauf ab, den Bürgerinnen und Bürgern das mehrfache Übermitteln von Daten und Dokumenten zu ersparen. Stattdessen sollen schrittweise Registerabfragen und zwischenbehördliche Datenaustausche genutzt werden, um Nachweisdokumente zu ersetzen (BMI).

2.2 Softwareentwicklung

In dem folgenden Kapitel wird die Softwareentwicklung mit den dazugehörigen Vorgehensweisen genauer betrachtet. Um das Potenzial von Low Code Plattformen bei der Softwareentwicklung zu analysieren, muss ein generelles Verständnis von Softwareentwicklung vorliegen.

2.2.1 Definition Softwareentwicklung

Aufgrund der Innovationsgeschwindigkeit ist es schwierig, eine verlässliche, einheitliche und systematische Definition von Software zu bestimmen. Software kann als allgemeiner Begriff für Programme verstanden werden, mit denen ein Computersystem betrieben werden kann (Balzert 2009, S. 3; Sommerville 2012, S. 31). Die Softwareentwicklung als Teilgebiet des Software-Engineering umfasst Werkzeuge und Methoden zur Neu- und Weiterentwicklung der Software. Der Entwicklungsprozess eines Softwareprodukts reicht von der Erstellung über die Fehlerbehebung und Wartung bis hin zum Austausch eines Produkts durch ein anderes Produkt. Es stellt den Prozess der Erstellung und Weiterentwicklung eines Softwaresystems dar, einschließlich aller in dieser Zeit erforderlichen Maßnahmen und Aktivitäten (Abts und Mülder 2013, S. 479; Kneuper et al. 1998, S. 18). Laut Sandhaus, Berg und Knott gibt es verschiedene IT-Vorhaben in der Softwareentwicklung, die folgendermaßen gegliedert sind (2014, S. 10):

- Neu-Entwicklung: In einem Projekt zur Neu-Entwicklung geht es um die Erstellung von Datenbanken und Programmen.
- Wartung: In der Wartungsphase ist das Ziel, die bestehenden Anwendungen und Systeme kontinuierlich weiterzuentwickeln.
- Migration: Bei der Migration werden Programme und Datenbanken von einer Umgebung in eine andere versetzt.
- Sanierung: Programme und Datenbanken werden in der gleichen Umgebung in einen besseren Zustand versetzt.
- Einführung: Methoden zur Migration von alten Daten und die Schaffung von Schnittstellen zu benachbarten Systemen.
- Integration: Ziel ist, die bestehenden Anwendungssysteme miteinander zu verknüpfen.

Im Rahmen der wissenschaftlichen Arbeit, liegt der Schwerpunkt auf der Neu-Entwicklung, Wartung und Einführung, da in der öffentlichen Verwaltung ein hoher Bedarf an neuen Anwendungen besteht.

2.2.2 Phasenorientierte Vorgehensweise

Charakteristisch für die phasenorientierte Vorgehensweise ist, dass die Softwareentwicklung in zeitliche Phasen unterteilt wird (Berg et al. 2014, S. 28). Mithilfe der Festlegung von Phasen bzw. Abschnitten, soll die Komplexität des kontinuierlichen Entscheidungsprozesses vermindert werden. Daher werden die Abschnitte nach Zeitpunkten unterteilt, in denen grundlegende Entscheidungen getroffen werden müssen (Hansen und Neumann 2005, S. 264).

Wasserfallmodell

Bei den Phasen des Wasserfallmodells, handelt es sich um die einzelnen „Arbeitsgänge“ der Softwareentwicklung. Am Ende jedes Arbeitsgangs werden die Ergebnisse in der nachfolgenden Phase weiterverarbeitet. An dieser Stelle fungieren die Ergebnisse als Meilensteine, die den derzeitigen Projektfortschritt prüfen. Eine Phase kann erst begonnen werden, wenn die vorherige abgeschlossen ist (Bunse 2008, S. 5). Das Wasserfallmodell lässt sich in folgende Phasen unterteilen: Analyse, Systemspezifikation, Grobentwurf, Feinentwurf, Komponententests, Integration, Installation und Betrieb (Berg et al. 2014, S. 30 f.; Scharch 2016, S. 18 f.)

Einsatz: Das Wasserfallmodell ist bei leicht verständlichen Softwareanforderungen besonders gut geeignet. Dies ist dann der Fall, wenn die Anforderungen beschreibbar und stabil sind. Wenn sich die Softwareanforderungen häufig ändern, kann es im schlimmsten Fall dazu führen, dass die Analysephase nicht beendet wird (Bunse 2008, S. 7).

V-Modell

Das V-Modell eignet sich für größere Entwicklungsprojekte und ist nach wie vor der Standard für die Entwicklung von IT-Systemen des Bundes. In einem umfangreichen Regelwerk wird die genaue Vorgehensweise bei der Systementwicklung beschrieben (Berg et al. 2014, S. 31; Scharch 2016, S. 29). Es gilt aufgrund der besonderen Berücksichtigung der Qualitätssicherung, mithilfe von Maßnahmen zur Validierung und Verifikation, als Weiterentwicklung des Wasserfallmodells (Abts und Mülder 2013, S. 429; Balzert 2009, S. 101).

Alle Aktivitäten, die mit der Softwareentwicklung in Verbindung stehen, befinden sich im linken Bereich des „V“s und ähneln den Phasen des Wasserfallmodells. Die bereits erwähnte Validierung und Verifikation findet im rechten Bereich des „V“s statt und wird durch Integrations- und Qualitätssicherungsaktivitäten gebildet. Bei jeder Aktivität wird geprüft, ob das System vom Auftragnehmer richtig entwickelt wird und ob das richtige

System in Auftrag gegeben wurde (Scharch 2016, S. 22; Henrich 2002, S. 57). Das besondere bei dem weiterentwickelten V-Modell-XT sind die Vorgehensbausteine, die eine Modularisierung der Abläufe ermöglichen und somit einen flexiblen Aufbau je nach Projekttyp erlauben (Friedrich 2009, S. 3).

2.2.3 Agile Vorgehensweise

Extreme Programming

Die Grundannahme für XP besteht darin, dass Software kein statisches Produkt ist, sondern durch neue Anforderungen, gesetzliche Regelungen, Designänderungen u.v.m. getrieben und verändert wird. Mithilfe eines erstellten Prototypen wird kontinuierliches Kundenfeedback eingeholt, wodurch keine ausführliche theoretische Analysephase notwendig ist (Goll 2011, S. 116; Beck und Andres 2005, S. 11). Der Schwerpunkt liegt auf der effizienten Entwicklung von qualitativ hochwertiger Software unter Berücksichtigung des Kosten- und Zeitbudgets (Rumpe 2012, S. 14; Ludewig und Lichter 2007, S. 222). Der Ablauf bei XP kann in zwei Phasen unterteilt werden. In der Planungsphase geht es darum, die Anforderungen zu verstehen und zu dokumentierten. Die iterative Realisierungsphase dient der Systementwicklung. Beide Phasen beinhalten weitere Aktivitäten zur Erstellung von Teilergebnissen (Bunse 2008, S. 116).

Einsatzbereich: XP ist gerade dann sinnvoll, wenn bei Projekten die Anforderungen zu Beginn noch nicht vollkommen geklärt sind oder sich häufig ändern. Projekte mit einer kurzen Implementierungsphase sind hierbei besonders geeignet. Außerdem sollten nicht mehr als 12 Mitarbeiter im Projektteam arbeiten (Bunse 2008, S. 116).

Scrum

Scrum ist ein Managementframework zur Softwareentwicklung, das aus wenigen klaren Regeln besteht und den Mensch in den Mittelpunkt der Softwareentwicklung stellt (Scharch 2016, S. 41; Pichler 2013, S. 1). Der Grundgedanke resultiert aus der Analyse, dass kleine, eigenständige Teams, deren Mitglieder alle Fähigkeiten beherrschen, die für die Softwareentwicklung notwendig sind, effizienter arbeiten als heterogene Teams mit Overheads in der Matrixdarstellung (Goll 2011, S. 118 f.). Zu Beginn wird die Planung nur grob vorgenommen und anschließend verfeinert. Dadurch entwickelt sich die Architektur und das Produkt nachhaltig, sodass es möglich ist, schnell auf Veränderungen zu reagieren, indem alle Auswirkungen berücksichtigt werden. (Wolf 2012, S. 30 f.). Im Wesentlichen besteht der Entwicklungszyklus in Scrum aus der Pre-Sprint-Phase, der Sprint-Phase und der Post-Sprint-Phase. Für Entwicklungsprojekte, bei

denen die Anforderungen kaum bekannt sind, eignet sich Scrum besonders gut, wenn der Kunde noch kein genaue Auskunft darüber gibt, was er möchte (Scharch 2016, S. 43 f.).

2.2.4 Erkenntnisse

Betrachtet man die Modelle der phasenorientierten- und agilen Vorgehensweisen der Softwareentwicklung im Kontext der öffentlichen Verwaltung, dann zeigt sich, dass es kein ideales Modell gibt, um die Herausforderungen der Digitalisierung zu bewältigen. Es gilt die jeweiligen Vorgehensweisen zu kombinieren, um die Vorteile bestmöglich zu nutzen. Dabei muss es das Ziel sein, ein Grundgerüst zu definieren bzw. Rahmenbedingungen festzulegen, die dabei helfen, effizient Software zu entwickeln und anzupassen.

2.3 Low Code Plattform

Damit das Potenzial bei der Softwareentwicklung mit Low Code Plattformen analysiert werden kann, ist ein grundsätzliches Verständnis von Low Code Plattformen essenziell. Daher werden in diesem Abschnitt alle relevanten Informationen bezüglich Low Code Plattformen zusammengetragen. Dabei liegt der Fokus darauf, ein allgemeingültiges Verständnis für den Begriff zu schaffen sowie die Vorteile, Nachteile, Merkmale und Anbieter herauszuarbeiten.

2.3.1 Definition und Abgrenzung: Low Code Plattform

Der Begriff „Low Code Plattform“ beschreibt eine Art von Anwendungsentwicklungsplattform, die eine schnelle und effektive Erstellung von Anwendungssystemen mit möglichst wenig manuell geschriebenem Programmiercode ermöglicht. Der Fokus liegt hauptsächlich auf der Entwicklung von Datenbanken, Geschäftsprozessen und Webanwendungen. Außerdem sollen Nicht-Programmierer („citizen developers“) mithilfe von Low Code Plattformen Anwendungssysteme entwickeln bzw. anpassen können, die ihren persönlichen Bedürfnissen entsprechen (Bock und Frank 2021, S. 1; Richardson und Rymer 2014, S. 2; Waszkowski 2019, S. 376). Ermöglicht wird dies durch die Verwendung grafischer Benutzeroberflächen und visueller Elemente. Low Code Plattformen bieten zudem die Möglichkeit der Wiederverwendbarkeit, da es nicht notwendig ist, manuell Elemente zu programmieren, die bereits über die Schnittstellen der Plattform zur Verfügung gestellt werden (Caldeira und Cardoso 2020, S. 8 f.; Rosa-Bilbao et al. 2022, S. 3). Die Technik von Low Code

Plattformen ist von der Ideologie der Programmiersprache der vierten Generation (4GL) und dem Konzept der schnellen Anwendungsentwicklung (RAD) abgeleitet. Mit der aktuellen Generation von Programmiersprachen ist es ohne tiefgehende Programmierkenntnisse möglich, Softwareanwendungen durch eine grafische Benutzeroberfläche zu entwickeln, bei der der Code automatisch im Hintergrund generiert wird. (Waszkowski 2019, S. 376; Kahanwal 2013, S. 10). Im Zusammenhang mit Low Code Plattformen ist häufig ebenfalls von No Code Plattformen die Rede. Dabei handelt es sich um Plattformen, bei der für die Entwicklung einer Anwendung praktisch keine Programmierkenntnisse erforderlich sind. Somit wird der Low Code Ansatz weiterverfolgt. Die Anwendungsentwicklung ist ausschließlich per Drag-and-Drop möglich. Dies hat zur Folge, dass im Gegensatz zu Low Code Plattformen nichts mit selbst geschriebenem Code angepasst werden kann und somit kaum Spielraum für Anpassungsmöglichkeiten besteht (Kukat und Human 2020; Augsten 2019).

2.3.2 Architektur

Ebenen einer Low Code Plattform

Aus architektonischer Sicht bestehen Low Code Plattformen aus vier Hauptebenen (siehe Abbildung 2), die im Folgenden genauer betrachtet werden (Sahay et al. 2020, S. 2):

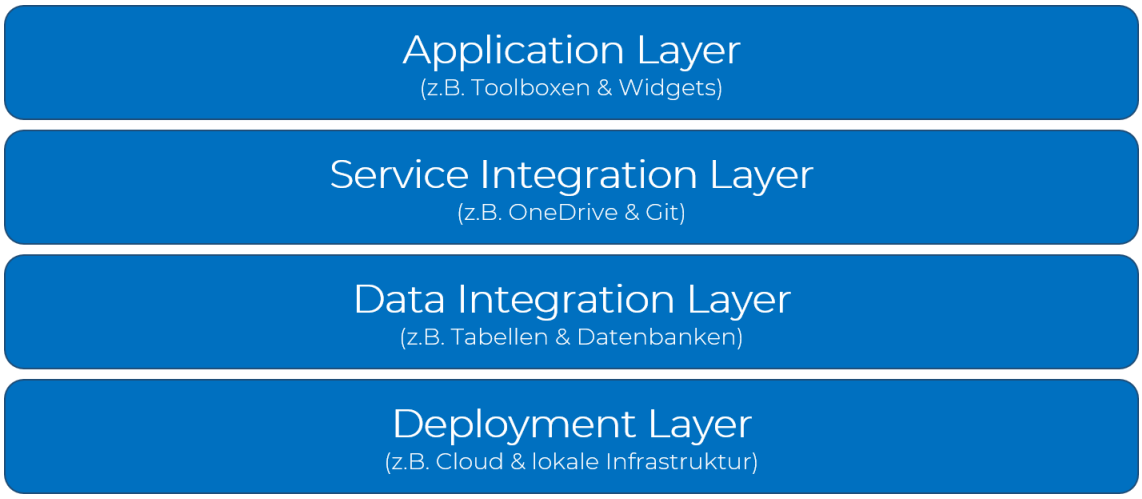

Abbildung 2: Ebenen der Low Code Plattform (Eigene Darstellung in Anlehnung an Sahay et al. 2020, S. 2)

Application Layer: Die oberste Ebene besteht aus der grafischen Oberfläche, mit der die Citizen Developer direkt interagieren, um ihre Anwendungen zu spezifizieren. Darin enthalten sind Toolboxen und Widgets, welche für den Aufbau der Benutzeroberflächen der spezifischen Anwendungen benötigt werden. In diesem Layer werden außerdem Authentifizierungs- und

Autorisierungsmechanismen definiert, die auf die spezifischen Artefakte angewendet werden können. Mithilfe der in dieser Ebene zur Verfügung stehenden Modellierungsmöglichkeiten kann der Citizen Developer die Eigenschaften der zu entwickelnden Anwendungen spezifizieren. So können diese Folgendes bestimmen:

- Wie Daten aus externen Datenquellen (Tabellen, Kalender, Sensoren und Dateien aus Cloud-Services) abgerufen werden.
- Wie Daten unter Verwendung von Plattformfunktionen oder externen Diensten verändert werden.
- Wie Daten nach definierten Regeln gesammelt und analysiert werden können.

Service Integration Layer: Diese Ebene integriert verschiedene Services, indem über entsprechende APIs und Authentifizierungsmechanismen eine Verbindung zu verschiedenen Diensten hergestellt wird. Eine spezielle Datenintegrationsschicht ermöglicht den Betrieb und die einheitliche Bearbeitung von Daten, auch wenn heterogene Quellen verwendet werden.

Data Integration Layer: Die Datenintegrationsschicht befasst sich mit der Einbindung von Daten aus unterschiedlichen Datenquellen.

Deployment Layer: Je nach Art der Plattform kann die entwickelte Anwendung auf fest zugeordneten Cloud-Infrastrukturen oder in lokalen Umgebungen bereitgestellt werden. Die Containerisierung und Orchestrierung von Anwendungen werden in dieser Ebene, zusammen mit weiteren fortlaufenden Integrations- und Bereitstellungsmöglichkeiten, die mit dem Service Integration Layer kooperieren, gehandhabt.

Hauptkomponenten einer Low Code Plattform

Durch Erweiterung der in Abbildung 2 gezeigten Ebenen-Architektur werden die einzelnen Komponenten, aus denen eine Low-Code-Entwicklungsplattform besteht, in Abbildung 3 dargestellt und können in drei Ebenen unterteilt werden. Die erste Ebene umfasst den Application modeler, die zweite Ebene bezieht sich auf die Serverseite und ihre verschiedenen Funktionalitäten und die dritte Ebene beinhaltet die externen Dienste, die in die Plattform integriert sind.

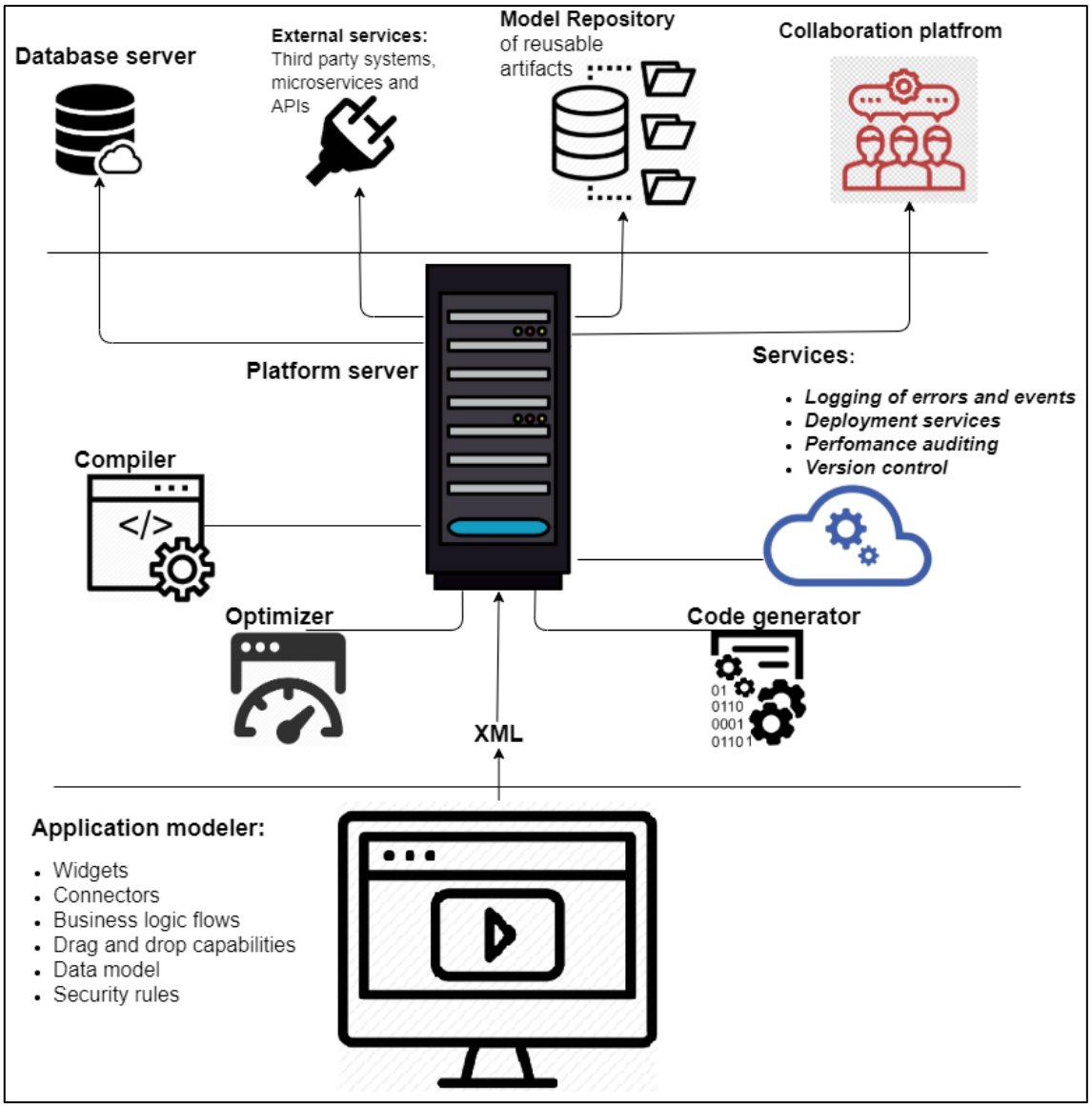

Abbildung 3: Hauptkomponenten der Low Code Plattform (Sahay et al. 2020, S. 2)

Die Pfeile in Abbildung 3 stellen mögliche Interaktionen dar, die zwischen den Einheiten der verschiedenen Ebenen auftreten können. Die Linien in der mittleren Ebene stellen die Hauptkomponenten dar, welche die Infrastruktur der Plattform bilden. Der Citizen Developer kann in der ersten Ebene der Low Code Plattform den Application modeler nutzen, der die Spezifikation von Anwendungen durch bereitgestellte Modellierungskonstrukte und Abstraktionen ermöglicht. Dazu gehört die Erstellung von Code (grafisch oder textuell), das Debugging, die Modellierung von Daten, die Codeüberprüfung, das Testen, die Versionierung, die Ereignisverwaltung und vieles mehr. Der Application modeler sollte einfach zu bedienen sein, eine Vorschau auf die entwickelte Software bieten und auf der Client-Seite angesiedelt sein. Sobald das Anwendungsmodell fertiggestellt ist, kann es zur weiteren Analyse und Bearbeitung an das Backend der Plattform gesendet werden, wo es zur

Generierung einer vollwertigen Anwendung verwendet wird, die getestet und in der Cloud bereitgestellt werden kann (Sahay et al. 2020, S. 2; Da Cruz et al. 2021, S. 7 f.).

Der Application modeler ermöglicht es ebenfalls, das System vor dem Deployment lokal auszuführen. Zu diesem Zweck nimmt die mittlere Ebene, wie in Abbildung 3 dargestellt, das vom Citizen Developer erhaltene Anwendungsmodell aus dem Application modeler und führt Modellverwaltungsoperationen durch. Die Umsetzung besteht in der Konvertierung des Quellcodes von einer Sprache in eine andere. Diese Einheit sollte Skripte zur Generierung und Interaktion mit einer Datenbank enthalten und ist auch für die Korrektheit und Optimierung des Codes verantwortlich. Dabei werden auch die beteiligten Dienste wie Datenbanksysteme, Mikrodienste, API-Connectors, Modell-Repositories mit wiederverwendbaren Artefakten berücksichtigt. Bei den Datenbankservern kann es sich sowohl um SQL als auch um NoSQL handeln. In jedem Fall sind die Anwendungsnutzer und Anwendungsentwickler nicht an der Art der verwendeten Datenbank oder an den Mechanismen interessiert, welche die Datenintegrität oder die Optimierung von Abfragen gewährleisten. Allgemeiner ausgedrückt: Der Entwickler kümmert sich nicht um die Details der Low-Level-Architektur der entwickelten Anwendung. Alle benötigten Microservices werden im Backend erstellt, orchestriert und verwaltet, ohne dass der Citizen Developer eingreifen muss. Obwohl dem Entwickler eine Umgebung zur Verfügung gestellt wird, in der er mit externen APIs interagieren kann, gibt es spezielle Konnektoren, die für die Nutzung dieser APIs im Backend zuständig sind. Auf diese Weise werden die Entwickler von der Verantwortung für die manuelle Verwaltung technischer Aspekte wie Authentifizierung, Lastausgleich, Konsistenz der Geschäftslogik, Datenintegrität und Sicherheit entlastet. Low-Code-Entwicklungsplattformen können Entwicklern auch Repositories zur Verfügung stellen, in denen wiederverwendbare Modellierungsartefakte gespeichert werden können, indem sie die Versionskontrolle übernehmen. Zur Unterstützung kollaborativer Entwicklungsaktivitäten bieten Low Code Plattformen Funktionen zur Unterstützung von Entwicklungsmethoden wie Agile, Kanban und Scrum. Somit können Citizen Developer den Anwendungsentwicklungsprozess leicht visualisieren, Aufgaben und Sprints definieren und Änderungen vornehmen, sobald der Kunde sie benötigt (Sahay et al. 2020, S. 2; Da Cruz et al. 2021, S. 7 f.).

2.3.3 Kategorien

Wenn man sich mit dem Thema Low Code Plattform beschäftigt, wird schnell klar, dass sich die im Low Code Sektor verfügbaren Plattformen in mehreren

Dimensionen stark unterscheiden. Obwohl bei den meisten Plattformen bestimmte technische Merkmale vorhanden sind, gibt es Unterschiede in Bezug auf Umfang, Hauptzweck, Bandbreite an involvierten Technologien, Möglichkeiten der Anwendbarkeit und Gestaltungsmöglichkeiten (Bock und Frank 2021, S. 735).

Frank et al. haben im Rahmen des ICB-Research-Reports 30 Plattformen untersucht und daraus vier prototypische Kategorien abgeleitet (2021, S. 29). Diese lauten:

- Einfache Datenverwaltungsplattform.
- Workflow Management System.
- GUI- und datenzentrierte integrierte Entwicklungsumgebung.
- Mehrzweckplattformen für die Konfiguration, Integration und Entwicklung von Geschäftsanwendungen.

Die vier prototypischen Kategorien dienen als erster Schritt zur Strukturierung des undurchsichtigen Begriffs "Low-Code" und den damit verbundenen Produkten. Die vier Kategorien werden im Folgenden genauer betrachtet.

Einfache Datenverwaltungsplattformen ähneln Datenverwaltungssystemen mit einem dazugehörigen GUI-Entwurfsmodul. Innerhalb dieser Low Code Plattformen dienen die Anwendungen hauptsächlich zur Ansicht und Bearbeitung eines begrenzten Bereichs von Daten, die in GUIs strukturiert sind und nach den Bedürfnissen der Citizen Developer angepasst werden. Eine Steigerung der Produktivität soll durch eine benutzerfreundliche Verwaltung von Datensätzen erreicht werden. Weitere Funktionen für das Workflow-Management oder funktionale Spezifikationen sind teilweise angedacht, stehen aber nicht im Vordergrund.

Workflow Management Systeme sind spezialisiert auf die visuelle Gestaltung von Workflows, mit zusätzlicher Unterstützung der Workflow-Ausführung und Anbindung von Drittanbietern. Im Vergleich zu einfachen Datenverwaltungsplattformen liegt der Fokus auf der Realisierung dynamischer Aspekte einer Anwendung. Quellcode-Editoren sind bei dieser Art von Low Code Plattform nicht überall verfügbar. Dies ist aber generell kein notwendiges Merkmal für die Nutzung der Plattform.

GUI- und datenzentrierte integrierte Entwicklungsumgebungen ermöglichen dem Anwender entweder umfangreiche Möglichkeiten zum Schreiben und Anpassen von Quellcode innerhalb der Plattform oder die spezielle Unterstützung zur Integration von Quellcodedateien aus externen Quellen. Eine effiziente und intensive Nutzung solcher Plattformen setzt

daher gewisse Grundkenntnisse in Bezug auf Programmiersprachen voraus. Typischerweise haben Plattformen, die dieser Kategorie zugeordnet werden, keinen speziellen Fokus auf Daten- oder Workflow-Management.

Mehrzweckplattformen für die Konfiguration, Integration und Entwicklung von Geschäftsanwendungen fassen die bisher erwähnten Elemente aus den vorherigen Kategorien zusammen, wobei der Schwerpunkt auf der Integration und Entwicklung verschiedenster Anwendungsartefakte[4] liegt. Wie bei Workflow Management Systemen liegt der Fokus darauf, mehrere interne und externe Entwicklungsartefakte zu integrieren (Frank et al. 2021, S. 29 f.).

Letztendlich kann gesagt werden, dass Plattformen, die unter der Bezeichnung „Low Code“ angeboten werden, überaus heterogen sind und keine definierte Klasse von technologischen Umgebungen darstellt (Bock und Frank 2021, S. 735).

2.3.4 Merkmale

In der Literatur werden in mehreren Marktanalysen verschiedene Low Code Plattformen betrachtet, um Merkmale zu identifizieren. Diese werden nach ihrer Häufigkeit des Auftretens dargestellt. Hierbei wird zwischen gemeinsamen, gelegentlichen und seltenen Merkmalen unterschieden (Adrian et al. 2020, S. 175 f.; Bock und Frank 2021, S. 735 ff.; Da Cruz et al. 2021, S. 6 f.)

Gemeinsame Merkmale

- Konzeptionelles Modellierungswerkzeug, das entweder eine klassische Datenmodellierungssprache oder eine vereinfachte proprietäre[5] Sprache implementiert.
- Zugriffs auf externe Datenquellen über verschiedene Anwendungsprogrammierschnittstellen (APIs). Alle Plattformen ermöglichen bspw. die Verwendung von Standard-APIs wie JDBC sowie verschiedene Konnektoren zu anderen Dateitypen und Systemen.
- GUI-Designer zur Entwicklung von grafischen Benutzeroberflächen und zur Integration mit anderen Implementierungsartefakten. Die GUI-Designer bringen Paletten mit vordefinierten Widgets mit, obwohl

[4] Ein Artefakt ist im allgemeinen Ansatz ein Gegenstand, der vom Menschen erzeugt wird. Im Kontext von Unternehmen repräsentiert ein Artefakt ein Ergebnis aus einem Arbeitsprozess (t2informatik).

[5] Properietär: (in Bezug auf ein Betriebssystem, auf Software u. Ä.) nur auf einem herstellereigenen Computermodell einsetzbar; herstellergebunden (Duden 2022).

ihr Umfang stark variiert. Mit einer Drag-&-Drop-Funktion können Steuerelemente wie Eingabefelder, Navigations-Buttons oder Dropdown-Felder einfach in die jeweiligen Bildschirmansichten eingefügt werden.

- Grundlegende funktionale Spezifikationen in Form von einfachen Ausdrucksprachen für Entscheidungsregeln und dialogbasierte Möglichkeiten zur Spezifikation von Programmablaufbedingungen.
- Code-Optimierung optimiert den generierten Code, da sonst die Software schwer zu skalieren ist. Im Idealfall ist der Quellcode für die Entwickler leicht zu lesen, da diese ihn überprüfen müssen.
- Bibliothek von generischen Standardoperationen, wie z.B. mathematische Funktionen.
- Deployment für Anwendungen macht Software für den Endnutzer verfügbar und kann je nach Zielanwender variieren.
- Debugging-Funktionen zur Ermittlung und Beseitigung von Fehlern durch die Softwareentwickler. Zusätzlich wird die Ausführung von Software in einer vollständig kontrollierten Umgebung ermöglicht.
- Aufruf und die Integration externer Funktionen über APIs.
- Verwaltung von Rollen- und Benutzerrechten.
- Versionskontrolle ermöglicht es Entwickler zu einer früheren Version des Codes zurückzukehren. Dies ist dann notwendig, wenn visuelle Codeblöcke geändert oder entfernt werden und dadurch Funktionen verloren gehen.

Gelegentliche Merkmale

- Workflow-Modellierungskomponente und Workflow-Engine sind bei größeren und komplexeren Low Code Plattformen im architektonischen Kern enthalten. Außerdem gibt es Low Code Plattformen, die in der Realität klassische Workflow-Management-Systeme sind. Für die Modellierung verwenden manche Plattformen entweder eine konzeptionelle Modellierungssprache wie Business Process Model Notation (BPMN) oder eine proprietäre Darstellungsstruktur. Bei anderen Systemen ist die Perspektive allgemeiner und technischer, indem hauptsächlich Komponenten verfügbar sind, mit denen Interaktionen und Übergange zwischen UI-Formen bestimmt werden können. Einige Plattformen stützen sich zu diesem Zweck auf proprietäre Komponenten zur Prozessmodellierung, andere auf einfache Menüs und Skripte zum Auffangen von Ereignissen.
- Klassische oder erweiterte Programmierkomponenten, bei denen prozedurale Spezifikationen mit traditionellen Programmiersprachen

und zugehörigen Technologien erstellt werden können. Am häufigsten wird hierfür Java oder JavaScript verwendet. Auf einer mehr oder weniger versteckten Ebene der Architektur gewähren fast alle Low-Code-Plattformen den Zugriff auf traditionellen Programmiercode. Bei den untersuchten Plattformen der großen Anbieter gibt es eine Vielzahl von konfigurierbaren, bausteinartigen Anwendungseinheiten. Diese Einheiten bieten z.B., in begrenztem Umfang vorimplementierte Funktionalitäten für die Bereiche Business Intelligence (BI), Künstliche Intelligenz (KI) und Robotic Process Automation.

- Cloud-Service für das Ablegen von Datenquellen wie bspw. Tabellen, damit die Nutzer der Plattform ortsungebunden dieselbe Datenbasis verwenden können.

Seltene Merkmale

- Domänenspezifische Referenzimplementierungsartefakte in Form einer Bibliothek von domänenspezifischen Referenzdatenmodellen, die gängige Geschäfts- und Kommunikationskonzepte wie "Kunde", "Adresse" und "E-Mail" abdecken.
- Katalog mit wiederverwendbaren Funktionen oder Beispiele für vordefinierte Prozesse, die jedoch meist sehr allgemein gehalten oder in Umfang und Tiefe sehr begrenzt sind.

2.3.5 Anbieter

Es gibt ein sehr großes Angebot an Low Code Plattformen, die sich auf die verschiedensten Bereiche spezialisieren – Tendenz steigend. Daher ist es gar nicht so einfach, einen Marktüberblick mit allen aktuell relevanten Anbietern von Low Code Plattformen zu erhalten. Das Forschungsinstitut Forrester beurteilt Anbieter von Low Code Plattformen für professionelle Entwickler anhand von 27 Kriterien. Dabei werden die 14 führenden Anbieter analysiert und bewertet (Bratincevic und Koplowitz 2021, S. 1).

Daraus resultiert eine Analyse der folgenden 14 Anbieter: AgilePoint, Appian, GeneXus, HCL Software, Mendix, Microsoft, Oracle, OutSystems, Pegasystems, Salesforce, ServiceNow, Thinkwise, Unqork und WaveMaker (siehe Abbildung 4).

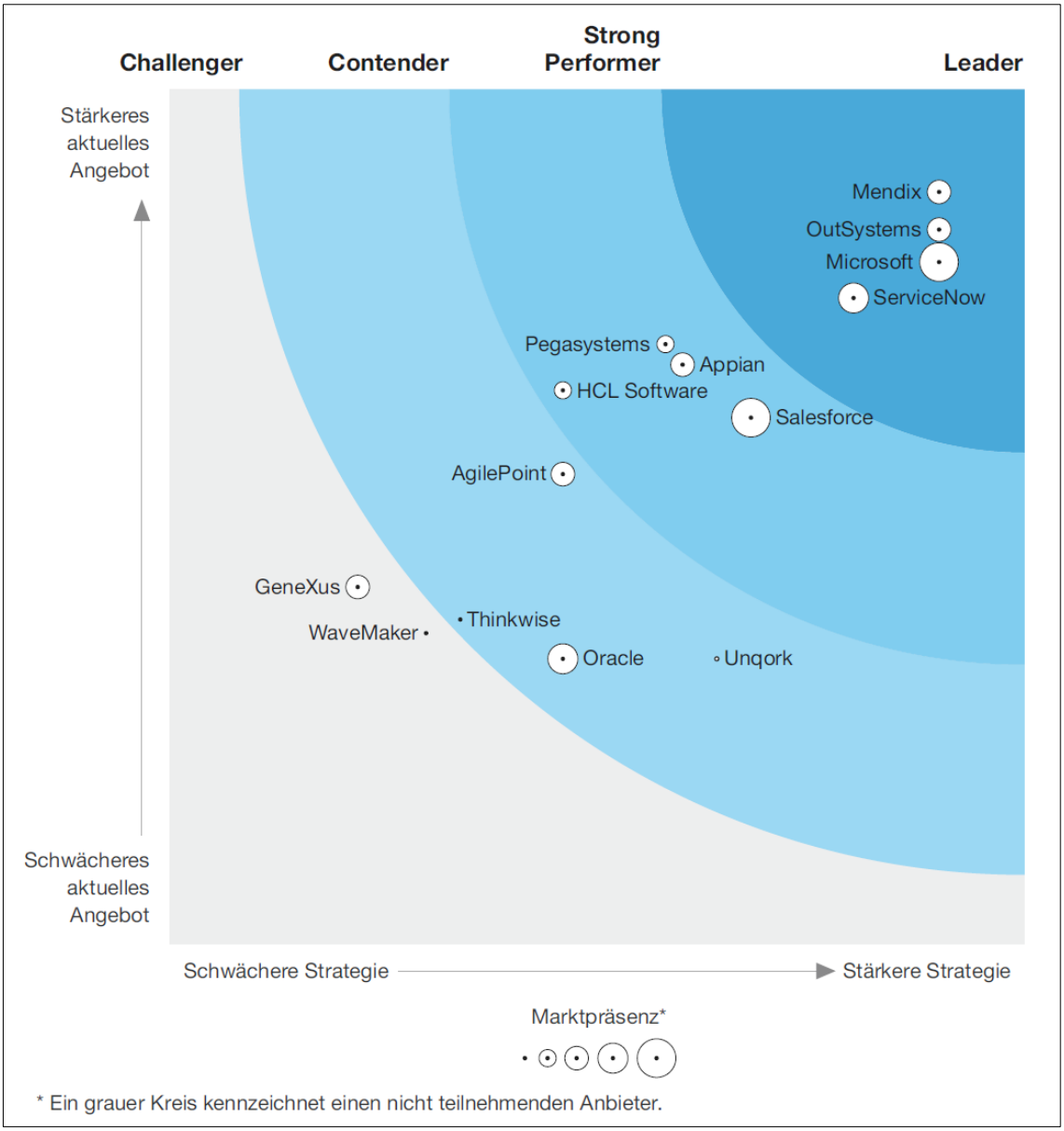

Abbildung 4: Low-Code-Entwicklungsplattformen für professionelle Entwickler, Q2 2021 (Bratincevic und Koplowitz 2021, S. 3)

Aufgrund des begrenzten Umfangs der wissenschaftlichen Arbeit, werden im Folgenden die vier führenden Anbieter: Mendix, OutSystems, Microsoft und ServiceNow genauer betrachtet. Nach Einschätzung des Autors handelt es sich bei Mendix und OutSystems um Low Code Plattformen der Kategorie „GUI- und datenzentrierte integrierte Entwicklungsumgebungen“. Die Plattformen von Microsoft und ServiceNow zählen zu der Kategorie „Mehrzweckplattformen für die Konfiguration, Integration und Entwicklung von Geschäftsanwendungen“.

Mendix

Bei Mendix handelt es sich um eine gepflegte, kollaborative Plattform, die auf viele Nutzergruppen und Anwendungsfälle ausgerichtet ist. Die Vision besteht darin, dass eine einzige Plattform für die Entwicklung und

Zusammenarbeit zwischen Business und IT ausreicht. Die Übernahme durch Siemens im Jahre 2018 hat der Plattform zu neuen Kunden und Märkten wie z.B. China verholfen. Außerdem führte die Übernahme zu Synergien mit dem vielfältigen Portfolio und der Forschung und Entwicklung von Siemens. Als größte strategische Schwäche zählt laut dem Bericht von Forrester, das im Vergleich zu den Mitbewerbern sehr kleine Netzwerk an Implementierungspartnern.

Positiv:

+ Vielfältige Funktionen
+ Tools für Datenmodellierung und -verwaltung – insbesondere der Datenkatalog
+ Cloud-Native Entwicklung in Kombination mit passenden Bereitstellungsoptionen

Negativ:

- Begrenzte Möglichkeiten der Verbesserung
- Content Management System

Geeignet für: Unternehmen, die eine einzige Cloud-native und Cloud-wettbewerbsfähige Plattform für eine Vielzahl von Low-Code-Entwicklerrollen und -Projekten suchen (Bratincevic und Koplowitz 2021, S. 6).

<u>OutSystems</u>

OutSystems ist die bevorzugte Low Code Plattform bei Entwicklern und Systemintegratoren. Die Vision von OutSystems ist es, die Nutzer zu „Elite-Software-Performern“ zu machen. Systemintegratoren nutzen die Softwarelösung für reine Low-Code-Implementierungen und Kernanwendungsprojekte. Außerdem profitiert OutSystems von der hohen Marktrelevanz und Glaubwürdigkeit. Als strategische Schwäche gilt die Unabhängigkeit. Im Vergleich zu den anderen führenden Anbietern, verfügt OutSystems nicht über das Ökosystem und die Ressourcen.

Positiv:

+ Tools für die Entwicklung des Nutzererlebnisses, für Datenmodellierung und -management, für die App-Entwicklung sowie Support und Zusammenarbeit im Entwicklungsprozess.
+ Hohe Investitionen in KI
+ Generierung von zugänglichem Code
+ Schnelle Entwicklung kleinerer Anwendungsmuster und -erlebnisse

Negativ:

- Eingeschränkte Business-Development-Tools
- Content Management System

Geeignet für: Unternehmen, die den Fokus auf professionelle Entwickler legen und ein funktionsreiches Produkt mit Zugriff auf den Anwendungscode wünschen (Bratincevic und Koplowitz 2021, S. 6).

Microsoft

Der Anwender erhält bei der Nutzung von Microsoft Power Apps eine vielfältige Palette von Tools, eine klare Vision und verschiedene Wege der Markteinführung. Power Apps steht bei der Technologiestrategie von Microsoft im Mittelpunkt und wird von Forrester als leistungsstark und einzigartig bezeichnet. Durch das vernetzte Produktportfolio engagiert sich das Unternehmen für die Anforderungen professioneller Programmierer und die Technologie-Demokratisierung. Der Übergang zwischen den deklarativen Tools der Plattform und den programmierbaren Erweiterungen wird in dem Bericht von einem Referenzkunden als „elegant“ bezeichnet.

Positiv

+ Umfangreiches Produkt mit vielfältigen Funktionen
+ Integrationsfähigkeiten
+ Excel-ähnliche Fx-Entwicklungssprache
+ KI-Funktionen speziell für Citizen Developer
+ Tools für UX und die App-Entwicklung

Negativ

- Portfolio-Ansatz mit separater Lizensierung
- Nur als PaaS-Modell verfügbar

Geeignet für: Unternehmen, die ihre professionellen Programmierer und technisch interessierten Office-Anwender das Thema Low Code näherbringen möchten (Bratincevic und Koplowitz 2021, S. 7).

ServiceNow

ServiceNow besitzt aufgrund seiner installierten Basis und seiner Beliebtheit bei IT-Entwicklern eine starke Marktpräsenz. Mit ServiceNow können Benutzer schnell und einfach Anwendungen erstellen, die zur Automatisierung von Geschäftsprozessen beitragen. Die Plattform bietet auch Werkzeuge zur Verwaltung von IT-Services und zur Integration von Anwendungen.

Positiv

+ Vielfältige ausgereifte Funktionen
+ Prozessautomatisierung
+ Datenmanagement mit vielen Datenmodellen
+ Usability

Negativ

- Nur als PaaS-Modell verfügbar
- Geschäftsmodell ist nicht öffentlich und transparent

Geeignet für ServiceNow-Bestandskunden und Unternehmen, die auf der IT-Glaubwürdigkeit und der digitalen Workflow-Philosophie der Plattform aufbauen möchten, ohne sich auf ein vollständiges Toolset im BPMN-Stil festlegen zu müssen (Bratincevic und Koplowitz 2021, S. 7).

2.3.6 Vorteile

Mithilfe von Low Code Plattformen kann Software schnell und einfach gestaltet werden. Dadurch können die langen und kostenintensiven Entwicklungszyklen für Software verkürzt werden. Zusätzlich reduziert sich beim Einsatz von Low Code Plattformen die Wartezeit auf Programmierdienstleistungen von Programmierern (Adrian et al. 2020, S. 177; Degenhardt 2020, S. 1). All dies ist durch die modellbasierte Erstellung von Anwendungen möglich. Somit können Business Systeme flexibler an veränderte Unternehmensanforderungen angepasst werden, indem vorgefertigte Module wie etwa eine standardisierte Benutzerverwaltung zur Verfügung stehen. Darüber hinaus ermöglichen die Modelle eine bessere Abstimmung zwischen der Fachabteilung und der IT (Degenhardt 2020, S. 2; Vogel und Thomas 2020, S. 222). Ein weiterer Grund für die schnelle Softwareentwicklung ist die Drag-and-Drop-Methode der vorgefertigten Schaltflächen, welche langwierige Analyse-, Design- und Softwareentwicklungsprozesse vermeidet (Sahinaslan et al. 2021, S. 3). Eine Studie mit 451 Forschern bestätigt die bereits erwähnten Vorteile von Low Code Plattformen. Die Forscher kommen zu dem Ergebnis, dass die Entwicklungszeit um 50%-90%, im Vergleich zur klassischen Programmierung, reduziert werden kann (Pathfinder Report 2018, S. 8). Diese Reduzierung der Entwicklungsgeschwindigkeit ermöglicht es Entwicklern schneller erste Prototypen der Software zu präsentieren, was zu schnellerem Nutzer-Feedback führt und somit das Kundenerlebnis verbessert (Da Cruz et al. 2021, S. 3).

Low Code Plattformen begegnen dem Fachkräftemangel, indem auch Citizen Developer bzw. Fachexperten unternehmensspezifische Anwendungen umsetzen können. Schließlich sind Anwendungen, die mit Low-Code-Plattformen erstellt werden, häufig kostengünstiger als ähnliche Softwareprojekte, die zum Erstellen von Anwendungen erforderlich sind (Vogel und Thomas 2020, S. 222). Durch die Einbindung von Citizen Developer kann der Auslastungsgrad der IT-Fachabteilung gesenkt werden, sodass diese weniger über den Syntax[6] des Codes nachdenken müssen und sich mehr auf die Gestaltung von Ästhetik und Funktionalitäten der Anwendung fokussieren können. Außerdem muss weniger Zeit für Fehlersuche und Implementierung aufgebracht werden. Durch die Möglichkeit, Fachexperten direkt in den Entwicklungsprozess einzubinden, besteht ein hohes Potenzial eine passgenaue Lösung zu entwickeln, da diese als Prozessexperten die Anforderungen an die Software am besten kennen (Adrian et al. 2020, S. 177; Waszkowski 2019, S. 376).

Bei dem Einsatz von Low Code Plattformen ist nicht der Entwickler, sondern der Hersteller verantwortlich für die Anwendungssicherheit. Hersteller nehmen Verbesserungen vor, indem sie internationale Informationssicherheitsstandards (ISO / IEC 27001, PCIDSS etc.) berücksichtigen (Sahinaslan et al. 2021, S. 2). Somit können sich, wie bereits erwähnt, die Citizen Developer und Entwickler auf ihre Kernkompetenzen konzentrieren.

Vor dem Hintergrund dieser Vorteile ist es auch nicht verwunderlich, dass dem Markt für Low-Code-Entwicklungsplattformen ein hohes Wachstum prognostiziert wird. Die Unternehmensberatung Mordor Intelligence schätzte den Markt im Jahre 2021 auf 7,61 Milliarden USD und erwartet für das Jahr 2027 einen Marktwert von 36,43 Milliarden USD, was eine durchschnittliche jährliche Wachstumsrate von 29,81% bedeutet (Mordor Intelligence 2021).

2.3.7 Nachteile

Ein Nachteil von Low-Code-Plattformen sind die limitierten Möglichkeiten im Vergleich zur klassischen Programmierung. Es gibt Einschränkungen bezüglich der Programmierfunktionalitäten und grafischen Darstellungsmöglichkeiten, da es zu Beginn in den meisten Fällen nur möglich ist, die standardmäßigen Elemente zu konfigurieren und zu nutzen. Der Grund hierfür ist der häufig fehlende Zugriff auf den Quellcode und die

[6] Syntax ist ein System von Regeln zur Verbindung von Wörtern zu Wortgruppen und Sätzen (Duden).

Möglichkeit der Bearbeitung. Low Code Plattformen verbergen ihre Komplexität, indem sie eine Datenbank, ein Front-End, ein Back-End und einen generierten Quellcode beinhalten. Somit können die Citizen Developer der Plattform wenig bis gar nichts für die Wartung der von ihnen modellierten Anwendung tun, wenn die Plattform eingestellt wird, da in den meisten Fällen kein Zugriff auf den Quellcode möglich ist. Der "Quellcode", der dem Anwender zur Verfügung steht, besteht aus visuellen Darstellungen, die von der Low Code Plattform in einen Quellcode umgewandelt werden. Daraus resultiert die eigene Abhängigkeit gegenüber dem Anbieter der Low Code Plattform. Wenn der Anwender Fehler bemerkt, kann nur auf eine Aktualisierung durch den Plattformanbieter gewartet werden. Zusätzlich besteht bei der Zentralisierung der Softwareentwicklung durch eine Low Code Plattform das Problem, dass der Quellcode von der Plattform generiert wird und Fehler enthalten kann. Diese Fehler bzw. Schwachstellen werden von jeder Anwendung, die mit der Plattform erstellt werden, geerbt. Die Entscheidung für eine Low Code Plattform bringt außerdem die Gefahr des Lock-in-Effekts[7] mit sich, was eine ernsthafte Gefahr für den Investitionsschutz darstellt. Erstellte Anwendungen oder erzeugte Darstellungen können nur eingeschränkt exportiert werden, was bei einem Wechsel der Low Code Plattform die Möglichkeiten der Bearbeitung der bereits bestehenden Anwendungen enorm erschwert. (Frank et al. 2021, S. 53; Bock und Frank 2021, S. 738; Adrian et al. 2020, S. 185; Da Cruz et al. 2021, S. 3 ff.).

2.4 Überblick der Grundlagen

Die Digitalisierung der öffentlichen Verwaltung in Deutschland steht vor zahlreichen Herausforderungen, wie dem Fachkräftemangel, der Altersstruktur, dem Kulturwandel, Vorbehalten innerhalb der Organisation, dem Föderalismus, dem IT-Budget und der Schatten-IT. Um diese Herausforderungen zu bewältigen, empfiehlt es sich, die Vorteile beider Vorgehensweisen der Softwareentwicklung, der phasenorientierten und der agilen Methode, zu nutzen, indem man ein Grundgerüst oder feste Rahmenbedingungen definiert. Low Code Plattformen ermöglichen die schnelle und einfache Gestaltung von Software und reduzieren die Entwicklungszeit um 50%-90% im Vergleich zur klassischen

[7] Der dem Lock-in-Effekt ist eine Art Abhängigkeitsverhältnis von einem Kunden zu einem Anbieter. Der Effekt tritt ein, wenn ein Wechsel des Kunden zu einem anderen Anbieter durch hohe Wechselkosten unwirtschaftlich ist (wirtschafts-lehre 2016).

Programmierung. Sie bieten Flexibilität durch vorgefertigte Module und ermöglichen eine bessere Zusammenarbeit zwischen Fachabteilungen und IT. Sie sind auch für Citizen Developer geeignet und kostengünstiger als andere Softwareprojekte. Allerdings sind die Hersteller von Low Code Plattformen für die Anwendungssicherheit verantwortlich und haben Schwächen wie mangelnde Integration von bestehenden Systemen und begrenzte Skalierbarkeit.

3 Die Low Code Plattform A12

Im bisherigen Verlauf der wissenschaftlichen Arbeit konnten bereits die aktuellen Herausforderungen der öffentlichen Verwaltung und die Grundlagen von Low Code Plattformen bestimmt werden. Zusätzlich gilt es zu ermitteln, wie Low Code Plattformen die Digitalisierung der öffentlichen Verwaltung im Kontext der OZG Umsetzung vorantreiben können. Neben der allgemeinen Betrachtung von Low Code Plattformen soll die Enterprise Low Code Plattform A12 von mgm genauer betrachtet werden, um praktisch aufzuzeigen, wie die Digitalisierung der der öffentlichen Verwaltung gemeistert werden kann. Um dies zu erreichen, wird das Anwendungsgebiet, die Modellierung, die Vorteile, die Komponenten und Betriebsmöglichkeiten von A12 genauer betrachtet.

3.1 Einführung

Das Unternehmen mgm

mgm consulting partners wurde im November 2005 in Hamburg gegründet und ist ein Tochterunternehmen der mgm technoloy partners. 63 Mitarbeiter, an 6 Standorten, begleiten Kunden in den Branchen E-Commerce, Utilities (Energiewirtschaft), Versicherung, IT-Service Provider und Public Sector bei der digitalen Transformation. Der Umsatz betrug im Jahre 2021 13 Mio. € (mgm 2022b).

A12 Allgemein

A12 ist eine Enterprise Low Code Plattform, mit der Unternehmensanwendungen in komplexen IT-Landschaften umgesetzt werden. Dabei steht A12 für „Allianz 2012“ (mgm 2022a, S. 7).

Die Low Code Plattform besteht aus der Modellierungsplattform und der Runtime-Plattform.

- Die Modellierungsplattform stellt die benötigten Werkzeuge zur Verfügung, um Teile von Anwendungen zu erstellen und zu warten, ohne selbst programmieren zu können.
- Die A12-Laufzeitplattform bietet die Flexibilität eigene Low Code Anwendungen zu entwickeln und dies mit professioneller kundenspezifischer Softwareentwicklung bzw. Systemintegration zu kombinieren. Am Ende ergibt sich daraus eine vollständig integrierte Unternehmensanwendung (Mehmanesh et al. 2022, S. 5).

3.2 Anwendungsgebiet

- A12 ist für professionelle, kundenspezifische Software-Entwicklungs-Projekte gedacht. Dazu gehören formularbasierte Anwendungen, Portale und Verwaltungsprozesse für den öffentlichen Sektor sowie integrierte Lösungen für Online-, Einzel- und Versandhandel (Mehmanesh et al. 2022, S. 5).
- Aufgrund des modellbasierten Ansatzes zeigen sich die Stärken besonders in geschäftskritischen Bereichen, in denen Anwendungen lange bestehen, jedoch auf Grund von veränderten regulatorischen Vorgaben oder fachlicher Weiterentwicklung häufig angepasst werden müssen (mgm 2022a, S. 7).
- Häufig werden in den einzelnen Abteilungen eines Unternehmens eigene provisorische Lösungen erstellt, die zu Beginn als kleine Excel-Tabelle beginnen und mit der Zeit zu einer Anwendung mit Makros wachsen, die Probleme wie Datensicherheit und IT-Sicherheit mit sich bringen. Low Code Plattformen bieten Unternehmen bzw. Abteilungen die Möglichkeit, eigene Anwendungen zu entwickeln und gleichzeitig unternehmensspezifische Richtlinien zu berücksichtigen. Zusätzlich ist es schwierig, die eigens entwickelten Micro-Apps in Form von Excel-Tabellen in die heterogene IT-Landschaft zu integrieren. Einfacher ist es integrierte Unternehmensanwendungen mit einer Low Code Plattform zu entwickeln (Mehmanesh et al. 2022, S. 8).

3.3 Modellierungskonzept

- Der Unterschied von A12 gegenüber anderen Low Code Plattformen ist, dass A12 das “data first” Modelling-Paradigma verfolgt. Anstatt mit der Gestaltung der Benutzeroberfläche zu beginnen, wird bei A12 zu Beginn die Geschäftsbeziehung definiert. Dadurch kann anwendungsübergreifend und kontextübergreifend Fachwissen genutzt werden, was zu Synergieeffekten führt. Außerdem entsteht eine große Flexibilität für die Weiterentwicklung und Pflege von langlebiger Unternehmenssoftware (Mehmanesh et al. 2022, S. 20).
- Daraus ergeben sich folgende Modellierungsarten (Mehmanesh et al. 2022, S. 20 ff.):
    - <u>App modeling (Anwendungsmodellierung)</u>
      Das Grundgerüst einer Anwendung wird im Anwendungsmodellierer festgelegt. Es fungiert als eine Art Container für alle anderen Modelle.
    - <u>Workflow modeling (Arbeitsprozessmodellierung)</u>

Für die Modellierung von Geschäftsprozessen stützt sich A12 auf BPMN 2.0 (Business Process Model and Notation). Dokumentenmodelle beschreiben die von einem Prozess verwendeten Daten.

- UI modeling (Benutzeroberflächenmodellierung)

  Basierend auf den Datenmodellen, können spezifische Teile der Benutzeroberfläche mit den A12-Modellierungswerkzeugen erstellt werden. Die Modellierung folgt nicht dem Prinzip "what-you-see-is-what-you get". Die Modelle beschreiben die zugrunde liegende Strukturen der Benutzeroberflächen. Dadurch sind die Modelle unabhängig von der technischen Implementierung, wodurch Formulare erstellt werden können, ohne von einer bestimmten technischen Umsetzung abhängig zu sein. Hierfür gibt es drei verschiedene Modellierungsarten (mgm 2022a, S. 13):
  - Form models: Zur Bestimmung der Struktur und des Inhalts der Onlineformulare.
  - Overview models: Bietet mehre Möglichkeiten, tabellarische Daten anzuzeigen.
  - Tree models: Datenstrukturen können hierarchisch angezeigt und bearbeitet werden.
- Data modeling (Datenmodellierung)

  Mithilfe von Modellierungswerkzeugen können Business Analysten und Softwareentwickler Datenstrukturen der bestehenden Objekte, wie z.B. Verträge oder Produkte, erstellen und anpassen. Mithilfe der integrierten Kernelsprache können Validierungsregeln und Berechnungen angewandt werden, sowie Verknüpfungen zwischen Datenmodellen hergestellt werden.

  Bei der Datenmodellierung bleibt die Unternehmensexpertise und die Technologie getrennt voneinander. Dadurch können Geschäftsanwendungen verändert werden, ohne dass die Technik dementsprechend angepasst werden muss. Gleichzeitig kann die Technologie weiterentwickelt werden, ohne dass sich dies auf die Geschäftsanwendungen auswirkt, sodass man sich schnell an Änderungen anpassen kann.
- Output Model (Ausgabemodellierung)

  Durch das Output Model lassen sich Druckvorlagen für die Generierung barrierefreier PDFs erstellen (mgm 2022a, S. 15).

- Wiederverwendbarkeit

  Die Wiederverwendung ist bei allen Modelltypen möglich. Da sich die Datenmodelle modular aufbauen lassen, sind untergeordnete Submodelle bspw. in mehreren anderen Modellen wiederverwendbar. Zusätzlich können Typdefinitionen wie Länderlisten oder

Rechtsformen erstellt und in mehreren Modellen genutzt werden. Somit sind Doppelungen und etwaige Inkonsistenzen vermeidbar.

3.4 Modellierungstools

- Simple Model Editor (SME)

  Der Simple Model Editor ist das webbasierte zentrale Modellierungstool von A12. Zusätzlich gibt es aktuell noch den UI-Designer, der für die Gestaltung von Formularen eingesetzt wird. Es ist allerdings in Zukunft geplant, dass der UI-Designer in den Simple Model Editor integriert wird. Mit dem Simple Model Editor können Anwendungen, Beziehungen, Übersichten, Daten, Validierungen und Kalkulationen modelliert werden.
- UI Designer

  Der UI Designer wird für die Modellierung von Formularen eingesetzt. Elemente wie Eingabefelder, Schaltfläche, Beschriftungen, Checkboxen können damit organisiert bzw. strukturiert werden (Mehmanesh et al. 2022, S. 28 f.).

3.5 Vorteile

- Bei A12 besteht eine klare Trennung zwischen Fachlichkeit und Technik. Dies ist gerade bei geschäftskritischer Software, die viele Jahre besteht, von Vorteil. Dadurch können technische Innovationen realisiert werden, ohne alle fachlichen Inhalte der Anwendung berücksichtigen zu müssen (mgm 2022a, S. 8).
- Business Experten und Analysten, ohne Programmierkenntnisse, können die A12-Modellierungswerkzeuge nutzen, um für ihren Bereich passende Anwendungen zu kreieren und langfristig zu warten (Mehmanesh et al. 2022, S. 6; mgm 2022a, S. 8).
  - Businessaspekte können ohne Programmierkenntnisse angepasst werden.
  - Schnelle Umsetzung von fachlichen Änderungen.
  - Automatisierung des Softwareentwicklungsprozesses.
- A12 ist ein offenes System. Es liefert eine große Menge an Flexibilität für Softwareentwicklung, langfristige Wartung und weitere Entwicklung (Mehmanesh et al. 2022, S. 6).
  - Modulare Nutzung von Laufzeitkomponenten.
  - Einsatz von Open Source Technologie.
  - Volle Kontrolle über den Betrieb von A12.
- Mithilfe der konsequenten fachlichen Trennung in A12 kann langlebige Software entwickelt werden, da der geschäftsspezifische Kern bei

Technologiesprüngen erhalten bleibt (Mehmanesh et al. 2022, S. 6; mgm 2022a, S. 9).

3.6 Vorgehensweise bei der Softwareentwicklung

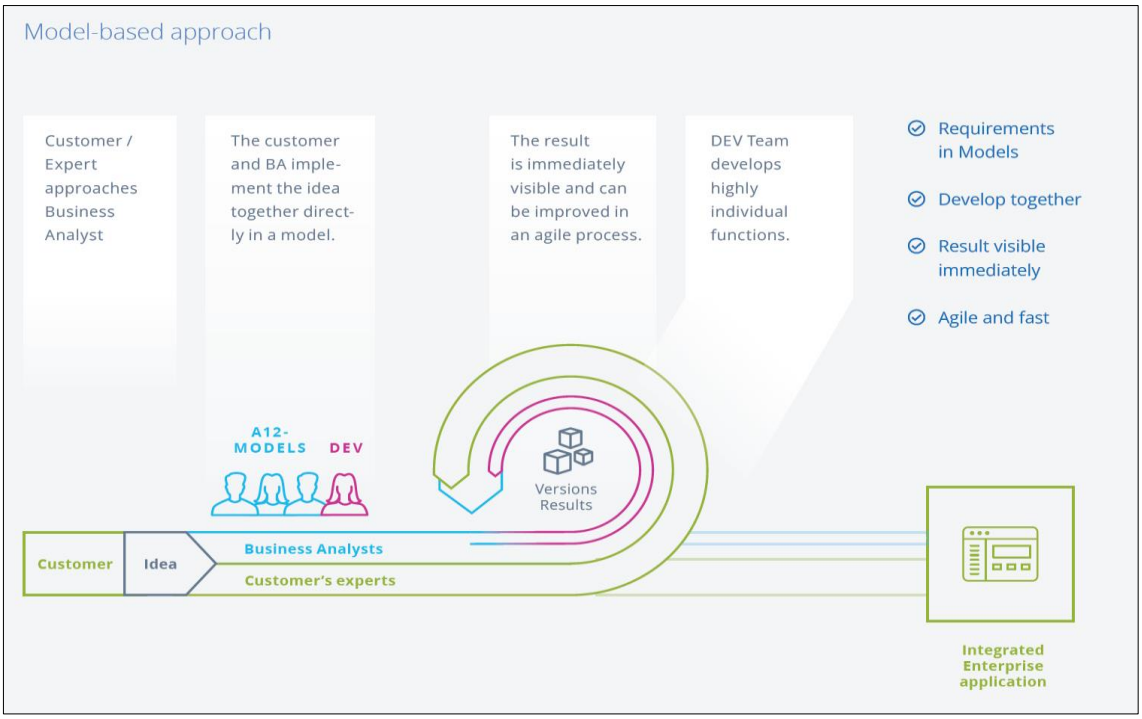

Abbildung 5: Softwareentwicklungsprozess mit A12 (Mehmanesh et al. 2022, S. 33)

Die Softwareentwicklung mit A12 läuft in der Regel folgendermaßen ab (Mehmanesh et al. 2022, S. 33):

1. Der Customer's Expert[8] und der Business Analyst[9] arbeiten gemeinsam die Anforderungen für die Software aus und setzen diese anschließend gemeinsam mit den Modellierungswerkzeugen in A12 um.
2. Die Ergebnisse der Modellierung können in agiler Form angepasst werden.
3. Die Softwareprogrammierer kümmern sich um individuelle Funktionen der Software, die mit der Modellierung nicht umgesetzt werden können.

[8] Der Customer Expert ist der Experte auf Seiten des Kunden, der mit den internen Prozessen und den eigenen Anforderungen vertraut ist.

[9] Der Business-Analyst hilft dem Entwickler, indem er Teile der Anwendung selbständig entwirft (Mehmanesh et al. 2022, S. 19).

3.7 Architektur

Die Architektur von A12 lässt sich in die Anwender- und Serverseite trennen.

Anwenderseite: Der BAP-Client basiert auf einer mehrstufigen Architektur und bietet auf der Anwenderseite die Grundlage, um schnell modulare Front-Ends zu erstellen (siehe Abbildung 6). Die meisten Aspekte des BAP-Clients können mit Plug-ins angepasst und überschrieben werden. Der BAP-Client basiert auf einer mehrstufigen Architektur. Der gesamte Code ist in TypeScript geschrieben (eine beliebte JavaScript Variante) und basiert auf React, Redux und Erweiterungen wie Redux Saga (Mehmanesh et al. 2022, S. 34 f.).

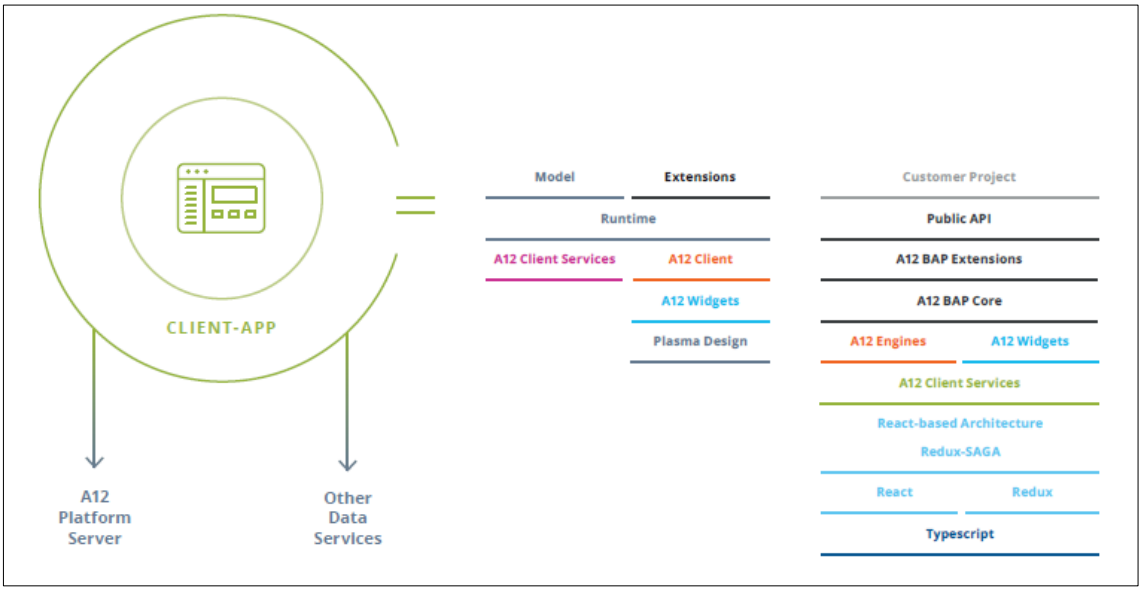

Abbildung 6: A12 Architektur – Anwenderseite (Mehmanesh et al. 2022, S. 35)

Serverseite: A12 bietet serverseitige Anwendungen, die mit A12-Client-Anwendungen über A12-Tools kommunizieren. Die Serverseite beinhaltet eine Workflow-Engine, die Möglichkeit sich zu authentifizieren bzw. einzuloggen und Nutzer bzw. Rollen zu verwalten (Mehmanesh et al. 2022, S. 34 ff.). Die Kernfunktionalität können über eine REST-API abgerufen werden (siehe Abbildung 7).

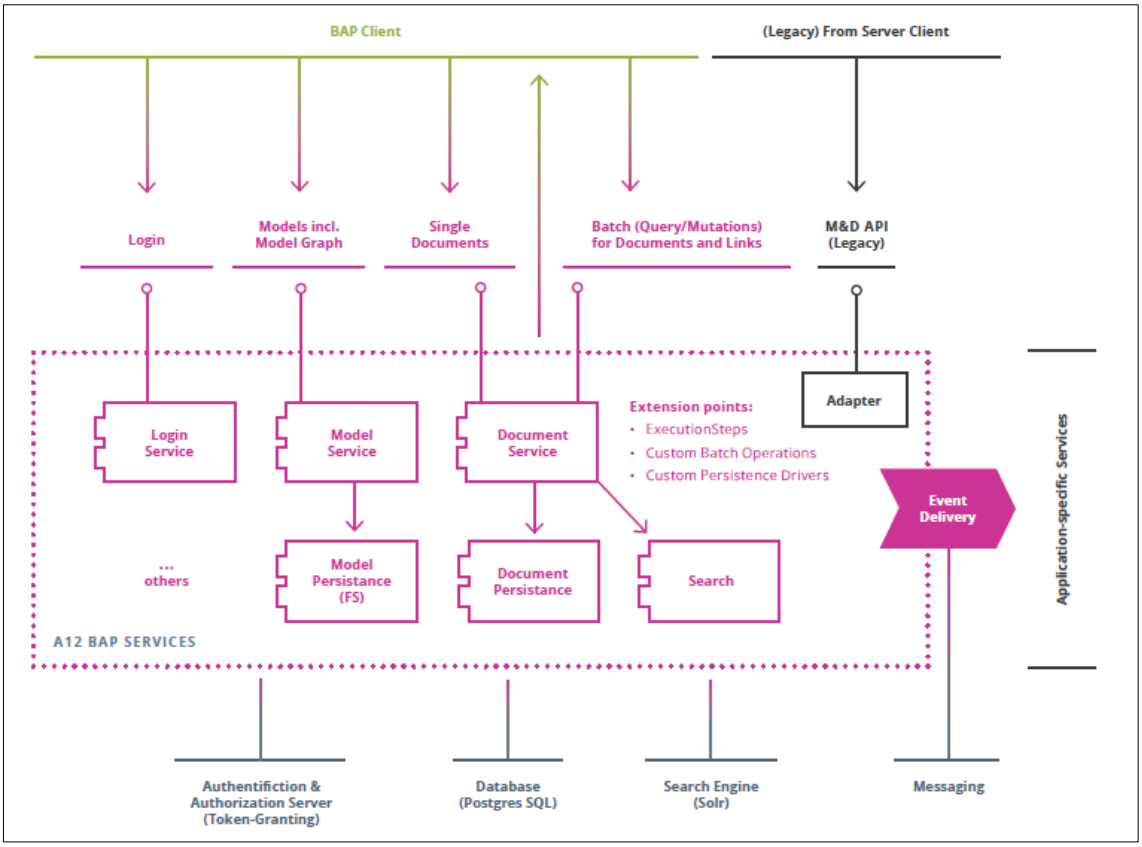

Abbildung 7: A12 Architektur – Serverseite (Mehmanesh et al. 2022, S. 36)

3.8 Komponenten

- Widgets
  Widgets sind wiederverwendbare Komponenten der Benutzeroberfläche. Sie unterstützen Unternehmensanwendungen, die auf Desktop-PCs, Tablets und Smartphones laufen (Mehmanesh et al. 2022, S. 39).
- Engines
  Engines sind in sich geschlossene Laufzeitkomponenten, die Daten und UI-Modelle interpretieren (Mehmanesh et al. 2022, S. 40).
- Client
  Die modellgesteuerte, client-seitige Laufzeitkomponente ermöglicht es, die Kernbereiche der Anwendung zu bestimmen. Darunter fallen Module, Navigation, Bildschirme und die wichtigsten Interaktionsmuster. Die Hauptaufgabe ist die Orchestrierung anderer UI-Komponenten, insbesondere der A12-Engines (Mehmanesh et al. 2022, S. 41).
- Kernel
  Die Kernelkomponente bündelt grundlegende Funktionen zur Erstellung und Bearbeitung von Datenmodellen. Dazu gehören alle

Grundlagen für Validierungen und Berechnungen, die Teil der Geschäftsmodellierung sind (Mehmanesh et al. 2022, S. 42).

- Data Services

  Die Komponente Data Services bietet eine API für die Verwaltung von Modellen. Sie umfasst Routinen für Client/Server-Kommunikation, Authentifizierung, Autorisierung, Validierung, Persistenz und Indizierung (Mehmanesh et al. 2022, S. 43).
- User Management, Authentication & Authorization (UAA)

  Die UAA-Komponenten von A12 bieten eine hochflexible und leistungsfähige Autorisierungslösung, die Zugriffsrechte auf verschiedenen Ebenen bereitstellen kann. Es können sowohl rollenbasierte als auch komplexere, attributbasierte Regeln festgelegt werden, sodass der Zugriff bis auf die Feldebene von Datendokumenten geschützt ist (Mehmanesh et al. 2022, S. 44).
- Workflows

  A12 Workflows bieten einen einfachen Service, der Geschäftsprozessmodell und -notation (BPMN) in A12 integriert. Dies ermöglicht die grafische Modellierung von serverseitigen Arbeitsabläufen und deren Ausführung (Mehmanesh et al. 2022, S. 45).

3.9 Betriebsmöglichkeiten

- Folgende Installationsmöglichkeiten sind bei A12 vorhanden:
  1. On-Premise-Betrieb im unternehmenseigenen Rechenzentrum
  2. Betrieb in der Private-Cloud von mgm, gehostet in einem deutschen Rechenzentrum
  3. Cloud-Betrieb bei beliebigem Cloud-Anbieter
- A12 bietet allen Kunden die Möglichkeit über die eigenen Daten und die Software selbstständig zu entscheiden bzw. diese zu verwalten. Sie behalten die volle Kontrolle, auch wenn die Anwendungen hochkomplex und integriert sind (Mehmanesh et al. 2022, S. 14).

3.10 Zusammenfassung

A12 ist eine Enterprise Low Code Plattform, die von dem Unternehmen mgm angeboten wird. Sie ist für professionelle, kundenspezifische Software-Entwicklungs-Projekte geeignet und setzt bei der Gestaltung von Projekten auf die Definition der Geschäftsbeziehungen, anstatt mit der Benutzeroberfläche zu beginnen. A12 ist besonders nützlich für geschäftskritische Software, die viele Jahre bestehen wird. Im nachfolgenden Kapitel wird die Methodik der wissenschaftlichen Arbeit genauer betrachtet.

4 Empirische Untersuchung

In diesem Kapitel wird die empirische Untersuchung vorgestellt, die im Rahmen der wissenschaftlichen Arbeit angewendet wird. Die Wahl der geeigneten Methodik ist von großer Bedeutung, da sie dazu beiträgt, die Zuverlässigkeit und Validität der Ergebnisse zu gewährleisten. Dabei wird die Methodenauswahl, die Expertenauswahl, die Vorbereitung, der Leitfaden und die Auswertungsmethode genauer betrachtet.

4.1 Methodenauswahl und -beschreibung: Experteninterviews

Die Auswahl der passenden empirischen Methode ist abhängig von dem entsprechenden Forschungsinteresse. Dabei wird zwischen der quantitativen und qualitativen Erhebungsmethode unterschieden. Die quantitative Sozialforschung ist durch das Zusammentragen von Daten, u.a. mittels standardisierter Fragebögen gekennzeichnet. Bei der qualitativen Forschung hingegen wird der Ansatz verfolgt, Lebenswelten aus der Betrachtungsweise der handelnden Person auszudrücken, indem Einstellungen, individuelle Werte, Abläufe oder Denkmuster sichtbar gemacht werden (Flick et al. 2000, S. 14). Für den Zugang zu diesen Lebenswelten sind qualitative Interviews als Erhebungsform geeignet, da sie dem Experten den Spielraum bzw. die Möglichkeit bietet, das eigene Wissen und die eigenen Erfahrungen vorzutragen.

Experteninterviews sind eine verbreitete und ausdifferenzierte Methode, qualitative Daten systematisch zu erheben. Dabei werden die Befragten aufgrund des ihnen zugeschriebenen Status als Experten interviewt. Als Experten gelten grundsätzlich alle Akteure einer Organisation oder Rollenträger in gesellschaftlichen Teilfeldern, die als Interviewpartner zur Rekonstruktion sozialer Prozesse und damit zur Analyse der Funktionsweise einer Organisation oder eines sozialen Systems beitragen (Helfferich 2022, S. 875 ff.; Blöbaum et al. 2016, S. 176). Experten können als sachkundige Wissensvermittler oder als Ratgeber agieren, die Erfahrungs- und Faktenwissen weitergeben und daher mit verhältnismäßig geringem Aufwand als wertvolle Quelle von Spezialwissen über den zu erforschenden Sachverhalt gelten (Helfferich 2022, S. 877; Gläser und Laudel 2010, S. 12; Bogner et al. 2014, S. 9). Das bedeutet, dass Experte zu sein keine personale Eigenschaft oder Fähigkeit ist, sondern eine Zuschreibung. Diese Zuschreibung geschieht in der Praxis, wenn aufgrund von spezifischem Forschungsinteresses gewisse Personen mittels der durchgeführten

Vorbereitung als Experte identifiziert werden (Bogner et al. 2014, S. 11). Aus forschungspraktischen Gründen ist ein positiver Aspekt beim Experteninterview der vergleichsweise unproblematische Zugang, da relevante Experten leicht zu identifizieren und in der Regel kooperationsbereit sind. Zusätzlich verfügen Experten für gewöhnlich über eine hohe sprachliche und soziale Kompetenz, was für die Erhebung und der Auswertung von Daten hilfreich ist bzw. Vorteile bietet (Bogner und Menz 2009, S. 8 ff.). Über die Bestimmung des Interviewten als Experten wird Anerkennung in dem Sinn vermittelt, dass Erfahrungen und Kompetenzen unterstellt werden und dass es sich um relevantes Wissen handelt, da es verallgemeinerbar und gültig ist (Helfferich 2022, S. 888). Aus diesem Grund werden in der wissenschaftlichen Arbeit Experteninterviews geführt, die wichtige Erkenntnisse zu den Potenzialen von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung liefern.

4.2 Expertenauswahl und Darstellung der Stichprobe

Eine der wichtigsten Entscheidungen bei Experteninterviews ist die begründete und zielgerichtete Determinierung von Experten. Dabei ist neben dem Wissen auch die Erreichbarkeit und Bereitschaft der befragten Interviewpartner entscheidend (Gläser und Laudel 2010, S. 117). Daraus ergibt sich für diese Arbeit eine Auswahl von 15 Experten, die für ein Interview zur Verfügung stehen. Aus statistischer Perspektive ist es bei der Expertenanzahl nicht möglich, eine Repräsentativität herzustellen. Dies wird in der qualitativen Sozialforschung als inhaltliche Repräsentativität bezeichnet. Die Überlegung dahinter ist, dass die befragten Experten nicht als einzelne Person aufgefasst werden, sondern eine Gruppe repräsentieren (Mayer 2008).

Bei der konkreten Auswahl der Experten wird versucht, so viele Perspektiven wie möglich einzuholen, um die Potenziale von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung bestmöglich zu ermitteln. Dabei wird grundsätzlich zwischen internen Mitarbeitern von mgm und externen Experten unterschieden. Außerdem spielt es eine entscheidende Rolle, ob sich die Person mit dem ganzheitlichen Thema Low Code Plattformen beschäftigt bzw. eine Expertise bezüglich der Low Code Plattform A12 aufweist.

Tabelle 1: Übersicht der Experten

Interview	Tätigkeit	Unternehmen	Datum des Interviews
P1	Steuerexperte	mgm technology partners	05.10.2022
P2	Wissenschaftliche Mitarbeiterin	Fraunhofer FOKUS	06.10.2022
P3	Projektleiter/Principal Consultant	mgm consulting partners	10.10.2022
P4	Business Analyst	mgm technology partners	11.10.2022
P5	Consultant	mgm technology partners	13.10.2022
P6	Business Analyst	mgm technology partners	19.10.2022
P7	Managing Consultant	adesso SE	25.10.2022
P8	Business Developer Intrexx/Low Code Plattformen	HanseVision	25.10.2022
P9	Preojektleiterin	mgm technology partners	26.10.2022
P10	Steuerung technische Konzeption	Freie und Hansestadt Hamburg	02.11.2022
P11	Fachlicher Chefarchitekt	Elster	08.11.2022
P12	Manager	BearingPoint	21.11.2022
P13	Bereichsleiter Public Sector	mgm technology partners	24.11.2022
P14	CEO	mgm technology partners	29.11.2022
P15	Senior Business Solution Manager	Zühlke Group	09.12.2022

4.3 Vorbereitung

Die Vorarbeit bei Experteninterviews ist für die Erhebung von relevanten Informationen von entscheidender Bedeutung. Je ausführlicher die Vorbereitung auf ein Interview ist, desto qualitativ hochwertiger sind die Aussagen der Experten. Dadurch können Potenziale für die Softwareentwicklung mit Low Code Plattformen besser ermittelt werden. Es gilt zu vermeiden, Informationen abzufragen, die aus alternativen Quellen leicht zugänglich sind und daher als Zeichen von mangelndem Respekt oder von Bequemlichkeit interpretiert werden können. Allerdings kann um eine Bestätigung von Informationen gebeten werden (Helfferich 2022, S. 889). Bezüglich des Vorwissens stellen sich bei Experteninterviews zwei Fragen: Wie viel Vorwissen und Recherche sind über die Person und den Werdegang des jeweiligen Experten notwendig? Und wie viel Wissen sollten Forschende in ihren jeweiligen Fachgebieten aufweisen? Vorkenntnisse in beiden Bereichen können dazu führen, dass man als kompetenter Gesprächspartner wahrgenommen und ernst genommen wird. Mangelndes Vorwissen kann den Befragten verärgern und das Gespräch gefährden. Zwischen einem Gespräch auf Augenhöhe, in der sich der Forscher als Experte auf dem jeweiligen Fachgebiet präsentiert, und der hierarchischen Kluft zwischen Experte und Forscher lassen sich viele Diskussionskonstellationen denken

und diskutieren (Buber und Holzmüller 2009, S. 458). Üblicherweise sind Informationen über den Experten wie berufliche Stellung, Aufgaben im Unternehmen, Berichte über bzw. Kommentare von der identifizierten Person öffentlich zugänglich und unterstützen die Präzisierung des Leitfadens. Es bietet sich bei der Entwicklung des Leitfadens an, die recherchierten Informationen über den jeweiligen Experten mit einfließen zu lassen. Dies hat zur Folge, dass personenbezogene Daten in die Frage mit einbezogen werden können, bei Widersprüchen zu anderen Quellen nachgehakt werden oder an Aussagen und relevante Zeitabschnitte erinnert werden kann (Scheu 2011, S. 75). Darüber hinaus können die Aussagen aus den Interviews mit den recherchierten Hintergrundinformationen in Beziehung gesetzt werden, um die Validität der in den Interviews erhaltenen Informationen zu überprüfen (Blöbaum et al. 2016, S. 185).

4.4 Leitfaden ausarbeiten

Bei dem im Rahmen der qualitativen Sozialforschung durchgeführten Experteninterviews handelt es sich um teilstrukturierte Interviews, bei der zur Vorbereitung und Durchführung Leitfäden entwickelt werden. Leitfäden besitzen zwei Funktionen, da sie zur Strukturierung des Themenfelds der Untersuchung dienen und ein konkretes Hilfsmittel in der Erhebungssituation sind. Dadurch ist es möglich, in den Experteninterviews, die bestmögliche Informationsgewinnung zu erzielen, damit die Potenziale von Low Code Plattformen optimal ermittelt werden können. Daher sollte bei der Leitfadenkonstruktion ausreichend Zeit eingeplant und sorgfältig gearbeitet werden. Im Gegensatz zur quantitativen Sozialforschung, bei der eine Standardisierung der Fragen notwendig ist, müssen bei qualitativen Interviews nicht immer die gleichen Fragen gestellt werden, um eine Vergleichbarkeit herzustellen. Grundsätzlich sind Einleitungs- und Filterfragen sinnvoll, um das Gespräch bzw. den Themenblock zu beginnen oder Einfluss auf den Verlauf des Interviews zu nehmen. Durch den Einsatz unterschiedlicher Fragetypen kann die ungewohnte Befragungssituation angenehmer gestaltet, die geforderte Offenheit eines Experteninterviews ermöglicht, sowie der Interviewablauf hinsichtlich der notwendigen Informationen positiv beeinflusst werden (Gläser und Laudel 2010, S. 129). Wichtig bei der Erhebung ist, den Leitfaden primär als Gedächtnisstütze und nicht als Redeskript zu nutzen, bei dem der Interviewer alle Fragen nacheinander abliest (Bogner et al. 2014, S. 27 ff.)

Für die Konstruktion des Leitfadens hilft es, zu dem Themenblock jeweils ein bis drei Hauptfragen aufzuschreiben, die als Gesprächsanreize zum jeweiligen

Hauptthema dienen. Situationsbedingt werden während des Gesprächs zusätzliche Nachfragen formuliert, die auf dem Leitfaden nicht zu finden sind. Die Themenblöcke sollten so strukturiert sein, dass sie mehr oder weniger in sich geschlossen sind und als Gesprächsbausteine je nach Situation in der Reihenfolge verschoben werden können (Bogner et al. 2014, S. 27 ff.)

Als Hilfestellung auf dem Weg zum Leitfaden, zählen Bogner et al. folgende sechs Schritte auf (2014, S. 32 ff.):

1. Sammlung und Systematisierung
   Sämtliche Forschungsfragen und -hypothesen, die bereits zur Verfügung stehen, werden bspw. in Form einer Mindmap, zusammengestellt. Anschließend geht es um die Systematisierung und Reduzierung der zur Verfügung stehenden Informationen, um die Forschungsfrage zu spezifizieren.
2. Methodenspezifizierung
   Bei der Methodenspezifizierung gilt es herauszufinden, ob es zur Beantwortung der einzelnen Forschungsfragen und -hypothesen eine verlässlichere Quelle als Experteninterviews gibt. Dabei stellt sich ggf. die Frage, welche Materialien anstelle des Interviews verwendet werden können. Schlussendlich werden die Fragen aussortiert, die mithilfe von Experteninterviews nicht bestmöglich beantwortet werden können.
3. Gruppierung
   Die verbleibenden Fragen werden sortiert und in Themenblöcken zusammengefasst. Für jeden dieser Themenblöcke werden Oberbegriffe und Leitfragen formuliert, denen dementsprechende Themen und Fragen zugeordnet werden.
4. Entwurf von Leitfadenfragen
   Bei dem Entwurf der Leitfadenfragen steht die Erarbeitung von Forschungsfragen nicht mehr im Fokus, sondern die Formulierung von Interviewfragen – Fragen, die im eigentlichen Erhebungsgespräch tatsächlich gestellt werden.
5. Differenzierung von Fragetypen
   Im fünften Schritt wird zischen verschiedenen Leitfragen differenziert. Dabei ist wichtig, welche Fragen sich als Hauptfrage eignen, welche untergeordnet sind und welche Nachfragen sinnvoll sind.
6. Pretest
   Wenn es die Situation erlaubt, sollte der Leitfaden getestet werden, bevor er zum Einsatz kommt. Dabei wird die Funktionalität des Leitfadens reflektiert und geprüft, ob sich der Leitfaden in der Praxis halbwegs umsetzten lässt.

Gemäß der Hilfestellung von Boger kann ein umfangreicher Leitfaden entwickelt werden. Da bei den Experteninterviews versucht wird verschiedene Perspektiven zu betrachten, unterscheiden sich die Experten in

Bezug auf ihren Hintergrund und ihre Expertise. Daher wird, abhängig von dem Experten, individuell entschieden, welche Fragen des Leitfadens gestellt werden.

Interviewleitfaden für die Experteninterviews

Gesprächseröffnung

- Können Sie sich kurz vorstellen und Ihre eigene Tätigkeit beschreiben?
    - Wie steht diese Tätigkeit im Zusammenhang mit Low Code?
    - Wie steht diese Tätigkeit im Zusammenhang mit A12?

Einführung in Low Code/A12

- Welche Unterschiede sehen Sie zwischen der Softwareentwicklung mit und ohne Low Code Plattformen in der öffentlichen Verwaltung? Und welche Vorteile ergeben sich daraus bei der Nutzung von Low Code Plattformen?
- Welche Voraussetzungen müssen erfüllt sein, um Low Code Plattformen für die Softwareentwicklung in der öffentlichen Verwaltung erfolgreich einzusetzen (Infrastruktur, Expertise, Fachlichkeit)?
- Welche Voraussetzungen müssen erfüllt sein, damit A12 erfolgreich eingesetzt werden kann?

Expertenmeinung

- Was ist für dich der größte Vorteil von Low Code Plattformen in der öffentlichen Verwaltung bzw. welche Probleme können gelöst werden?
    - Wie verändert sich die Arbeit der Entwickler auch in Bezug auf den Aufwand?
- Was ist der Vorteil von A12 gegenüber der klassischen Softwareentwicklung bzw. welches Problem wird gelöst?
- Bei welcher Art von Software/Anwendungen sind Low Code Plattformen besonders geeignet? Wann ist es eher unpassend?
- Kann die absolute Zeit, die Softwareentwickler für ein Projekt mit Low Code/A12 aufbringen müssen, reduziert werden im Vergleich zu Projekten mit klassischer Softwareentwicklung?
- Welche Herausforderungen müssen bewältigt werden, damit Low Code noch häufiger eingesetzt wird?
    - Wird eine IT-Affinität für die Nutzung von Low Code Plattformen vorausgesetzt?
- Gibt es aktuell Probleme oder Herausforderungen bei der Softwareentwicklung mit A12?
    - Welche IT-Kenntnisse sind für die Nutzung von A12 erforderlich? Wird eine IT-Affinität vorausgesetzt?
- Gibt es Verbesserungen in der Softwareentwicklung mit A12, die Sie sich wünschen?
- Welche Schritte sind notwendig, um das Potenzial zur Reduzierung der Schatten-IT mithilfe von Low Code Plattformen maximal auszunutzen?
    - Welche Voraussetzungen müssen hierfür erfüllt sein?
- Kann die Schatten-IT durch den Einsatz von A12 in der Softwareentwicklung reduziert werden?
    - Wenn ja, welche Voraussetzungen müssen erfüllt sein?

- Welche Nachteile gibt es beim Einsatz von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung?
- Welcher Nachteil besteht bei der Softwareentwicklung mit A12 gegenüber der klassischen Softwareentwicklung?
- Können Sie beurteilen, ob die Usability der Software durch den Einsatz von A12 gestiegen ist?

Ausblick

- Inwieweit müssen sich Low Code Plattformen weiterentwickeln, damit sich das Einsatzgebiet in der öffentlichen Verwaltung erweitert?
- Welche zukünftigen möglichen Anwendungsgebiete sehen Sie beim Einsatz von Low Code Plattformen, die aktuell noch nicht oder selten umgesetzt werden?
- Fallen Ihnen weitere Einsatzmöglichkeiten von A12 ein, die bisher nicht zum Einsatz kommen?

Der Leitfaden wird mit einer kurzen Einführung zur Masterarbeit eingeleitet, um dem Experten den benötigten Kontext zur Beantwortung der Fragen zu liefern. Zusätzlich wird auf den fehlenden Sperrvermerk hingewiesen und nach der Erlaubnis zur Aufzeichnung gefragt. Im Nachgang werden die Antworten auf die Fragen den Experten per E-Mail zugesandt. Diese haben im Anschluss die Möglichkeit, auf Missverständnisse, vertrauliche Informationen, die nicht veröffentlicht werden sollen und Fehler hinzuweisen. Die Notizen der Experteninterviews befinden sich im Anhang der wissenschaftlichen Arbeit.

4.5 Auswertungsmethode Qualitative Inhaltsanalyse

In der Wissenschaft wird das Experteninterview häufig zur Informationsgewinnung eingesetzt. Wenn die Informationsgewinnung im Fokus steht, dann ist die qualitative Inhaltsanalyse das passende Auswertungsverfahren. Der Vorteil liegt in der Kontrollierbarkeit der Materialidentifikation.

Das Ziel ist es, die Information der Experteninterviews bestmöglich auszuwerten, damit die Potenziale bei der Softwareentwicklung mit Low Code Plattformen erkennbar sind. Im Kern geht es darum, Texte systematisch zu analysieren, indem schrittweise das Material mit theoriegeleitetem Kategoriesystem bearbeitet wird. Im Einzelfall mögen die Informationen selektiv und widersprüchlich sein, weswegen man mehrere Interviews führt. Jedoch geht man prinzipiell davon aus, dass das Expertenwissen in der Lage ist, die Welt korrekt abzubilden. Durch die systematische Analyse und den Vergleich der Informationen, die Experten liefern, sollen mithilfe der qualitativen Inhaltsanalyse Kausalabhängigkeiten aufgedeckt werden (Bogner et al. 2014, S. 72 f.; Mayring 2016, S. 114).

Für den Ablauf der qualitativen Inhaltsanalyse gibt es je nach Forschungsprojekt verschiedene Techniken, die dabei zum Einsatz kommen können. Der Fokus dieses Konzeptes ist es, so wie bei allen inhaltsanalytischen Vorgehen, ein Kategoriesystem auf das untersuchte Material anzuwenden. Daraus ergeben sich folgende fünf Schritte (Mayring und Brunner 2009, S. 674; Gläser und Laudel 2010, S. 206).

1. Fragestellung und Materialauswahl

   Zu Beginn sind die Fragestellung und die konkrete Frageperspektive zu bestimmen. Bei der Bestimmung der Frageperspektive muss festgelegt werden, unter welcher Perspektive der Interviewtext gelesen werden sollte. Bei informatorischen Interviews geht es um den Inhalt der Interviews und nicht um den emotionalen Zustand der Interviewten.
2. Aufbau eines Kategoriesystems

   Unter einem Kategoriesystem versteht man eine durch Abhängigkeitsbeziehungen strukturierte Vielzahl von Kategorien. Darin enthalten sind daher die einzelnen Kategorien sowie deren Beziehung zueinander. Diese werden nach einem vorgegebenen Definitionskriterium aus dem Material heraus entwickelt.
3. Extraktion

   Der dritte Schritt besteht darin, den Interviewtext systematisch zu „entpacken“ und daraus eine thematisch geordnete Sammlung relevanter Informationen zu machen. Dazu wird der Text nach relevanten Informationen durchsucht, d.h. die Rohdaten werden dem Kategoriensystem zugeordnet.
4. Aufbereitung der Daten

   Im vierten Schritt geht es im Wesentlichen darum, die Qualität der Datenbasis zu verbessern. Das bedeutet, über verschiedene Interviews verstreute, aber inhaltlich zusammengehörige Informationen zu aggregieren, redundante Informationen zu reduzieren und bei einem Textvergleich offensichtliche Fehler der Befragten zu korrigieren. Zusätzlich werden unklare Stellen durch die schrittweise Analyse ihres Textkontextes erklärt.
5. Auswertung

   Bei der Auswertung geht es um die Beantwortung der Forschungsfrage auf Grundlage der verarbeiteten Daten. Dabei soll geklärt werden, welche Faktoren zu dem Ergebnis führen und welche Ursachen welche Wirkung hervorrufen. Zusätzlich ist noch die vergleichende Analyse von Fällen relevant. Welche Kausalfaktoren treten in allen Fällen auf und welche nur gelegentlich? Lassen sich bestimmte Muster erkennen?

5 Ergebnisse

Im folgenden Kapitel richtet sich der Blick auf die Darstellung der Ergebnisse der empirischen Untersuchung. Diese sollen genutzt werden, um die Forschungsfrage zu beantworten und die Ziele der wissenschaftlichen Arbeit zu erreichen. Hypothesen der Literatur, bezüglich des Begriffs und den Vorteilen von Low Code Plattformen, werden zu Beginn mit den Erkenntnissen aus den Experteninterviews gegenübergestellt. Des Weiteren sind die Ergebnisse über Low Code Plattformen und A12 nach dem Einsatzgebiet, der Anwendung und der Zukunftsperspektive gegliedert.

5.1 Begriffserklärung und Vorteile von Low Code Plattformen

Für P7 ist Low Code mehr ein Ansatz und weniger eine Technologie. Dabei handelt es sich um einen sehr breiten Begriff, der bisher keine allgemeingültige Definition besitzt. Dies zeigt sich daran, dass sowohl das Content-Management-System WordPress als auch PowerBI zur Erstellung von Dashboards als Low Code Plattformen bezeichnet werden. Somit sind nicht alle Ergebnisse dieses Kapitels pauschal auf alle Low Code Plattformen übertragbar. P2 erklärt, dass sich eine Low Code Plattform aus verschiedenen Komponenten zusammensetzt, die schon seit längerem in der Softwareentwicklung eingesetzt werden. Das Neue daran ist, dass diese Komponenten nahtlos in einer Plattform vereint werden.

5.1.1 Allgemein

Eine Hypothese, aus der Literatur, bezüglich der Vorteile einer Low Code Plattform, ist die Begegnung des Fachkräftemangels, da Fachexperten eigene Anwendungen entwickeln können und somit nicht mehr so stark auf die Softwareentwickler angewiesen sind. Somit können sich die Fachexperten auf ihre Kernkompetenzen konzentrieren, indem sie die Software entwickeln, die sie selbst auch nutzen (siehe Kapitel 2.3.6 Vorteile). P8 erwähnt ebenfalls die entstehenden Vorteile der Anwendungsentwicklung durch Fachexperten und sieht einen großen Vorteil von Low Code Plattformen darin, dass die Fachlichkeit von der Technik getrennt ist und sich somit der Anwender nicht um die technischen Systeme, die im Hintergrund laufen, kümmern muss. Der Fachexperte kann sich somit vollkommen auf die Fachlichkeit fokussieren. Bei der klassischen Programmierung muss der Anwender, laut P8, bei einer neuen Version des Codes sicherstellen, dass die Kommunikation zu den Datenbanken und Betriebssystemen weiterhin funktioniert. Häufig stammen

die Komponenten der IT-Infrastruktur von mehreren Anbietern, was dazu führt, dass das System fehleranfällig ist und es ggf. einiges zu beachten gibt. Bei Low Code Plattformen besteht, laut P12, der Vorteil darin, dass das gesamte Ökosystem voll integrierbar an einem Ort gebündelt ist. Dadurch ist die Infrastruktur untereinander abgestimmt, wodurch neue Versionen erst von dem Plattformhersteller freigegeben werden, wenn diese ausgiebig getestet sind. Damit der Fachexperte seine Kernkompetenzen in die Softwareentwicklung einbringen kann, stehen ihm je nach Plattform verschiedene Konfigurationswerkzeuge zur Verfügung. P5 ist der Überzeugung, dass dadurch Missverständnisse und Mehrfachdeutungen bei der Anforderungsermittlung vermieden werden können. Diese Missverständnisse und Mehrfachdeutungen treten regelmäßig bei der Anforderungsermittlung der klassischen Softwareentwicklung auf, da zum einen generell bei der Kommunikation Unklarheiten auftreten können. Zum anderen sprechen die Softwareentwickler und Fachexperten in der Regel eine andere Sprache, da ein komplett unterschiedlicher fachlicher Hintergrund vorliegt. Durch die geführten Experteninterviews ergibt sich eine weitere Lösung für den bereits angesprochenen Fachkräftemangel. Da bei Low Code Plattformen die Fachlichkeit von der Technik getrennt ist, könnten alle technischen Aspekte der Plattform zentral durch den Bund in engem Austausch mit dem Hersteller verwaltet werden. Somit ist nicht mehr jedes Bundesland und jede Kommune für die technische Wartung der eigenen Anwendungen zuständig, wodurch weniger Personal benötigt wird. Ein weiterer positiver Aspekt ist die direkte Partizipation der Fachexperten in dem Softwareentwicklungsprozess, da diese mithilfe von Low Code eigenständig konfigurieren können und somit nicht mehr von den Entwicklern abhängig sind. Dadurch kann die Akzeptanz für die Digitalisierung gesteigert werden, da die Mitarbeiter aus den Fachabteilungen stärker in den Entwicklungsprozess eingebunden sind. P11 sieht als wichtige Voraussetzung für die Akzeptanz, die kritischen Mitarbeiter fachlich abzuholen und ihre persönlichen Vorteile bei der Nutzung deutlich herauszustellen.

Ein weiterer Vorteil von Low Code Plattformen, der häufig in der Literatur genannt wird, ist die schnelle Entwicklung von Software, wodurch die langen und kostenintensiven Entwicklungszyklen für Software verkürzt werden (siehe Kapitel 2.3.6 Vorteile). Dieser Aspekt der Low Code Plattformen wird von den Experten mehrmals aufgegriffen und erklärt sich durch die Tatsache, dass, aufgrund des modularen Ansatzes, bei der Softwareentwicklung mit Low Code Plattformen nicht bei jedem Projekt von null begonnen werden muss. P14 nennt als Beispiel hierfür, dass die Programmierschnittstellen

(APIs) der Low Code Plattform gesamtheitlich von Experten überprüft werden. Darunter fällt eine Qualitätssicherung, ein Security Review und eine Barrierefreiheitsprüfung. Bei der klassischen Programmierung werden die Aufwände zur Prüfung der Schnittstellen bei jedem Softwareprojekt benötigt, um den gleichen Standard zu gewährleisten. Für P7 und P15 hat die schnelle Entwicklung von Software zur Folge, dass aufgrund von Low Code, die Bereitstellung von ersten Prototypen nicht mehr einige Wochen, sondern wenige Stunden dauert. Durch die Wiederverwendbarkeit von Modulen kann die Entwicklung von Software in rasanter Geschwindigkeit entstehen und eine echte Standardisierung ermöglicht werden, wenn die Module aus den unterschiedlichsten Projekten nach den gleichen Regeln entwickelt werden. Somit werden im Laufe der Zeit die standardmäßigen Module der Low Code Plattform durch eigens erstellte Module erweitert, wodurch eine Art Schneeballsystem mit immer mehr verwendbaren Modulen entsteht. Zu Beginn jedes Projekts ist es laut P7 hilfreich zu prüfen, welche passende Module bereits vorhanden sind. Diese gilt es dann, abhängig von den eigenen Anforderungen, anzupassen. Somit können Module wiederverwendet werden, was den Konfigurationsaufwand reduziert, wenn an anderer Stelle dieselben oder ähnliche Anforderungen, wie z.B. die Definition aller Eigenschaften einer Textbox, bereits umgesetzt sind. Dadurch entfällt die häufig langwierige Suche nach passenden Bausteinen in Bibliotheken oder Internetplattformen wie z.B. Stack Overflow. P3 erwähnt im Interview den Vorteil, dass Grundthemen wie Responsive Design oder Barrierefreiheit in vorgefertigten Modulen vorgedacht werden können, sodass jede entwickelte Software die gleichen Grundanforderungen erfüllt. Durch die Wiederverwendbarkeit der Module ist die entwickelte Software gleich aufgebaut und besitzt das gleiche Benutzeroberflächenkonzept, was die Nutzbarkeit steigert. Speziell in der öffentlichen Verwaltung wird die schnelle und somit kostengünstige Softwareentwicklung dringend benötigt. Gerade im Hinblick auf den aktuellen Stand der OZG-Umsetzung und die kommenden Herausforderungen der digitalen Transformation, ist die Verwendung von Low Code unumgänglich. Ein Grund für den aktuellen Stand der Digitalisierung in Deutschland ist der Föderalismus (siehe Kapitel 2.1.4 Strukturelle Herausforderungen), wodurch bisher Kommunen und Bundesländer nach ihren individuellen Anforderungen und Regularien eigene Anwendungen entwickeln. P10 erwähnt, dass durch den modularen Ansatz das „Einer für Alle“-Prinzip sehr gut umgesetzt werden kann, indem bspw. eine Kommune die Software zur Digitalisierung der Verwaltungsleistung Hundesteuer entwickeln und den anderen Kommunen in Deutschland zur Verfügung stellt. Diese können dann die vorhandene Leistung nach ihren individuellen Bedürfnissen anpassen. Ein weiteres

Beispiel wäre ein Modul für die Adresseingabe, die bei fast allen Verwaltungsleistungen benötigt wird. Ist dies einmal erstellt, können die Citizen Developer darauf zugreifen und es ggf. anpassen. Außerdem erwähnt P10, dass durch die schnelle und damit kostengünstige Softwareentwicklung mit Low Code Plattformen selten genutzte Verwaltungsleistungen digitalisiert werden können.

In der Literatur wird die schnelle Präsentation erster Prototypen der Software als weiterer Vorteil erwähnt. Dadurch kann das Kundenerlebnis verbessert werden, indem direkt Nutzer-Feedback eingeholt werden kann (siehe Kapitel 2.3.6 Vorteile). In den Experteninterviews ist dieser Aspekt ebenfalls zur Sprache gekommen. Laut P3 ist es durch den modularen Ansatz der Low Code Plattformen möglich, ab dem ersten Tag Funktionalitäten für die Software zu entwickeln. Es entsteht direkt zu Beginn des Projektes, gegenüber dem Endanwender in der Behörde, ein positives, fortschrittliches Erscheinungsbild, da nach kurzer Zeit, mithilfe der vorgefertigten Module, ein vorzeigbares, erstes Konzept präsentiert werden kann. Dadurch können die Citizen Developer iterativ die Software konfigurieren und der gesamte Entwicklungsprozess kann beschleunigt werden.

Der sinkende Auslastungsgrad der IT-Fachabteilung durch den Einsatz von Low Code Plattformen, wird in der Literatur ebenfalls als Vorteil erwähnt. Die Theorie besagt, dass die klassischen Softwareentwickler sich dadurch weniger auf den Syntax des Codes konzentrieren müssen und sich mehr auf ihre Kernkompetenzen fokussieren können wie bspw. die Software-Architektur. Die Experten stimmen dieser Aussage nur bedingt zu, da zum einen P2 kein Fall bekannt ist, in der die Softwareentwickler in die Fachlichkeit nicht involviert sind und sich ausschließlich um die technischen Aspekte kümmern. Jedoch wird die Expertise des Softwareentwicklers nicht mehr in gleichem Maße bei der Entwicklung benötigt. Grund dafür ist, dass die Konfiguration neuer Software mit Low Code Plattformen zum jetzigen Zeitpunkt in der Praxis häufig nicht so intuitiv ist wie von den Herstellern behauptet. Daher gibt es nur wenige Ausnahmen, in denen der Fachexperte ohne die Hilfe von klassischen Softwareentwicklern Anwendungen vollkommen eigenständig erstellen kann. Zum anderen entwickeln sich die Plattformen immer weiter und vereinfachen die Konfiguration von Software. Laut P1 und P3 wird der Auslastungsgrad der IT-Fachabteilungen voraussichtlich nicht sinken, da die meisten IT-Fachabteilungen, vor allem in den Behörden, unterbesetzt sind und es neben der Entwicklung von Software viele weitere Tätigkeitsfelder, wie bspw. die IT-Sicherheit, gibt. Jedoch sinkt die absolute Zeit, die ein klassischer Softwareentwickler aus der Fachabteilung für Entwicklung einer Software aufbringen muss. Da, wie bereits erwähnt, die klassischen Softwareentwickler

durch Themen wie IT-Sicherheit weiterhin ausgelastet sein werden und in der öffentlichen Verwaltung regelmäßig kleine Anpassungen bspw. aufgrund von Gesetzesänderungen notwendig sind, ist der Einsatz von Low Code Plattformen in der öffentlichen Verwaltung besonders sinnvoll. P1 erwähnt als Beispiel hierfür Gesetzesänderungen im Steuerbereich. Eine Änderung kann dazu führen, dass ein weiteres Eingabefeld in einem Formular notwendig ist oder ein neuer Button benötigt wird. Dies kann direkt, ohne die Hilfe eines klassischen Softwareentwicklers, von den Fachexperten der Behörde umgesetzt werden.

5.1.2 A12

P3 und P9 erwähnen beide in ihrem Gespräch, dass A12 out of the box eine Grundarchitektur mit lauffähigen Funktionalitäten anbietet, wodurch bei der Entwicklung nicht komplett von null begonnen werden muss. Somit können erste Templates schon während des Ausschreibungsprozesses dem Auftraggeber präsentiert werden, damit dieser einen ersten Eindruck davon erhält, was möglich ist bzw. was umgesetzt werden kann. P3 zählt drei verschiedene Arten von A12-Funktionalitäten auf. Mit Designpattern können Benutzeroberflächen gemanagt und wiederverwendet werden. Softwaremodule wie bspw. ein Authentifizierungsmechanismus sind ebenfalls Teil der Funktionalitäten. Des Weiteren stehen dem Anwender Widgets (vorgefertigte Elemente wie Listen, die man nicht separat entwickeln muss) zur Verfügung. P11 kann die Aussage von P3 und P9 bestätigen, da in seinem eigenen Projekt die Entwicklungszeit durch den Einsatz von A12 maßgeblich verkürzt werden konnte.

Für P5 ist ein unternehmensübergreifender Erfahrungsaustausch zwischen den gemeinsamen A12-Projekten notwendig, um das Potenzial und die Vorteile von A12 bestmöglich zu nutzen. Passiert dies nicht, werden in den isolierten Projekten Ideen und Lösungen entwickelt, von denen die restlichen Projekte nicht erfahren, obwohl diese eventuell vor den gleichen Herausforderungen stehen. Durch den Austausch können Lösungen für Probleme bereitgestellt werden, die allen Nutzern zugutekommen, indem diese in A12 zurückfließen. Durch diesen Erfahrungsaustausch wächst im Laufe der Zeit die Anzahl der verfügbaren Module, wodurch immer mehr Anforderungen der Citizen Developer bereits vorhanden sind und nicht mehr entwickelt werden müssen. Dabei ist, laut P9, allerdings bei der Entwicklung im Vorhinein zu prüfen, welche benötigten Funktionen auch für weitere Projekte relevant sein können, um diese nach der Fertigstellung in das Produkt A12 mit aufzunehmen. P5 erwähnt, dass die vielfältigen verfügbaren Module von A12, die finanzielle Wettbewerbsfähigkeit verbessern, da je nach

Projekt ein Teil der Anforderungen bereits von Beginn an umgesetzt ist und dadurch weniger Restaufwand benötigt wird. Außerdem ist es bei A12 problemlos möglich, verschiedene Module miteinander zu kombinieren und weiterzuentwickeln. Laut P10 ist A12, im Vergleich zu anderen Low Code Plattformen, nicht monolithisch aufgebaut und ermöglicht es dem Kunden den Quellcode einzusehen bzw. anzupassen.

P6 sieht einen Schwerpunkt von A12 in der Konfiguration von Formularen. Daher gibt es eine Vielzahl von Formular-Modellen als Vorlage in A12, die individuell angepasst werden können. A12 wird hauptsächlich in den Bereichen Steuern und Finanzen eingesetzt, wo es viele Gesetzte zu beachten gilt und somit eine hohe fachliche Expertise für die Softwareentwicklung notwendig ist. Aufgrund der Trennung zwischen der Fachlichkeit und der Technik können sich die Softwareentwickler laut P1 auf ihre Kernkompetenzen konzentrieren und müssen sich nicht mehr mit Steuerthematiken auseinandersetzen. Gleichzeitig sind die fachlichen Experten aus der Steuer- oder Finanzbehörde bei der Formularkonfiguration nicht mehr auf die klassischen Softwareentwickler angewiesen und können ihr Fachwissen direkt umsetzen.

5.2 Einsatz von Low Code

5.2.1 Voraussetzungen für den erfolgreichen Einsatz von Low Code Plattformen

Grundsätzlich gilt es, laut P5, bei der Entwicklung von neuer Software zu prüfen, ob die Verwendung von Low Code Plattformen sinnvoll ist. Dies hängt davon ab, wie individuell die Anforderungen an die Software sind bzw. wie stark sich diese Anforderungen von bisherigen Softwareentwicklungsprojekten unterscheiden. Je stärker sich die Anforderungen der aktuellen Software mit denen der bisherigen Software gleichen, desto eher lohnt sich der Einsatz von Low Code Plattformen. Wenn sich die Anforderungen stark unterscheiden, kann es durchaus sinnvoll sein, die Software ohne Low Code Plattform von null an zu entwickeln, da der gesamte Entwicklungsaufwand geringer sein kann. Dies liegt daran, dass der individuell programmierte Code bei der Verwendung einer Low Code Plattform in das Gesamtkonstrukt passen muss, was mit einem höheren Aufwand verbunden ist. Für P3 gilt es, den Aufwand und somit auch die Kosten der Softwareentwicklung so gering wie möglich zu halten, indem es bei der Anforderungsermittlung darum geht, den Auftraggeber für die Out-of-the-box-Lösung der Low Code Plattform zu sensibilisieren. Dem Kunden

muss klar sein, welchen Mehraufwand weitere gewünschte Funktionen mit sich bringen und wie viel diese kosten.

P8 sieht für den erfolgreichen Einsatz von Low Code die Wahl der richtigen Plattform als wichtige Voraussetzung. Dies ist eine bedeutende strategische Entscheidung, da der Zeithorizont für eine Low Code Plattform ca. vier bis zehn Jahre beträgt. Hierfür ist es notwendig, dass die Entscheider die eigenen unternehmensübergreifenden Anforderungen und die Vor- und Nachteile der einzelnen Low Code Plattformen sehr genau kennen. Unterschiede zwischen den verschiedenen Low Code Plattformen gibt es, laut P14, in Bezug auf die Usability und die Konfigurationsmöglichkeiten. Die Usability ist anwenderfreundlich, wenn in kurzer Zeit und ohne lange Einarbeitung erste Anwendungen erstellt werden können. Dies geht allerdings in der Regel auf Kosten der Gestaltungsoptionen. Bietet eine Low Code Plattform dem Nutzer umfangreiche Konfigurationsmöglichkeiten, geht dies mit einer längeren Einarbeitungszeit und einer schlechteren Usability einher. Daher gilt es schon bei der Entscheidung für eine Low Code Plattform zu prüfen, wie individuell und anspruchsvoll die eigenen Anforderungen sind. P7 kommt zu dem Entschluss, dass je stärker die Vorteile der Plattform mit den eigenen Anforderungen übereinstimmen, desto effizienter kann Software entwickelt werden. Bei der Einarbeitung der Nutzer in eine neue Low Code Plattform, sollte bezüglich der Dauer auf die IT-Vorkenntnisse geachtet werden. Obwohl kein IT-Wissen für die Bedienung der meisten Low Code Plattformen notwendig ist, benötigen Mitarbeiter mit IT-Background, laut P12, eine kürzere Einarbeitungszeit und erhalten ein schnelleres Verständnis von der Funktionsweise der Plattform, da die Bedienung der Plattformen häufig nicht so intuitiv ist wie vom Hersteller angegeben. Zu dieser Einschätzung kommt ebenfalls P15, der für einen erfolgreichen Einsatz einer Low Code Plattform ein Grundverständnis der Projektbeteiligten für Datenbanken und Software als wichtige Voraussetzung sieht. P8 erwähnt außerdem, dass durch den Einsatz von Low Code zwar die Entwicklungszeit von Software im Idealfall deutlich reduziert werden kann, jedoch sollte den Beteiligten und Vorgesetzten trotzdem klar sein, wie viel Zeit für die Umsetzung benötigt wird und dass die Umsetzung nicht nebenher geschieht. Zusätzlich ist es wichtig, dass der Projektleiter die Funktionen der Low Code Plattform gut kennt und ungefähr den gesamten Arbeitsaufwand der Softwareentwicklung prognostizieren kann, um ein passendes Projektteam zusammenzustellen. Die bei einer Low Code Plattform standardmäßig zur Verfügung stehenden Templates sollten, laut P10, für eine erfolgreiche Softwareumsetzung ansprechend gestaltet und leicht verständlich sein, damit sich der Nutzer bei der entwickelnden Software leicht orientieren kann, indem alles selbsterklärend bzw. barrierefrei ist. Für

P8 und P15 ist es wie so häufig auch bei der Softwareentwicklung mit Low Code wichtig, dass einige der Projektbeteiligten hochmotiviert sind. Ist dies nicht der Fall, scheitern selbst die kleinsten Projekte.

5.2.2 Vorrausetzung für den erfolgreichen Einsatz von A12

Für P6 ist grundsätzlich von Beginn an wichtig zu prüfen, ob es sinnvoll ist, das Projekt mit A12 umzusetzen oder nicht. Dabei ist entscheidend, wie viele Projektanforderungen mit den standardmäßigen Funktionalitäten von A12 umgesetzt werden können. Wenn z.B. 70% der Anforderungen direkt umgesetzt werden können, dann ist die Verwendung von A12 sinnvoll. Sind es weniger als 50% dann ist eine klassische Softwareentwicklung naheliegender. Wird sich für den Einsatz von A12 entschieden, muss, laut P1 und P3, der Projektleiter ein passendes Projektteam zusammenstellen. Hierfür ist es wichtig A12 und die Projektanforderungen zu kennen, um zu entscheiden, wie viel konfiguriert werden kann und wie viel individuell entwickelt werden muss. Aus dieser Einschätzung ergibt sich die Anzahl an Customer's Experts, Business Analysten und Softwareentwickler. Für den Projektleiter gilt es, den Arbeitsaufwand der individuell zu entwickelnden Kundenanforderungen richtig einzuschätzen und rechtzeitig an die A12-Entwickler weiterzuleiten, damit ihnen ausreichend Zeit für die Umsetzung zur Verfügung steht. Dabei ist es wichtig, die Anforderungen nicht ohne Absprache mit den A12-Entwicklern im Projekt intern umzusetzen, damit eine zentrale Wartung der Funktionalität gewährleistet werden kann. Des Weiteren müssen klare Regeln festgelegt werden, an wen sich die Business Analysten bei welcher Art von Problemen wenden können, damit Front- und Backendprobleme so schnell und effizient wie möglich gelöst werden können. P6 sieht geschulte Business Analysten ebenfalls als eine wichtige Voraussetzung für den erfolgreichen Einsatz von A12, da es dadurch zu weniger Nachfragen kommt. Zusätzlich ist bei der Arbeit der Business Analysten ein gewisser Perfektionismus vorteilhaft, damit am Ende die Anwendung strukturiert und nachvollziehbar aussieht. Für P4 spielen die Anforderungen an das Projekt in Bezug auf Benutzerfreundlichkeit eine wichtige Rolle, um entscheiden zu können, wie gut sich die Business Analysten mit dem Designen von Benutzeroberflächen auskennen müssen. Eine weitere Voraussetzung sind regelmäßige A12-Updates, damit aktuelle Wünsche der Nutzer umgesetzt und Fehler behoben werden. Damit die Business Analysten schnell mit A12 arbeiten können, ist, laut P6, eine A12-Expertengruppe für jeden Bereich hilfreich, an den man sich bei speziellen Fragen wendet. Dadurch kann ebenfalls schnell abgeklärt werden, ob Funktionen standardmäßig umgesetzt werden können oder nicht.

5.2.3 Voraussetzungen für die Nutzung von A12

Die interviewten Experten, die mit A12 arbeiten, sind sich alle einig, dass weder ein IT-Studium noch IT-Vorkenntnisse für die Nutzung von A12 notwendig sind. Allerdings arbeiten bei mgm viele Quereinsteiger mit A12, die ein naturwissenschaftlichen oder wirtschaftlichen Hintergrund besitzen. Für P4 und P11 sind die Fähigkeiten strukturiert zu denken und offen für neues zu sein hilfreich, um sich schnell in A12 zurechtzufinden. P1 nennt als Beispiel, dass ein Mitarbeiter aus seinem Projektteam mit einer 50%-Stelle in ca. einem halben Jahr eigenständig in A12 Anforderungen umsetzen kann.

5.3 Expertenmeinung zu Low Code und welche Probleme es löst

5.3.1 Software bzw. Anwendungen, bei denen Low Code Plattformen besonders geeignet sind

Insbesondere eignet sich, laut P2, die Verwendung einer Low Code Plattform dann, wenn damit entwickelte Systeme häufig angepasst werden müssen. Grund dafür ist, dass Softwareanpassungen verhältnismäßig leicht durchführbar sind und normalerweise keine klassischen Softwareentwickler benötigt werden. P14 ist der Meinung, dass Low Code Plattformen gerade dann geeignet sind, wenn sie in dem Bereich eingesetzt werden, wofür sie gedacht sind, und dies sind in der Regel Fachanwendungen. Für P8 werden Low Code Plattformen für die Realisierung von Webanwendungen, mobile Anwendungen, Dashboards und Websites der Standard werden, da diese sehr gut modular entwickelt werden können. Im Projektalltag sind, laut P5, Low Code Plattformen bei der Softwareentwicklung gerade dann geeignet, wenn zum Projektstart noch keine funktionsfähige Anwendung auf Kundenseite vorhanden ist, es sich also um eine Neuentwicklung handelt. Die Weiterentwicklung bzw. Anpassung einer umfangreichen funktionsfähigen Software durch Einbindung eine Low Code Plattform kann aufgrund der zu etablierenden Schnittstellen sehr aufwendig werden.

5.3.2 Herausforderungen bei der Softwareentwicklung mit Low Code Plattformen in der öffentlichen Verwaltung

Als Herausforderung in der öffentlichen Verwaltung sieht P7 die statische klassische Vorgehensweise der Softwareentwicklung, bei der zu Projektbeginn der Entwicklungsprozess genau durchgeplant wird, bevor man mit der Umsetzung beginnt. Die Einführung der Low Code Plattform ist eine Möglichkeit, das Vorgehensmodell der Softwareentwicklung langfristig zu verändern, indem agiler gearbeitet wird und nicht alles perfekt

beschrieben, oder geplant sein muss, bevor der nächste Entwicklungsschritt beginnt. Zusätzlich ist für P7 der Aufbau von Vertrauen in der öffentlichen Verwaltung, zwischen Behörde und Dienstleister eine Herausforderung, die für eine erfolgreiche Umsetzung von Softwareprojekten entscheidend ist. Je größer das Vertrauen, desto größer ist, laut P7, der interne Austausch im Projekt, wodurch Geschwindigkeit gewonnen und Kosten gesenkt werden können. Aufgrund der schnellen Bereitstellung von visuellen und erlebbaren Anwendungen beim Einsatz von Low Code, entsteht Vertrauen, da anschauliche Lösungsansätze präsentiert und mit den Nutzern diskutiert werden können. Im Idealfall werden Nutzer in die Anwendungsgestaltung direkt involviert. Für P8 ist eine der Herausforderungen in der öffentlichen Verwaltung, den Begriff „Low Code Plattform“ zu konsolidieren. Da für viele nicht genau klar ist, was sich hinter dem Begriff verbirgt, ist es für Verantwortliche schwer zu entscheiden, in welchen Szenarien der Einsatz von Low Code sinnvoll ist. P14 sieht bei der Softwareentwicklung in der öffentlichen Verwaltung keine Herausforderung, die zu komplex ist, um sie mit einer Low Code Plattform umzusetzen. Als Beispiel hierfür nennt er die Komplexität der Steuerlogik, die mithilfe der Low Code Plattform A12 modular umgesetzt wird.

5.3.3 Optimierung der IT-Sicherheit durch Reduzierung der Schatten-IT

Die Schatten-IT lässt sich, laut P7, nur vollständig verhindern, wenn alle Mitarbeiter nur mit zentral bereitgestellten Anwendungen arbeiten. Da dies nicht realistisch ist, wird es immer Hilfslösungen mit Excel, Access oder anderen Tools geben. Jedoch sind P2, P5 und P7 überzeugt, dass Low Code Plattformen grundsätzlich die Schatten-IT reduzieren kann. Hierfür muss es den Anwendern so leicht wie möglich gemacht werden, eigene Modelle für Anwendungen zu erstellen, wie z.B. ein Formular zu gestalten oder anzupassen. Zusätzlich ist, laut P8 und P10, für jedes Softwareprojekt eine ganzheitliche Betrachtung der Projektanforderungen und gesetzlichen Regularien notwendig. Denn Schatten-IT entsteht, wenn Mitarbeiter eine Aufgabe übertragen bekommen, die es, egal wie, zu erledigen gilt. Denn dann greifen die Angestellten auf die bereits vorhandenen Excel- oder Access-Dateien zurück. Um dies zu verhindern, muss sich das Unternehmen um die Belange der Mitarbeiter kümmern und nachfragen, was genau benötigt wird, um die Aufgabe zu erledigen und das Ergebnis in die interne Infrastruktur zu integrieren. Für P7 muss die zentrale IT bei der Nutzung einer Low Code Plattform akzeptieren, dass Anwendungen auch aus Fachbereichen heraus entwickelt werden können. Allerdings müssen dabei bestimmte Regeln strikt befolgt werden. Der Fachbereich muss akzeptieren,

dass die zentrale IT die Verantwortung für die meisten Anwendungen und auch für das Framework übernimmt. Regeln der IT-Abteilung sind erforderlich, die von Fachabteilungen verpflichtend eingehalten werden. Die Schwierigkeit liegt darin, die Einschränkungen und Freiheiten abzuwägen, um zum einen die Kreativität der Mitarbeitenden möglichst stark zu nutzen und ihnen zum anderen ein sicheres Umfeld für solche Innovationen zu gewährleisten. Dies lässt sich, laut P7, mit einem Citizen Developer Framework umsetzen. Hierfür sollte im Unternehmen ein Exzellenzzentrum für Low Code etabliert werden, das Best Practice, Namenskonventionen[10] und Sicherheitsregeln für die Entwicklung mit Low Code Plattformen festlegt und in engem Austausch mit der zentralen IT steht. Zusätzlich ist es wichtig im Exzellenzzentrum Know-how aufzubauen, um entscheiden zu können, welche Anwendungen in den Fachabteilungen verwaltet werden, wie z.B. kleine Anwendungen, die keine kritischen Prozesse abbilden. Außerdem ist es strategisch wichtig, welche Anwendungen sie selbst im Exzellenzzentrum betreuen, wie z.B. mittelgroße Anwendungen ohne kritische Anforderungen und welche Anwendungen sie an die zentrale IT übergeben. Damit entsteht eine zentrale Kontrolle durch das Exzellenzzentrum und eine flexible dynamische Anwendungsgestaltung durch die Nutzer. Diese Transparenz ist, laut P7, ein wichtiger Erfolgsfaktor, um die Schatten-IT bestmöglich zu reduzieren. Da die Anwendungsfälle so vielfältig sind, muss die Einführung eines solchen Frameworks stufenweise und zielgerichtet erfolgen. Für P14 muss für die Optimierung der IT-Sicherheit jede Software des Unternehmens, die die Fachlichkeit oder Fachanwendungen abbildet, in eine Anwendungslandschaft integriert werden. Es entsteht kein großer Vorteil, wenn ein Excel-Sheet durch eine in sich geschlossene Low Code Anwendung abgelöst wird, da die Datenintegration fehlt und somit weiterhin eine Insellösung besteht. Der Datenaustausch ist notwendig, damit die Daten aus der Anwendung nicht in eine Excel-Datei exportiert und an anderer Stelle wieder importiert werden. Laut P14 ist wichtig, dass die Low Code Plattform von vornherein dafür geschaffen ist, die einzelnen entwickelten Anwendungen in die Low Code Plattform zu integrieren. Damit die Vorteile der Low Code Plattform genutzt werden können, muss eine Anwendungslandschaft mit automatischem Datenaustausch das Ziel sein. Als Beispiel zur Reduzierung der Schatten-IT nennt P5 ein aktuelles Projekt, indem es dem Citizen Developer ermöglicht wird, mithilfe eines Editors, Seiten und Formulare intuitiv anzupassen, indem der Anwender visuell die

[10] Namenskonventionen sind eine freiwillige Vereinbarung zwischen bspw. Datenbankentwicklern und Programmierern, wie zum Beispiel Variablen, Objekte oder Konstanten zu definieren sind (datenbanken-verstehen).

selbst durchgeführten Änderungen sieht. Dadurch muss er nicht umdenken und sich vorstellen, welche Auswirkungen die eigenen Änderungen besitzen. Anforderungen, die in Form von Excel-Listen und Tickets an die klassischen Entwickler weitergeleitet werden, fallen dadurch weg.

5.3.4 Nachteile von Low Code Plattformen in der Softwareentwicklung

Neben dem bereits in Kapitel 2.3.7 Nachteile erwähnten Lock-in-Effekts sieht P12 bei der Nutzung einer Low Code Plattform die Prüfung der eigenen Sicherheitsstandards als Problem. Man vertraut dem Hersteller, dass die Plattform gegen Cyberattacken geschützt ist und regelmäßig Sicherheitspatches veröffentlicht werden, wodurch eine Abhängigkeit entsteht. Außerdem ist es nur eingeschränkt möglich, die Ursache für Fehler zu finden und diese zu beheben. Die Kosten sind ebenfalls ein Aspekt, der, laut P12 und P15, bei der Verwendung von Low Code Plattformen interessant ist. Bei den meisten Plattformen ist die Softwareinitiierung mit Low Code günstiger, da eine schnellere Umsetzung möglich ist. Allerdings können, je nach Lizenzmodell, die laufenden Kosten der Software höher sein als bei der klassischen Entwicklung. Für P2 ist die tatsächliche Einarbeitungszeit für die Nutzung einer Low Code Plattform in der Regel länger als von den Herstellern angegeben, da meistens angenommen wird, dass bei den Nutzern ein IT-Grundverständnis vorhanden ist. Allerdings ist dies häufig nicht der Fall, da es bspw. vielen schwer fällt, den Datentyp Boolean[11] zu verstehen. Problematisch kann, laut P13, die Verwendung einer beliebigen Low Code Plattform sein, wenn zu Beginn des Softwareprojekts nicht sichergestellt wird, dass die gewählte Low Code Plattform die Anforderungen an die zu entwickelnde Software überhaupt abbilden kann. Ist eine Low Code Plattform im Einsatz, deren standardmäßigen Funktionen nicht ausreichen und zudem keine eigene Entwicklung möglich ist, bspw. wenn kein Drittsystem angebunden werden kann, dann muss im schlimmsten Fall die Softwareentwicklung neu begonnen werden. Bei einer Low Code Plattform mit der Möglichkeit der eigenen Entwicklung muss das Projekt nicht von neuem begonnen werden. Die Umsetzung der hinzugekommenen Anforderungen, die nicht konfigurativ abgebildet können, stellen somit Programmieraufwand dar. Von einem ähnlichen Problem sprechen P5 und P15, welche die komplexe Integration, Weiterentwicklung und Anpassung von bereits funktionsfähigen umfangreichen Kernprozessen des Kunden in die Low Code Plattform als herausfordernd ansehen. Problematisch wird es gerade dann, wenn

[11] In der Informatik bedeutet der Begriff Boolean ein Ergebnis, das nur einen von zwei möglichen Werten annehmen kann: wahr oder falsch (Awati 2022).

Funktionen aus dem bisher laufenden System mit den standardisierten Möglichkeiten der Low Code Plattform nicht umgesetzt werden können und eine individuelle Alternative benötigt wird. Aus diesem Grund ist es sehr wichtig, schon zu Beginn klar zu kommunizieren, welche Funktionen durch den Einsatz der Low Code Plattform gar nicht oder nur mit einem hohen Programmieraufwand umgesetzt werden können. Laut P2, P7, P8 und P14 gibt es Anwendungsfelder, für die Low Code Plattformen eher ungeeignet sind, wie z.B. betriebssystemnahe Anwendungen, die eng mit dem Betriebssystem interagieren und auf dessen Funktionalität und Ressourcen direkt zugegriffen wird. Je sicherer die Software sein muss, desto höher sind auch die Anforderungen an die eingesetzte Low Code Plattform oder individuell entwickelten Lösungen. Dieses Beispiel zeigt, dass den klassischen Softwareentwicklern durch den Einsatz von Low Code die Arbeit nicht ausgehen wird und sie sich verstärkt auf große komplexe Anwendungen konzentrieren werden. Laut P12 bestehen einige Unternehmen und Behörden darauf, Software wie z.B. eine Low Code Plattform im eigenen Unternehmen zu installieren und zu betreiben. Allerdings bieten nicht alle Low Code Anbieter eine solche Lösung für ihr Produkt an, was die Auswahl der passenden Plattform einschränkt.

5.3.5 Probleme, die A12 bei der Softwareentwicklung in der öffentlichen Verwaltung löst

Ein großes Problem bei der Softwareentwicklung in der öffentlichen Verwaltung ist der hohe Aufwand für die Entwicklung und Wartung von Anwendungen, die komplexe Thematiken wie z.B. Steuern betreffen. Die meisten Steuerarten, wie z.B. die Investmentsteuer sind, laut P9, sehr komplex und für Personen, die sich mit Steuern weniger gut auskennen, sehr schwer zu verstehen. Gleichzeitig führen politische Maßnahmen zur regelmäßigen Erlassung neuer Gesetze, die sich auf die Anwendung auswirken können und ggf. Anpassungen erfordern. Bisher mussten die Fachexperten in den Behörden aus den Steuergesetzen passende Anforderungen an die Software so formulieren, dass die Softwareentwickler diese verstehen und umsetzen können. P5 erwähnt in seinem Gespräch, dass es bei der Anforderungsermittlung zu Missverständnissen kommt, da die Fachexperten und klassischen Entwickler einen komplett unterschiedlichen fachlichen Hintergrund besitzen. Dies führt zu einer längeren und somit teureren Entwicklung. A12 ermöglicht es, die Fachlichkeit von der Technik zu trennen. Dies bedeutet, dass die Softwareentwickler sich auf ihre Kernkompetenz konzentrieren können, indem sie den technischen Rahmen schaffen, um den Fachexperten die Konfiguration von bspw.

Steuerformularen zu ermöglichen. Zum einen können Missverständnisse, laut P5, vermieden werden, da kein fachlicher Austausch zwischen Fachexperten und Entwicklern notwendig ist. Zum anderen können Anpassungen, die z.B. durch neue Gesetzte erforderlich sind, direkt von dem Fachexperten durchgeführt werden. Der Fachkräftemangel in der öffentlichen Verwaltung kann durch den Einsatz von A12 angegangen werden, da weniger klassische Entwickler für die Softwareentwicklung benötigt werden. Zukünftig wird sich, laut P1 und P4, die Benutzerfreundlichkeit der mit A12 entwickelten Anwendungen weiter verbessern, da kontinuierlich das Designkonzept und bspw. auch Anforderungen an die Barrierefreiheit optimiert wird.

5.3.6 Herausforderungen bei der Softwareentwicklung mit A12

P1 und P10 sehen es als Herausforderung bei A12 an, zu Projektbeginn den Gesamtaufwand der zu entwickelnden Software einzuschätzen. Dabei stellt sich die Frage, welche Softwareanforderungen mit den standardmäßigen Funktionalitäten umgesetzt werden können und welche individuell programmiert werden müssen. Anforderungen, bei denen erst bei der Umsetzung auffällt, dass eine individuelle Programmierung notwendig ist, sorgen für eine Verlängerung der Projektdauer und somit zu einer Erhöhung der Kosten. Für P14 ist eine weitere Herausforderung, die Möglichkeit zum Persistieren[12] von Daten zu verbessern. Die aktuellen Möglichkeiten von A12 reichen nicht für jede Anforderungen der fachlichen Welt aus. Somit muss in der Regel individuell entwickelt werden und diese Entwicklung ist auch früh im Projekt erforderlich. Zusätzlich gilt es in Zukunft die Integration mit Umsystemen[13] modulbasiert umzusetzen, um ebenfalls den Anteil an individueller Programmierung zu reduzieren.

5.3.7 Verbesserung der IT-Sicherheit durch Verwendung von A12

Für P3 kann das Potenzial zur Verbesserung der IT-Sicherheit, mithilfe von A12, dann ausgeschöpft werden, wenn das gesamte Softwareprojekt mit A12 umgesetzt wird und nicht bspw. zwei Unternehmen separat mit unterschiedlichen Programmen das Front- und Backend entwickeln. Dadurch sind im Prozess keine manuellen Schritte notwendig, da ein Datenaustausch ermöglicht wird. Eine weitere Verbesserung der IT-Sicherheit ist, dass die Nutzung und Wartung von A12-Anwendungen, laut P9, leichter

[12] Persistenz ist die Fähigkeit, Objekte oder Daten auf nichtflüchtigen Medien speichern zu können (Siepermann 2018).

[13] Umsystem ist ein Bereich, der außerhalb der Systemgrenzen liegt, zu dem das System aber Beziehungen aufweisen kann (ibo Akademie).

von Personen durchgeführt werden kann, welche die Anwendung nicht selbst entwickelt haben, da sich im Vergleich zur klassischen Entwicklung der Aufbau und die Struktur aller A12-Anwendungen aufgrund des modularen Ansatzes ähneln. Somit kann die Anzahl an Hilfstools, die eine Sicherheitslücke darstellen, reduziert werden. P10 bestätigt die Einschätzung von P3 und P9, indem er von seiner täglichen Arbeit mit A12 berichtet und der Meinung ist, dass A12 teilweise die IT-Sicherheit verbessert. Vor dem Einsatz von A12 kam es bei der Formularerstellung zu Medienbrüchen, indem erstellte Formulare händisch abgeschrieben wurden und somit eine Gefahr für den Datenschutz und die IT-Sicherheit darstellten. Allerdings sind das Frontend und die internen Fachverfahren weiterhin voneinander getrennt, was den Datenaustausch erschwert und somit eine weitere Verbesserung der IT-Sicherheit verhindert. Diese Trennung gilt es laut ihm nach dem Vorbild der OZG-Plattform aufzulösen. P1 und P6 sind beide der Meinung, dass A12 seinen Beitrag dazu leisten kann, die IT-Sicherheit zu verbessern. Jedoch wird laut ihnen die Schatten-IT weiterbestehen, da zum einen der Scope der A12-Projekte zu groß ist, um alle Funktionen damit abzudecken. Zum anderen werden die Fachexperten trotz A12 für bestimmte Anforderungen Excel benutzen, da es eine schnellere Umsetzung ermöglichen kann und gleichzeitig auch nicht alle Fachexperten in der Behörde A12 bedienen können.

5.3.8 Nachteile von A12

Wie bereits erwähnt, ist A12 für die Entwicklung von komplexen Anwendungen geeignet, die in eine vielschichtige IT-Infrastruktur integriert werden muss. Um dies zu ermöglichen sind umfangreiche Konfigurationsmöglichkeiten notwendig, die eine ausführliche Einarbeitung voraussetzt. Dadurch dauert, laut P13, die Entwicklung von kleinen Tools bzw. Anwendungen mit A12 länger als bei Konkurrenzprodukten und länger als bei der klassischen Softwareentwicklung. Somit ist A12 eher ungeeignet für die Entwicklung einer einfachen App oder eines in sich geschlossenen Systems und dadurch eine zu große Lösung für ein kleines Problem. Ähnlich wie P13 sieht es auch P10 in seinem aktuellen Projekt, da die Komplexität bzw. die Möglichkeiten von A12 vielseitiger sind als für den Anwendungsfall notwendig und für das Szenario des Projekts Einfachheit wichtig ist. Außerdem erwähnt P10, dass trotz der umfangreichen Konfigurationsmöglichkeiten von A12, die reellen Prozesse in der öffentlichen Verwaltung ohne individuelle Anpassungen nicht vollständig dargestellt werden können.

5.4 Zukunftsperspektive

5.4.1 Möglichkeiten das Einsatzgebiet von Low Code Plattformen in der öffentlichen Verwaltung zu erweitern?

Damit sich das Einsatzgebiet von Low Code Plattformen in der öffentlichen Verwaltung erweitert und noch häufiger eingesetzt werden kann, ist für P14 eine stärkere Unterstützung des Softwareentwicklungsprozesses notwendig. Dies ist dann möglich, wenn schneller erste Prototypen der Software getestet werden können und der Anwender weiß, wie seine Betriebs-, Test- und Entwicklungsumgebung aussieht. Für den breiteren Einsatz von Low Code Plattformen soll im Idealfall die Möglichkeit bestehen, die Prozesse und Umgebungen der Softwareentwicklung zu überblicken und ggf. zu optimieren. Solange dies nicht möglich ist, geschieht die Optimierung nur im Rahmen von Großprojekten und diese Projekte sind die einzige Möglichkeit, um die aktuelle Qualität der Entwicklung zu beurteilen. Durch die Optimierung der Low Code Plattform kann im Idealfall der Betriebsablauf vorgedacht werden. Als weitere Möglichkeit, das Einsatzgebiet von Low Code Plattformen zu erweitern, sieht P15 die richtige Positionierung und Vermarktung der bereits bestehenden Low Code Plattformen. Sie bieten bereits umfangreiche bzw. vielfältige Funktionen an, die allerdings den wenigsten bekannt sind.

5.4.2 Zukünftige mögliche Anwendungsgebiete für den Einsatz von Low Code Plattformen in der öffentlichen Verwaltung

Als zukünftiges, mögliches Anwendungsgebiet bietet der Bereich des Business Process Managements für Fachverfahren in der öffentlichen Verwaltung, laut P7, ein hohes Potenzial. Damit ist es möglich, mit der Hilfe von Low Code Plattformen nicht nur einen Teilbereich, sondern komplette Prozesse zu digitalisieren. Somit sind alle Aufgaben in einem Workflow sichtbar und ermöglichen, Potenziale und Engpässe in den Prozessen zu ermitteln. Die Sichtbarkeit der Prozesse ermöglicht es, auf Kapazitätsengpässe dynamisch zu reagieren. Als weiteres Anwendungsgebiet erwähnt P7 das Antrainieren einer Künstlichen Intelligenz, die per Sprachinterface Anwendungen baut. Low Code Plattformen bieten aufgrund der Standardisierung ein riesiges Potenzial, da nicht nur Nutzer Interfaces entwickelt werden können, sondern auch das komplette Backend integriert ist. Somit ist es für die Künstliche Intelligenz wesentlich leichter, neue Anwendungen zu bauen, auf bereits vorhandene Bausteine zurückzugreifen und diese dann wiederzuverwenden. Bausteine, die zusammengefügt

werden, sind für eine Künstliche Intelligenz leichter zu verarbeiten als individuelle High Code Anwendungen.

5.4.3 Weitere Einsatzmöglichkeiten mit A12 die bisher nicht angewandt werden

Für P9 wäre es wünschenswert, wenn Fachexperten die Möglichkeit besitzen würden, in A12 die Benutzeroberflächen selbst gestalten zu können. Zusätzlich kann, laut P14, ein weiteres Einsatzgebiet sein, mit A12 an die Anwendungsentwickler heranzutreten, die in erster Linie kleinere, eigenständige Anwendungen erstellen. Hierdurch kann eine mittlere Ebene zwischen Fachexperten, die konfigurieren und Enterprise-Entwicklern geschaffen werden. P6 erwähnte im Gespräch, dass grundsätzlich im E-Commerce Bereich der Einsatz von A12 ebenfalls sinnvoll sein kann, wenn die Funktionalitäten von A12 erweitert werden.

5.4.4 Wünschenswerte Verbesserungen in A12

Als wünschenswerte Weiterentwicklung von A12 nennt P13 die Möglichkeit für den Fachanwender eigenständig einfache Anwendungen zu konfigurieren, ohne auf den Entwickler angewiesen zu sein. Dadurch kann die Anwenderfreundlichkeit gesteigert werden, indem die Gesamtnutzung verbessert wird. Zusätzlich können sich Entwickler auf andere Aufgaben konzentrieren. P4 nennt als weiteren Aspekt die Vereinfachung der Konfiguration mit A12, damit noch mehr Fachexperten eigenständig Anwendungen erstellen können und weniger auf die Hilfe der mgm-Mitarbeiter angewiesen sind. Die Versionierung in A12 kann, laut P1, gerade für Personen mit weniger Programmiererfahrung hilfreich sein, da dadurch gängige Versionsverwaltungen wie Git oder Bitbucket nicht mehr genutzt werden müssen. Für P3 sind weitere Add-ons für A12 hilfreich, die so binärisch sind, dass sie in den meisten A12-Projekten zum Einsatz kommen, wie z.B. der Export als PDF oder Excel-Datei.

6 Fazit und Ausblick

Mithilfe der Literaturrecherche und den durchgeführten Experteninterviews wurde im Rahmen der wissenschaftlichen Arbeit erfolgreich eine Potenzialanalyse bei der Softwareentwicklung mit Low Code Plattformen in der öffentlichen Verwaltung durchgeführt. Die Erkenntnisse dieser Analyse werden im Folgenden übersichtlich dargestellt.

Ein Aspekt der Potenzialermittlung und ein Ziel der wissenschaftlichen Arbeit ist die Identifizierung der Vorteile, die der Einsatz von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung bietet. Low Code Plattformen haben das Potenzial, dem Fachkräftemangel in der IT entgegenzuwirken. Fachexperten in der öffentlichen Verwaltung können, mithilfe von Low Code Plattformen, eigene Software entwickeln und müssen sich nicht mit den technischen Systemen im Hintergrund auskennen. Dadurch können Missverständnisse bei der Anforderungsermittlung zwischen Softwareentwickler und Fachexperten minimiert werden. Zusätzlich bieten Low Code Plattformen die Möglichkeit, im Vergleich zur klassischen Softwareentwicklung, Software schneller zu entwickeln, wodurch lange und kostenintensive Entwicklungszyklen verkürzt werden. Low Code Plattformen bieten als weiteres Potenzial für die Softwareentwicklung in der öffentlichen Verwaltung, eine Steigerung der Benutzerfreundlichkeit der Software, da eine schnelle Präsentation erster Prototypen der Software möglich ist und direktes User-Feedback eingeholt werden kann.

Mithilfe der empirischen Untersuchung in der vorliegenden wissenschaftlichen Arbeit, konnten die erwähnten Vorteile von Low Code Plattformen, durch die praxisnahen Erfahrungsberichte mit der Low Code Plattform A12, bestätigt werden. So bietet A12 das Potenzial, dass, aufgrund des modularen Ansatzes, bei der Softwareentwicklung nicht jedes Mal von null begonnen werden muss und dem Auftraggeber bereits erste Prototypen in Form von Templates direkt präsentiert werden können, was die gesamte Entwicklungszeit maßgeblich beschleunigt. Ein weiterer Vorteil des modularen Ansatzes ist, dass sich die Struktur und der Aufbau aller A12-Anwendungen ähneln. Somit kann die Nutzung und Wartung von Anwendungen leichter von Personen durchgeführt werden, welche die Anwendungen nicht selbst entwickelt haben. Ein weiteres Potenzial von A12 ist, dass die fachlichen Experten aus den Behörden nicht mehr so stark auf die klassischen Softwareentwickler angewiesen sind. Durch die Trennung von

Fachlichkeit und Technik müssen den Softwareentwicklern bspw. keine komplizierten Steuergesetze erklärt werden.

Ein weiteres Ziel der wissenschaftlichen Arbeit besteht darin die Voraussetzungen für den erfolgreichen Einsatz von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung zu ermitteln. Low Code Plattformen können gerade dann ihr Potenzial entfalten, wenn die Softwareprojekte für solche Plattformen geeignet sind. Hierfür gilt es beim Projektbeginn zu prüfen, wie sehr sich die Anforderungen von bisherigen Softwareentwicklungsprojekten unterscheiden. Je stärker sich die Anforderungen der zu entwickelnden Software denen der bisherigen Software gleichen, desto eher lohnt sich der Einsatz einer Low Code Plattform. Eine zusätzliche Voraussetzung für die bestmögliche Potenzialentfaltung von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung, ist die Wahl der richtigen Plattform. Hierfür gilt es grundlegend zu prüfen, wie individuell und anspruchsvoll die eigenen unternehmensübergreifenden Softwareanforderungen sind, da sich eine neue Software umso effizienter entwickeln lässt, wenn eine große Überschneidung zwischen den Vorteilen der Plattform und den eigenen Anforderungen vorhanden ist. Ein weiterer Aspekt ist die Tatsache, dass wenn eine Low Code Plattform dem User umfangreiche Konfigurationsmöglichkeiten bietet, dies mit einer längeren Einarbeitungszeit und einer schlechteren Usability einhergeht. Um die Möglichkeiten der Low Code Plattformen bestmöglich zu nutzen, sollte bezüglich der Einarbeitungszeit zur Nutzung einer Low Code Plattform auch auf die IT-Vorkenntnisse der Projektbeteiligten geachtet werden. Obwohl kein IT-Wissen für die Bedienung der meisten Low Code Plattformen notwendig ist, benötigen Mitarbeiter mit IT-Background eine kürzere Einarbeitungszeit und erhalten ein schnelleres Verständnis von der Funktionsweise der Plattform. Es ist wichtig, dass der Projektleiter die Funktionen der Low Code Plattform gut kennt und ungefähr den gesamten Arbeitsaufwand der Softwareentwicklung prognostizieren kann, um ein passendes Projektteam zusammenzustellen.

Für die Potenzialanalyse der Low Code Plattform A12 bei der Softwareentwicklung in der öffentlichen Verwaltung, gilt es zu prüfen, wie viele Projektanforderungen mit den standardmäßigen Funktionalitäten von A12 umgesetzt werden können. Wenn z.B. 70% der Anforderungen direkt umgesetzt werden können, dann ist die Verwendung von A12 sinnvoll. Sind es weniger als 50% dann ist eine klassische Softwareentwicklung naheliegender. Zusätzlich ist für die bestmögliche Potenzialentfaltung beim Einsatz von A12 die Expertise des Projektleiters von entscheidender Bedeutung, um den

Arbeitsaufwand der individuell zu entwickelnden Kundenanforderungen richtig einzuschätzen und rechtzeitig an die A12-Entwickler weiterzuleiten, damit ihnen ausreichend Zeit für die Umsetzung zur Verfügung steht. Des Weiteren sind für die Potenzialsteigerung von A12 bei der Softwareentwicklung in der öffentlichen Verwaltung, geschulte und akribisch arbeitende Business Analysten eine wichtige Voraussetzung, damit am Ende die Anwendung strukturiert und nachvollziehbar aussieht.

Ein weiterer Aspekt der Potenzialanalyse ist, die Art von Software zu bestimmen, bei der sich die Nutzung von Low Code Plattformen besonders lohnt. Das Potenzial von Low Code Plattformen kann gerade dann bestmöglich genutzt werden, wenn zum Projektstart noch keine funktionsfähige Anwendung auf Kundenseite vorhanden ist oder wenn die entwickelte Software häufig angepasst werden muss, wie z.B. aufgrund gesetzlicher Regelungen. Es sei noch darauf hingewiesen, dass Low Code Plattformen gerade bei der Entwicklung von Fachanwendungen, Webanwendungen, mobile Anwendungen, Dashboards und Websites ihr volles Potenzial entfalten können. Damit das Potenzial von Low Code Plattformen optimal genutzt werden kann, ist die Identifizierung von Herausforderungen bei der Softwareentwicklung in der öffentlichen Verwaltung entscheidend, um potenzielle Probleme rechtzeitig aus dem Weg zu räumen. Ein Aspekt ist sicherlich die häufig statische Vorgehensweise in den Behörden, bei der die Prozesse von Anfang bis Ende genau geplant werden, bevor die Umsetzung beginnt. Für die bestmögliche Potenzialentfaltung von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung, ist der Vertrauensaufbau zwischen Behörde und Dienstleister entscheidend, damit der Austausch und die Umsetzungsgeschwindigkeit zunehmen. Außerdem kann das Potenzial von Low Code Plattformen gesteigert werden, wenn ein Verständnis für den Begriff Low Code geschaffen wird, da für viele Mitarbeiter in der öffentlichen Verwaltung nicht genau klar ist, was sich hinter der Bezeichnung verbirgt und welche Vorteile diese Technologie mit sich bringt.

Low Code Plattformen bieten das Potenzial die IT-Sicherheit zu verbessern, indem sie die Schatten-IT reduzieren. Das kann gelingen, wenn es den Mitarbeitern in den Behörden so leicht wie möglich gemacht werden, eigene Anwendungen zu erstellen, die ihren Anforderungen entsprechen. Wichtig ist, dass klare Vereinbarungen und Sicherheitsregeln definiert werden, die festlegen, welche Anwendungen in den Fachabteilungen und welche in der zentralen IT verwaltet werden. Darüber hinaus müssen entwickelte Anwendungen in die Anwenderlandschaft integriert werden, damit ein effizienter Datenaustausch ermöglicht wird. Geschieht dies nicht, werden

Daten nach wie vor in eine Excel-Datei exportiert, um sie an anderer Stelle wieder zu importieren.

Um das volle Potenzial von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung auszuschöpfen, ist es ebenfalls wichtig, die Nachteile von Low Code Plattformen zu kennen und vorab klar zu kommunizieren. Durch die Entscheidung für eine Low Code Plattform begibt sich die Behörde in ein Abhängigkeitsverhältnis mit dem Plattformanbieter, da Anwendungen nur eingeschränkt exportiert werden können und somit eine Bearbeitung der bereits bestehenden Anwendungen zum Problem werden kann, wenn ein Wechsel der Low Code Plattform als sinnvoll erachtet wird. Zusätzlich wird mit Einführung einer Low Code Plattform die Prüfung der eigenen Sicherheitsstandards erschwert, da man dem Plattformhersteller vertraut, sich gegen Cyberangriffe zu schützen. Die Einarbeitungszeit für die Nutzung einer Plattform dauert erfahrungsgemäß länger als von den Herstellern angegeben, da von den Nutzern ein IT-Grundverständnis angenommen wird, das häufig nicht vorhanden ist. Außerdem kann die komplexe Integration, Weiterentwicklung und Anpassung von bereits funktionsfähigen umfangreichen Kernprozessen des Kunden in die Low Code Plattform, zu Problemen führen und mit hohen Kosten verbunden sein, wenn Funktionen aus dem bisher laufenden System mit den standardisierten Möglichkeiten der Low Code Plattform nicht umgesetzt werden können und eine individuelle Alternative benötigt wird. Des Weiteren gilt es zu erwähnen, dass Low Code Plattformen bei betriebssystemnahen Anwendungen mit individuellen Anforderungen und Anwendungen, bei denen der Sicherheitsaspekt eine entscheidende Rolle spielt, eher ungeeignet sind.

Damit sich das Potenzial von A12 bei der Softwareentwicklung in der öffentlichen Verwaltung bestmöglich entfalten kann, ist es wichtig Probleme zu identifizieren, die mithilfe von A12 gelöst werden. A12 bietet das Potenzial, Probleme bei der Anforderungsermittlung in Form von Missverständnissen zu minimieren und eine schnelle Umsetzung zu ermöglichen. Dies ist möglich, da die Fachlichkeit von der Technik getrennt wird, sodass sich alle Beteiligten auf ihre Kernkompetenzen konzentrieren können und dadurch die Low Code Entwicklung und Wartung von Anwendungen mit komplexer Thematik wie z.B. Steuern möglich ist. Außerdem bewirkt die Trennung zwischen Fachlichkeit und Technik, dass Fachexperten die Konfiguration bzw. Gestaltung der Anwendung übernehmen können und somit weniger Kapazitäten von klassischen Entwicklern im Softwareprojekt benötigt werden.

Für eine optimale Potenzialentfaltung von A12 bei der Softwareentwicklung in der öffentlichen Verwaltung, müssen die Herausforderungen klar identifiziert werden. Zu Beginn des Softwareentwicklungsprojektes liegt die Schwierigkeit darin, den Gesamtaufwand der zu entwickelnden Software einzuschätzen, da bestimmt werden muss, welche Softwareanforderungen mit den standardmäßigen Funktionalitäten umgesetzt werden können und welche individuell programmiert werden müssen. Dies ist nicht immer leicht einzuschätzen.

A12 bietet außerdem das Potenzial, die IT-Sicherheit zu verbessern, wenn das gesamte Softwareprojekt mit A12 umgesetzt wird und nicht Front- und Backend mit unterschiedlichen Plattformen entwickelt werden.

Bei der Entwicklung von kleinen Anwendungen und in sich geschlossenen Systemen, ist der Einsatz von A12 nicht optimal. Grund dafür ist, dass es sich bei A12 um eine Enterprise Low Code Plattform handelt, die sich auf komplexe Anwendungen spezialisiert und eine ausführliche Einarbeitung voraussetzt. Zudem können die reellen Prozesse in der öffentlichen Verwaltung nicht vollständig ohne individuelle Anpassungen dargestellt werden.

Zusammenfassend konnte im Rahmen dieser Arbeit dargelegt werden, dass Low Code Plattformen das Potenzial bieten, mit weniger IT-Experten eine qualitativ hochwertigere Software in kürzerer Zeit zu entwickeln. Außerdem ist der Einsatz von Low Code Plattformen gerade dann sinnvoll, wenn es große Überschneidungen zwischen den Anforderungen der zu entwickelnden Software und den Anforderungen der bisherigen Softwares gibt. Low Code Plattformen entfalten gerade dann ihr Potenzial, wenn zum Projektstart keine funktionsfähige Anwendung auf Kundenseite vorhanden ist oder wenn die entwickelte Software häufig angepasst werden muss.

6.1 Ausblick

Eine Möglichkeit das Potenzial von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung zu vergrößern, ist das Einsatzgebiet zu erweitern. Hierfür ist zum einen eine effizientere Unterstützung des Softwareentwicklungsprozesses erforderlich, indem erste Prototypen der Software schneller getestet und verbessert werden und ich als Anwender meine Betriebs-, Test- und Entwicklungsumgebung kenne. Zum anderen kann das Einsatzgebiet erweitert werden, indem die Prozesse und Umgebungen der Softwareentwicklung klar definiert sind und ggf. optimiert werden, um die Qualität der Entwicklung nicht nur im Rahmen von Großprojekten sondern kontinuierlich beurteilen zu können.

Eine weitere Möglichkeit um das Potenzial von Low Code Plattformen bei der Softwareentwicklung zu vergrößern besteht darin neue Anwendungsgebiete zu erschließen. Hierfür bietet das Business Process Management für Fachverfahren in der öffentlichen Verwaltung ein hohes Potenzial. Durch die Digitalisierung von kompletten Prozessen und die visuelle Darstellung in Workflowdiagramme, können Potenziale und Engpasssituationen leichter ermittelt werden. Dadurch können Potenziale und Engpässe ermittelt werden, damit dynamisch darauf reagiert werden kann. Ein weiteres Anwendungsgebiet ist das Antrainieren einer Künstlichen Intelligenz, die per Sprachinterface Anwendungen baut. Aufgrund des modularen Ansatzes von Low Code Plattformen ist es für die Künstliche Intelligenz leichter Anwendungen zu entwickeln, da Module leichter verarbeitet werden können als individueller High Code.

Das Potenzial von A12 bei der Softwareentwicklung in der öffentlichen Verwaltung kann dann erweitert werden, wenn die Low Code Plattform noch häufiger eingesetzt wird. Dies ist dann der Fall, wenn Fachexperten in Zukunft die Möglichkeit besitzen, die Benutzeroberflächen selbst zu gestalten. Eine Verbesserung der Speichermöglichkeiten von Objekten und die modulbasierte Integration von externen Systemen sorgt für eine Reduzierung des Anteils an individueller Programmierung und somit ebenfalls zu einem häufigeren Einsatz. Eine weitere Möglichkeit das Potenzial zu erweitern, ist A12 in Zukunft als Mittel einzusetzen, um Anwendungsentwickler anzusprechen, die in erster Linie kleinere, eigenständige Anwendungen erstellen. Hierdurch kann eine Verbindungsebene zwischen Fachexperten, die konfigurieren, und Enterprise-Entwicklern geschaffen werden. Zusätzlich ist es langfristig möglich, A12 im E-Commerce Bereich zu integrieren, wenn die Funktionalitäten von A12 zunehmen. Das Potenzial von A12 kann zudem gesteigert werden, wenn sich die Konfiguration von A12 vereinfacht, damit mehr Fachexperten Anwendungen erstellen und weniger Entwicklerkapazitäten gebunden werden. Außerdem spielt bei Fachanwendungen in der öffentlichen Verwaltung, der Austausch mit dem Bürger eine immer wichtigere Rolle. Dies kann erreicht werden, indem in A12 zukünftig moderne Kanäle wie Chat oder Videotelefonie unter Einhaltung des Datenschutzes, angeboten werden. Einen positiven Einfluss auf das Potenzial hätte zudem eine eigene Versionierung von A12, um Versionsverwaltungen wie Git oder Bitbucket zu ersetzen und zum anderen weitere Add-ons wie PDF oder Excel-Export für gängige Anwendungen einzuführen.

6.2 Limitierung

Da das Thema Low Code Plattform relativ neu ist, gibt es wenig Literatur, wodurch die Aussagen und Ergebnisse der Arbeit eher als Anstoß für weitere Forschung betrachtet werden sollte. Daher kann es hilfreich sein, in ca. drei Jahren die Aussagen und Ergebnisse der Arbeit mit der zukünftigen Literatur abzugleichen. Außerdem ist die öffentliche Verwaltung sehr divers und undurchsichtig, wodurch es schwierig ist, allgemeingültige Aussagen zu treffen. Hierfür könnte weitere Forschung hilfreich sein, welche die internen Abläufe und die IT-Infrastruktur der einzelnen Behörden analysiert und vergleicht, damit die Ergebnisse der wissenschaftlichen Arbeit differenzierter betrachtet werden können. Die Erkenntnisse der vorliegenden Arbeit können außerdem nur teilweise auf andere Branchen übertragen werden, da sich die private Wirtschaft und die öffentliche Verwaltung unterscheiden. Auch die Anzahl der geführten Interviews in der vorliegenden Arbeit ist nicht ausreichend, um wissenschaftlich fundierte Aussagen zu treffen. Daher sollten die Ergebnisse als Tendenzen betrachtet werden. Es gilt in Zukunft weitere Interviews zu führen, um eine größere Stichprobe zu erhalten und damit wissenschaftlich fundierte Aussagen treffen zu können. Auch die Aussagen der Experten in der vorliegenden Arbeit können schwer überprüft werden, da es, wie bereits erwähnt, an ausreichenden Daten und Informationen fehlt.

Literaturverzeichnis

Abts, Dietmar; Mülder, Wilhelm (2013): Grundkurs Wirtschaftsinformatik. Eine kompakte und praxisorientierte Einführung. 8., überarb. und erw. Aufl. Wiesbaden: Springer Vieweg (Lehrbuch). Online verfügbar unter http://www.vlb.de/GetBlob.aspx?strDisposition=a&strIsbn=9783834816696.

Adrian, Benjamin; Hinrichsen, Sven; Schulz, Andreas; Voß, Ernst (2020): Low-Code-Programmierung als Ansatz zur Gestaltung bedarfsgerechter informatorischer Assistenzsysteme – eine Fallstudie. In: Manfred Bornewasser und Sven Hinrichsen (Hg.): Informatorische Assistenzsysteme in der variantenreichen Montage. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 173–186.

Augsten, Stephan (2019): Was ist Low-Code Development? Online verfügbar unter https://www.dev-insider.de/was-ist-low-code-development-a-794952/, zuletzt geprüft am 18.10.2022.

Awati, Rahul (2022): Definition Boolean. Online verfügbar unter https://www.techtarget.com/whatis/definition/Boolean, zuletzt geprüft am 09.12.2022.

Bahrke, Michael; Kempermann, Hanno; Schmitt, Katharina (2016): eGovernment in Deutschland. Bedeutung und Potenzial für das deutsche Innovationssystem.

Balzert, Helmut (2009): Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. Unter Mitarbeit von Heide Balzert, Rainer Koschke, Uwe Lämmel, Peter Liggesmeyer und Jochen Quante. 3. Auflage. Heidelberg: Spektrum Akademischer Verlag. Online verfügbar unter http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1507176.

Beck, Kent; Andres, Cynthia (2005): Extreme programming explained. Embrace change. 2nd ed. Boston, MA: Addison-Wesley.

Berg, Björn; Knott, Philip; Sandhaus, Gregor (2014): Hybride Softwareentwicklung. Berlin, Heidelberg: Springer Berlin Heidelberg.

bitkom (2021): Die Deutschen fordern mehr Tempo bei der Digitalisierung an ihrem Wohnort. Online verfügbar unter https://www.bitkom.org/Presse/Presseinformation/Deutsche-fordern-mehr-Tempo-Digitalisierung-Wohnort.

bitkom (2022): IT-Fachkräftelücke wird größer: 96.000 offene Jobs. Online verfügbar unter https://www.bitkom.org/Presse/Presseinformation/IT-Fachkraefteluecke-wird-groesser.

Blöbaum, Bernd; Nölleke, Daniel; Scheu, Andreas M. (2016): Das Experteninterview in der Kommunikationswissenschaft. In: Stefanie Averbeck-Lietz und Michael Meyen (Hg.): Handbuch nicht standardisierte Methoden in der Kommunikationswissenschaft. Wiesbaden: Springer Fachmedien Wiesbaden, S. 175–190.

BMI: Definition Once-Only-Prinzip. Online verfügbar unter https://www.onlinezugangsgesetz.de/SharedDocs/faqs/Webs/OZG/DE/servicestandard/nutzerzentrierung-faqs/4-once-only-prinzip.html#:~:text=B%C3%BCrgerinnen%20und%20B%C3%BCrger%20sollten%20ihre,Registerabfragen%20und%20zwischenbeh%C3%B6rdliche%20Datenaustausche%20ersetzt., zuletzt geprüft am 09.01.2023.

BMI: Der Servicestandard für die digitale Verwaltung. Online verfügbar unter https://www.onlinezugangsgesetz.de/Webs/OZG/DE/grundlagen/servicestandard/servicestandard-node.html, zuletzt geprüft am 8.11.22.

BMI: OZG-Leistungen. Online verfügbar unter https://www.onlinezugangsgesetz.de/Webs/OZG/DE/grundlagen/info-ozg/info-leistungen/info-leistungen-node.html, zuletzt geprüft am 8.11.22.

BMI: Rechtliches und Beschlüsse. Online verfügbar unter https://www.onlinezugangsgesetz.de/Webs/OZG/DE/grundlagen/konjunkturprogramm/rechtliches-und-beschluesse/rechtliches-und-beschluesse-node.html, zuletzt geprüft am 8.11.22.

BMI: Servicestandard für die OZG-Umsetzung. Online verfügbar unter https://www.onlinezugangsgesetz.de/SharedDocs/downloads/Webs/OZG/DE/poster-servicestandard.pdf?__blob=publicationFile&v=7, zuletzt geprüft am 8.11.22.

BMI (2013): Gesetz zur Förderung der elektronischen Verwaltung sowie zur Änderung weiterer Vorschriften. Online verfügbar unter https://www.verwaltung-innovativ.de/SharedDocs/Publikationen/Artikel/20130731_EGovG_verkuendet.pdf;jsessionid=2935A606C9AAA04BFCFCD67B11DB6DC5.2_cid332?__blob=publicationFile&v=1, zuletzt geprüft am 7.11.22.

BMI (2014a): Digitale Verwaltung 2020 In Kürze. Regierungsprogramm 18. Legislaturperiode. Online verfügbar unter https://www.verwaltung-innovativ.de/SharedDocs/Publikationen/Regierungsprogramm/management

_summary_DiV_Druckversion.pdf?__blob=publicationFile&v=3, zuletzt geprüft am 7.11.22.

BMI (2014b): Eckpunkte zum Programm „Digitale Verwaltung 2020“. Online verfügbar unter https://www.verwaltung-innovativ.de/SharedDocs/Publikationen/Artikel/deutsch_eckpunkte_digitale_verwaltung_2020.pdf?__blob=publicationFile&v=3, zuletzt geprüft am 7.11.22.

BMI (2017a): Das Onlinezugangsgesetz. Online verfügbar unter https://www.onlinezugangsgesetz.de/Webs/OZG/DE/grundlagen/info-ozg/info-ozg-node.html;jsessionid=723F9BCEC11D2F49908A9CD42A0F15D3.1_cid350, zuletzt geprüft am 7.11.22.

BMI (2017b): Onlinezugangsgesetz (OZG). Die Interaktion zwischen Bürgerinnen, Bürgern und Unternehmen mit der Verwaltung soll in Zukunft deutlich schneller, effizienter und nutzerfreundlicher werden. Online verfügbar unter https://www.bmi.bund.de/DE/themen/moderne-verwaltung/verwaltungsmodernisierung/onlinezugangsgesetz/onlinezugangsgesetz-node.html, zuletzt geprüft am 7.11.22.

BMI (2021): Registermodernisierungsgesetz verkündet. Online verfügbar unter https://www.bmi.bund.de/SharedDocs/pressemitteilungen/DE/2021/04/registermodernisierungsgesetz-verkuendet.html, zuletzt geprüft am 8.11.22.

BMI (2022a): Dashboard Digitale Verwaltung. Online verfügbar unter https://dashboard.ozg-umsetzung.de/, zuletzt geprüft am 9.11.22.

BMI (2022b): E-Government-Gesetz. Online verfügbar unter https://www.verwaltung-innovativ.de/DE/Verwaltungsdigitalisierung/E-Government-Gesetz/e-government-gesetz_node.html, zuletzt geprüft am 7.11.22.

Bock, Alexander C.; Frank, Ulrich (2021): Low-Code Platform. In: *Bus Inf Syst Eng* 63 (6), S. 733–740. DOI: 10.1007/s12599-021-00726-8.

Bogner, Alexander; Littig, Beate; Menz, Wolfgang (2014): Interviews mit Experten. Wiesbaden: Springer Fachmedien Wiesbaden.

Bogner, Alexander; Menz, Wolfgang (Hg.) (2009): Experteninterviews. Theorien, Methoden, Anwendungsfelder. 3., grundlegend überarb. Aufl. Wiesbaden: VS, Verl. für Sozialwiss.

Bogumil, Jörg (2021): Öffentliche Verwaltung. In: Uwe Andersen, Jörg Bogumil, Stefan Marschall und Wichard Woyke (Hg.): Handwörterbuch des politischen Systems der Bundesrepublik Deutschland. Wiesbaden: Springer Fachmedien Wiesbaden, S. 680–689.

Bratincevic, John; Koplowitz, Rob (2021): The Forrester Wave™: Low-Code-Entwicklungs- plattformen für professionelle Entwickler, Q2 2021.

Buber, Renate; Holzmüller, Hartmut H. (2009): Qualitative Marktforschung: Konzepte - Methoden - Analysen. 2., überarbeitete Auflage. Wiesbaden: Gabler Verlag / GWV Fachverlage GmbH, Wiesbaden (SpringerLink Bücher).

Bundesministerium des Innern und für Heimat (2022): Onlinezugangsgesetz (OZG). Online verfügbar unter https://www.bmi.bund.de/DE/themen/moderne-verwaltung/verwaltungsmodernisierung/onlinezugangsgesetz/onlinezugangsgesetz-node.html.

Bundesregierung (2017): "Digitale Verwaltung 2020" Digitalisierung der Verwaltung voranbringen. Online verfügbar unter https://www.bundesregierung.de/breg-de/service/archiv/alt-inhalte/digitalisierung-der-verwaltung-voranbringen-436460, zuletzt geprüft am 7.11.22.

Bundesregierung (2021): Verbesserung digitaler Verwaltungsleistungen. Online verfügbar unter https://www.bundesregierung.de/breg-de/themen/digitaler-aufbruch/registermodernisierungsgesetz-1790176, zuletzt geprüft am 8.11.22.

Bundestag (2017): Verfassungsmäßigkeit von Art. 91c Abs. 5 GG n.F. Portalverbund für Verwaltungsleistungen. Online verfügbar unter https://www.bundestag.de/resource/blob/526410/a85ffd5c5ca5a13e24942c0b7e6be327/wd-3-132-17-pdf-data.pdf, zuletzt geprüft am 8.11.22.

Bunse, Christian (2008): Vorgehensmodelle kompakt. 2. Aufl. [Place of publication not identified]: Springer (kompakt-Reihe).

BWI (2019): Wie gelingt die Konsolidierung der Bundes-IT? Online verfügbar unter https://www.bwi.de/news-blog/blog/artikel/digitaler-staat-konsolidierung-der-bundes-it, zuletzt geprüft am 7.11.22.

Caldeira, João; Cardoso, Jorge (2020): Unveiling process insights from refactoring practices. Computer Standards & Interfaces. Online verfügbar unter https://www.researchgate.net/profile/Joao-Caldeira-2/publication/344953698_Unveiling_process_insights_from_refactoring_practices/links/60210fe7a6fdcc37a80ed25b/Unveiling-process-insights-from-refactoring-practices.pdf?origin=publication_detail, zuletzt geprüft am 18.10.2022.

Capgemini (2017): Studie IT-Trends 2017: Überfordert Digitalisierung etablierte Unternehmensstrukturen? Online verfügbar unter

https://www.capgemini.com/de-de/wp-content/uploads/sites/5/2017/02/it-trends-studie-2017.pdf.

Capgemini (2019): IT-Trends 2019: Intelligente Technologien. Vorreiter erzielen bereits Ergebnisse. Online verfügbar unter https://www.capgemini.com/de-de/wp-content/uploads/sites/5/2019/02/IT-Trends-Studie-2019.pdf, zuletzt geprüft am 18.10.2022.

City of New York (2020): COVID-19: Data. Trends and Totals. Online verfügbar unter https://www1.nyc.gov/site/doh/covid/covid-19-data-totals.page.

Correal, Annie; Jacobs, Andrew (2020): 'A Tragedy Is Unfolding': Inside New York's Virus Epicenter. Online verfügbar unter https://www.nytimes.com/2020/04/09/nyregion/coronavirus-queens-corona-jackson-heights-elmhurst.html.

Da Cruz, Mauro A. A.; Paula, Heitor T. L. de; Caputo, Bruno P. G.; Mafra, Samuel B.; Lorenz, Pascal; Rodrigues, Joel J. P. C. (2021): OLP—A RESTful Open Low-Code Platform. In: *Future Internet* 13 (10), S. 249. DOI: 10.3390/fi13100249.

datenbanken-verstehen: Was ist eine Namenskonvention? Online verfügbar unter https://www.datenbanken-verstehen.de/namenskonventionen-datenbank/namenskonvention-definition/, zuletzt geprüft am 9.12.22.

Degenhardt, Daniel (2020): Low-Code-Plattformen und deren Ursprünge. In: *Researchgate*.

Drüke, Helmut (2008): Standardisierung im E-Government – Herausforderungen und Chancen für Nordrhein-Westfalen. Hg. v. Informationsbüros d-NRW. Online verfügbar unter https://silo.tips/download/standardisierung-im-e-government-herausforderungen-und-chancen-fr-nordrhein-west.

Duden: Definition Syntax. Online verfügbar unter https://www.duden.de/rechtschreibung/Syntax, zuletzt geprüft am 25.11.2022.

Duden (2022): Wortbedeutung "pro-p-ri-e-tär". Online verfügbar unter https://www.duden.de/rechtschreibung/proprietaer, zuletzt geprüft am 31.10.22.

European Center for Digital Competitiveness (2021): Digitalreport 2021. Online verfügbar unter https://digital-competitiveness.eu/wp-content/uploads/Digitalreport_2021-2.pdf.

European Commission (2022): Digital Economy and Society Index (DESI) 2022. Digital public services. Online verfügbar unter https://digital-strategy.ec.europa.eu/en/policies/desi-digital-public-services.

Flick, Uwe; Kardorff, Ernst von; Steinke, Ines (Hg.) (2000): Qualitative Forschung. Ein Handbuch. Reinbek bei Hamburg: Rowohlt Taschenbuch Verlag (Rororo, Rowohlts Enzyklopädie).

Frank, Ulrich; Maier, Pierre; Bock, Alexander (2021): Low Code Platforms: Promises, Concepts and Prospects: A Comparative Study of Ten Systems. Unter Mitarbeit von DuEPublico: Duisburg-Essen Publications online, University of Duisburg-Essen, Germany.

Franz, Thorsten (2013): Einführung in die Verwaltungswissenschaft. Wiesbaden: Springer Fachmedien Wiesbaden.

Friedrich, Jan (2009): Das V-Modell XT. Für Projektleiter und QS-Verantwortliche kompakt und übersichtlich. 2. Aufl. 2009. Berlin: Springer (Informatik im Fokus). Online verfügbar unter http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1525884.

Garms, Christoph (2021): Von der analogen zur mobilen Instandhaltung. In: *CITplus* 24 (5), S. 40–42. DOI: 10.1002/citp.202100526.

Gläser, Jochen; Laudel, Grit (2010): Experteninterviews und qualitative Inhaltsanalyse als Instrumente rekonstruierender Untersuchungen.

Goll, Joachim (2011): Methoden und Architekturen der Softwaretechnik. Wiesbaden: Vieweg+Teubner Verlag (Studium).

Haberl, Thomas (2022): OZG kurz vor der Deadline - IT-Fachkräftemangel bremst Digitalisierung aus. Online verfügbar unter https://www.infopoint-security.de/ozg-kurz-vor-der-deadline-it-fachkraeftemangel-bremst-digitalisierung-aus/a32601/, zuletzt geprüft am 9.11.22.

Hansen, Hans Robert; Neumann, Gustaf (2005): Wirtschaftsinformatik. 9., neu bearb. Aufl. Stuttgart: Lucius & Lucius (Grundwissen der Ökonomik Betriebswirtschaftslehre, 2670).

Helfferich, Cornelia (2022): Leitfaden- und Experteninterviews. In: Nina Baur und Jörg Blasius (Hg.): Handbuch Methoden der empirischen Sozialforschung. Wiesbaden: Springer Fachmedien Wiesbaden, S. 875–892.

Henrich, Andreas (2002): Management von Softwareprojekten. 1. Aufl. München, Wien: Oldenbourg (Lehr- und Handbücher der praktischen Informatik).

Hertie School (2019): Ergebnisse des Zukunftspanels Staat & Verwaltung 2019. Online verfügbar unter https://www.zukunftskongress.info/sites/default/files/2019-11/Ergebnisse%20Zukunftspanel%202019.pdf.

Heuermann, Roland; Tomenendal, Matthias; Bressem, Christian (2018): Digitalisierung in Bund, Ländern und Gemeinden. Berlin, Heidelberg: Springer Berlin Heidelberg.

ibo Akademie: Definition Umsystem. Online verfügbar unter https://www.ibo.de/glossar/definition/umsystem, zuletzt geprüft am 09.12.2022.

INSM (2022): Deutschland scheitert beim E-Government. Online verfügbar unter https://www.insm.de/insm/themen/digitalisierung/deutschland-scheitert-beim-e-government, zuletzt geprüft am 9.11.22.

IT-Planungsrat (2020): Die Umsetzung des Onlinezugangsgesetzes (OZG). Online verfügbar unter https://www.it-planungsrat.de/foederale-zusammenarbeit/ozg-umsetzung, zuletzt geprüft am 7.11.22.

ITZBund (2015): IT-Konsolidierung Bund. Online verfügbar unter https://www.itzbund.de/DE/digitalemission/itkonsolidierungbund/itkonsolidierungbund.html, zuletzt geprüft am 7.11.22.

Jackwerth-Rice, Thomas; Böcker, Dominic; Derse, Matthias; Elsebrock, Kirsten; Jehnichen, Thomas; Kley, Thomas et al. (2022): Gestaltung einer digitalisierungsförderlichen Unternehmenskultur bei mittelständischen Industrieunternehmen. In: Verena Nitsch, Christopher Brandl, Roger Häußling, Jacqueline Lemm, Thomas Gries und Bernhard Schmenk (Hg.): Digitalisierung der Arbeitswelt im Mittelstand 1. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 99–132.

Jochimsen, Beate (2022): Lehren aus der Covid-19-Pandemie: Der Weg zu größerer Resilienz im deutschen Gesundheitssystem. In: *ifo Schnelldienst*. Online verfügbar unter https://www.ifo.de/DocDL/sd-2022-03-resilienz-deutsches-gesundheitssystem.pdf, zuletzt geprüft am 17.10.2022.

Kahanwal, Brijender (2013): Abstraction Level Taxonomy of Programming Language Frameworks. In: *IJPLA* 3 (4). Online verfügbar unter https://www.researchgate.net/profile/Brijender-Kahanwal-2/publication/258499542_Abstraction_Level_Taxonomy_of_Programming_Language_Frameworks/links/593683010f7e9beee7f11411/Abstraction-Level-Taxonomy-of-Programming-Language-Frameworks.pdf?origin=publication_detail, zuletzt geprüft am 18.10.2022.

Kneuper, Ralf; Müller-Luschnat, Günther; Oberweis, Andreas (Hg.) (1998): Vorgehensmodelle für die betriebliche Anwendungsentwicklung. Wiesbaden: Vieweg+Teubner Verlag (Teubner-Reihe Wirtschaftsinformatik).

Kostädt, Peter (2022): IT-Organisation in Hochschulen und ihren Bibliotheken. Hg. v. De Gruyter. Online verfügbar unter https://www.degruyter.com/document/doi/10.1515/bfp-2022-0009/pdf.

Kukat, Falk; Human, Sebastian (2020): Low-Code vs. No-Code: Gemeinsamkeiten, Unterschiede und Einsatzszenarien. Online verfügbar unter https://www.industry-of-things.de/low-code-vs-no-code-gemeinsamkeiten-unterschiede-und-einsatzszenarien-a-937195/, zuletzt geprüft am 18.10.2022.

Landesarchiv Baden-Württemberg: Definition Fachverfahren. Online verfügbar unter https://www.leo-bw.de/themenmodul/sudwestdeutsche-archivalienkunde/archivaliengattungen/strukturierte-informationssammlungen/fachverfahren, zuletzt geprüft am 09.01.2023.

Ludewig, Jochen; Lichter, Horst (2007): Software Engineering. Grundlagen, Menschen, Prozesse, Techniken. 1. Aufl. Heidelberg: dpunkt-Verl. Online verfügbar unter http://bvbr.bib-bvb.de:8991/F?func=service&doc_library=BVB01&doc_number=012989845&line_number=0002&func_code=DB_RECORDS&service_type=MEDIA.

Mayring, Philipp (2016): Einführung in die qualitative Sozialforschung. Eine Anleitung zu qualitativem Denken. 6., überarbeitete Auflage, Online-ausgabe. Weinheim und Basel: Beltz Verlag (Pädagogik). Online verfügbar unter http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1127318.

Mayring, Philipp; Brunner, Eva (2009): Qualitative Inhaltsanalyse. In: Renate Buber und Hartmut H. Holzmüller (Hg.): Qualitative Marktforschung. Wiesbaden: Gabler, S. 669–680.

Mehmanesh, Hamarz; Gandenberger, Sabine; Weiss, Ansgar; Kneist, Thomas; Lorenz, Sebastian (2022): A12 - Low code for custom enterprise software (02/2022). Online verfügbar unter https://geta12.com/assets/whitepaper/A12_Whitepaper_EN.pdf.

mgm (2022a): A12. Frequently Asked Questions (FAQ). Online verfügbar unter https://a12.mgm-tp.com/insights/mgm-A12-FAQ.pdf.

mgm (2022b): mgm consulting partners. Innovation Implemented. Online verfügbar unter https://www.mgm-cp.com/wp-content/uploads/2022/06/mgm-consulting-partners-2022.pdf.

Möltgen-Sicking, Katrin; Winter, Thorben (2018): Verwaltung und Verwaltungswissenschaft. Wiesbaden: Springer Fachmedien Wiesbaden.

Mordor Intelligence (2021): Markt für Low-Code-Entwicklungsplattformen – Wachstum, Trends, Auswirkungen von COVID-19 und Prognosen (2022 –

2027). Online verfügbar unter https://www.mordorintelligence.com/de/industry-reports/low-code-development-platform-market, zuletzt geprüft am 20.10.2022.

Nationaler Normenkontrollrat (2021): Monitor Digitale Verwaltung #5.

Pathfinder Report (2018): Intelligent Process Automation and the Emergence of Digital Automation Platforms. The transformation of application development and its emerging role to enable new competitive advantage. Online verfügbar unter https://www.redhat.com/cms/managed-files/mi-451-research-intelligent-process-automation-analyst-paper-f11434-201802.pdf.

Perplies, Philipp (2022): Onlinezugangsgesetz 2022: Wo steht die bürgerfreundlichste Verwaltung Europas heute? d.velop. Online verfügbar unter https://www.d-velop.de/blog/compliance/onlinezugangsgesetz-2022/, zuletzt geprüft am 9.11.22.

Pichler, Roman (2013): Scrum. Agiles Projektmanagement erfolgreich einsetzen. 1. Auflage. Heidelberg: dpunkt.verlag. Online verfügbar unter http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1299575.

PR Newswire (2020): New York City has Deployed a Covid-19 Engagement Portal Built in Partnership with Unqork to Manage Growing Crisis. Unqork built the portal in 72 hours using its unique enterprise no-code platform Additional capabilities include medical equipment donations and GetFoodNYC Delivery. Online verfügbar unter https://www.prnewswire.com/news-releases/new-york-city-has-deployed-a-covid-19-engagement-portal-built-in-partnership-with-unqork-to-manage-growing-crisis-301033065.html?tc=eml_cleartime/.

Rentrop, Christopher; van Laak, Oliver; Mevius, Marco (2011): Schatten-IT: ein Thema für die Interne Revision. Hg. v. IT-Revision. Online verfügbar unter https://www.researchgate.net/profile/Christopher-Rentrop/publication/281510845_Schatten-IT_ein_Thema_fur_die_Interne_Revision/links/55fbb7db08aec948c4afb11d/Schatten-IT-ein-Thema-fuer-die-Interne-Revision.pdf.

Richardson, Clay; Rymer, John (2014): New Development Platforms Emerge For Customer-Facing Applications. Online verfügbar unter https://silo.tips/queue/new-development-platforms-emerge-for-customer-facing-applications?&queue_id=-1&v=1666100451&u=MTQxLjQ3LjEyMi4xODA=, zuletzt geprüft am 18.10.2022.

Rosa-Bilbao, Jesús; Boubeta-Puig, Juan; Rutle, Adrian (2022): EDALoCo: Enhancing the accessibility of blockchains through a low-code approach to the development of event-driven applications for smart contract

management. In: *Computer Standards & Interfaces* 84, S. 103676. DOI: 10.1016/j.csi.2022.103676.

Rudl, Thomas (2021): IT-Konsolidierung des Bundes: Pleitenserie ohne Ende. Netzpolitik. Online verfügbar unter https://netzpolitik.org/2021/it-konsolidierung-des-bundes-pleitenserie-ohne-ende/, zuletzt geprüft am 9.11.22.

Rumpe, Bernhard (2012): Agile Modellierung Mit UML. Codegenerierung, Testfälle, Refactoring. 2nd ed. Berlin, Heidelberg: Springer Berlin / Heidelberg (Xpert. press Ser). Online verfügbar unter https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=968550.

Sahay, Apurvanand; Indamutsa, Arsene; Di Ruscio, Davide; Pierantonio, Alfonso (2020): Supporting the understanding and comparison of low-code development platforms. In: 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). Portoroz, Slovenia, 26.08.2020 - 28.08.2020: IEEE, S. 171–178.

Sahinaslan, Ender; Sahinaslan, Onder; Sabancıoglu, Mehmet (2021): Low-code application platform in meeting increasing software demands quickly: SetXRM. In: FOURTH INTERNATIONAL CONFERENCE OF MATHEMATICAL SCIENCES (ICMS 2020). FOURTH INTERNATIONAL CONFERENCE OF MATHEMATICAL SCIENCES (ICMS 2020). Istanbul, Turkey, 17–21 June 2020: AIP Publishing (AIP Conference Proceedings), S. 70007.

Scharch, Manuel (2016): Vorgehensmodelle in der Software-Entwicklung.

Scheu, Andreas M. (2011): Adornos Erben in der Kommunikationswissenschaft. Eine Verdrängungsgeschichte? Zugl.: München, Univ., Diss., 2010. Neue Ausg. Köln: Herbert von Halem Verlag (Theorie und Geschichte der Kommunikationwissenschaft, 11).

Siepermann, Markus (2018): Definition Persistenz. Online verfügbar unter https://wirtschaftslexikon.gabler.de/definition/persistenz-53469/version-276558, zuletzt geprüft am 09.12.2022.

Smitham, Mark (2022): Mit Low-Code den digitalen Behördendienst beschleunigen. Online verfügbar unter https://www.egovernment-computing.de/mit-low-code-den-digitalen-behoerdendienst-beschleunigen-a-59765fc8840f4a15d40b81d47a82f09b/, zuletzt geprüft am 18.10.2022.

Sommerville, Ian (2012): Software Engineering. München: Pearson Deutschland; Pearson Studium (Pearson Studium - IT). Online verfügbar unter https://elibrary.pearson.de/book/99.150005/9783863265120.

Strack, Rainer; Carrasco, Miguel; Kolo, Philipp (2021): The Future of Jobs in the Era of AI. Online verfügbar unter https://web-assets.bcg.com/ec/bc/da7341af41358367d26db742eb6c/bcg-the-future-of-jobs-in-the-era-of-ai-may-2021-r.pdf.

Streicher, Hans Werner (2020): Digitale Transformation in der öffentlichen Verwaltung. Berlin, Heidelberg: Springer Berlin Heidelberg.

t2informatik: Artefakt – das erzeugte Ergebnis. Online verfügbar unter https://t2informatik.de/wissen-kompakt/artefakt/, zuletzt geprüft am 14.11.2022.

techconsult (2021): Low-Code-/No-Code-Development Enabler der digitalen Transformation. Online verfügbar unter https://a12.mgm-tp.com/insights/Low-Code-Studie%202021-mgm-technology-partners.pdf.

United Nations (2022a): UN E-Government Knowledgebase: Country Data. Online verfügbar unter https://publicadministration.un.org/egovkb/en-us/Data-Center.

United Nations (2022b): UN E-Government Knowledgebase: E-Government Development Index (EGDI). Online verfügbar unter https://publicadministration.un.org/egovkb/en-us/About/Overview/-E-Government-Development-Index.

Vial, Gregory (2019): Understanding digital transformation: A review and a research agenda.

Vogel, Jannis; Thomas, Oliver (Hg.) (2020): Low-Code-Plattformen zur Verbreitung von Wearable-Technologien. Berlin, Heidelberg: Springer Berlin Heidelberg.

Vollmann, Katalin (2021): Die öffentliche Verwaltung. In: Katalin Vollmann (Hg.): Compliance in der öffentlichen Verwaltung. Wiesbaden: Springer Fachmedien Wiesbaden, S. 17–20.

Waszkowski, Robert (2019): Low-code platform for automating business processes in manufacturing. In: *IFAC-PapersOnLine* 52 (10), S. 376–381. DOI: 10.1016/j.ifacol.2019.10.060.

Windoffer, Alexander (2018): Herausforderungen der Digitalisierung aus der Perspektive der öffentlichen Verwaltung. In: Christian Bär, Thomas Grädler und Robert Mayr (Hg.): Digitalisierung im Spannungsfeld von Politik, Wirtschaft, Wissenschaft und Recht. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 363–376.

wirtschafts-lehre (2016): Lock-in-Effekt. Online verfügbar unter https://www.wirtschafts-lehre.de/lock-in-effekt.html, zuletzt aktualisiert am 2.3.16, zuletzt geprüft am 14.11.22.

Wolf, Henning (Hg.) (2012): Agile Projekte mit Scrum, XP und Kanban im Unternehmen durchführen. Erfahrungsberichte aus der Praxis. Online-Ausg. Heidelberg: dpunkt.verl. (EBL-Schweitzer). Online verfügbar unter http://swb.eblib.com/patron/FullRecord.aspx?p=952243.

Anhang

Interview P1

In welchem Bereich wird A12 eingesetzt und welche Funktion erfüllt A12?

- A12 wird im Bereich Steuern eingesetzt.
- Mit A12 wird ein Tool für Sachbearbeiter im Steuerbereich gebaut.
- MGM Mitarbeiter implementieren die Fachlichkeit über die Modellierung.

Können Sie kurz skizzieren, wie der Softwareentwicklungsprozess in Ihrem Projekt abläuft?

- Tickets mit Anforderungen werden in Jira erstellt.
- Business Analysten stimmen sich mit den Kunden ab.
- Die Business Analysten bestimmen wie aufwendig das Ticket ist und welche Komponenten es betrifft.
- Bei reinem Modellierungstickets muss nichts abgestimmt werden.

Welche Rolle/Positionen nehmen Sie bei der Softwareentwicklung in Ihrem Projekt ein?

- Erstes Projekt: Modellierer der Anforderungen in Form von Tickets mit A12 umsetzt.
- Zweites Projekt: Modellierer und Business Analyst in einer Person. Bearbeitet teilweise Tickets aber setzt sich auch mit den Inhalten der Formulare auseinander und erstellt Tickets mit Anforderungen.

Welche IT-Kenntnisse sind hierfür erforderlich? Wird ein IT-Affinität vorausgesetzt?

- IT-Kenntnisse sind nicht erforderlich. Es gibt viele Quereinsteiger aus der Naturwissenschaft aber auch aus dem BWL-Bereich.
- Mitarbeiter mit einer 50%-Stelle können in ca. einem halben Jahr mit A12 modellieren und eigene Tickets umsetzen.
- Interesse neue Dinge zu lernen sehr wichtig (Mindset).

Weißt du wie der Softwareentwicklungsprozess vor A12 ablief?

- Klassischer Entwicklungsprozess mit phasenorientierter Vorgehensweise. Dabei sind die typischen Probleme aufgetreten.
- Ineffizient da Anwender erst am Ende des Entwicklungsprozesses Verbesserungsvorschläge einbringen kann.
- Softwareentwickler musste sich mit der Fachlichkeit auskennen.

Wann ist der Einsatz einer Low Code Plattform in der Softwareentwicklung sinnvoll?

- Sinnvoll in der öffentlichen Verwaltung Low Code Plattformen einzusetzen, da aufgrund von Gesetzesänderungen regelmäßig kleine Änderungen notwendig sind und somit das Programm nicht immer neu geschrieben werden muss. Außerdem lohnt sich der Einsatz von Low Code Plattformen gerade bei langfristigen Projekten, die in der öffentlichen Verwaltung üblich sind.

- Im Bereich Steuern und Finanzen gibt es viele Gesetze zu beachten, was eine hohe fachliche Expertise voraussetzt. Mit Low Code Plattformen kann die Fachlichkeit von der Technik getrennt werden. Dies bedeutet, dass sich die Softwareentwickler auf ihre Kernkompetenzen konzentrieren können und sich nicht mehr mit Steuergesetzten auseinandersetzen müssen.
- Mit Low Code Plattformen können Module erstellt werden, die gewisse Funktionen erfüllen und im Anschluss auch für andere Projekte einsetzbar sind. Dies ist nur möglich, wenn alle Projekte auf dem gleichen technischen Fundament aufbauen und allgemeine Standards gesetzt werden.

Gibt es weitere Anpassungen, die sie sich im Entwicklungsprozess wünschen?

- Versionierung in A12 umsetzen. Gerade für Personen mit weniger Programmiererfahrung kann dies hilfreich sein, da sie nicht die klassischen Programmiertools nutzen müssen.

Können Sie beurteilen, ob die Usability der Software durch den Einsatz von A12 gestiegen ist?

- Die Benutzerfreundlichkeit bzw. die Zufriedenheit der Benutzer kann schneller erreicht werden.
- Es kann in kürzeren Abständen Anpassungen umgesetzt werden.

Gibt es aktuell Probleme oder Herausforderungen bei der Softwareentwicklung mit A12?

- Mit A12 können nur begrenzt individuelle Probleme umgesetzt werden, da die Möglichkeiten beschränkt sind.
- Modellierer kann nur mit dem ihm zu Verfügung stehenden Tools arbeiten.

Welche Voraussetzungen müssen erfüllt sein, damit eine Softwareentwicklung erfolgreich abläuft?

- Der Projektleiter muss die Aufgabengebiete und Fähigkeiten der Projektbeteiligten kennen und verstehen wie viel mit Modellierern gelöst werden kann und wie viele individuelle Entwickler benötigt werden.
- Wichtig ist, dass die Entwickler verstehen, dass Modellierer Arbeit abnehmen
- Projektbeteiligte müssen wissen, an wen sie sich bei Problemen wenden müssen. Muss der Modellierer bei einem Problem mit dem Front- oder Backendentwickler sprechen?

Konnte der Auslastungsgrad von IT-Fachabteilungen gesenkt werden?

- Entwickler muss sich nicht mehr so stark mit der Fachlichkeit beschäftigen.
- Mehr Zeit sich mit den wichtigen IT-Themen wie z.B. Sicherheit zu beschäftigen.

Konnte die Einführung von A12 die Nutzung von in den Fachabteilungen erstellte IT-Produkte oder Applikation reduzieren? (Schatten IT)

- Glaube wenn Fachexperten ein Tool brauchen, werden sie immer noch zu Excel greifen, da es am schnellsten umgesetzt werden kann.

- Wenn ein Projekt startet, wird eventuell Schatten-IT reduziert aber es werden immer neue Tools entstehen, da auch nicht alle modellieren können/möchten.

Interview P2

Was verstehst du unter dem Begriff Low Code Plattform?

- Eine Low Code Plattform ist zusammengesetzt aus verschiedenen Werkzeugen, die es schon sehr lange in der Softwareentwicklung gibt. Diese verschiedenen Werkzeuge werden nahtlos in einer Plattform vereint.

Gibt es Bereiche in der öffentlichen Verwaltung in denen Low Code Plattformen besser geeignet sind als in anderen?

- Eignet sich dann sehr gut wenn Systeme häufig angepasst werden müssen. Anpassungen können mit kurzen Wegen durchgeführt werden.
- Bei sehr individuellen Anforderungen ist eine Low Code Plattform eher nicht geeignet

Ist die Softwareentwicklung seit der Einführung von Low Code nun schneller, einfacher und günstiger?

- Man kann viel wiederverwenden und macht vieles leichter, wenn an anderer Stelle ein ähnliches Problem aufgetreten ist.
- Probleme wenn ich Dinge anpassen muss die in der Plattform nicht vorgesehen sind - also sehr spezifische Änderungen.

Kannst du einschätzen, ob sich der Auslastungsgrad der Softwareentwickler vermindert hat?

- Kein Fall bekannt in der es so gut funktioniert, dass die Softwareentwickler weniger ausgelastet sind.
- Umstrukturierung findet noch statt.
- Der Softwareentwickler kann sich im Idealfall auf komplexere Probleme konzentrieren.

Kann mithilfe von Low Code Plattformen die Schatten-IT reduziert werden?

- Kann man so pauschal nicht sagen.
- Hilft wenn man die Mitarbeiter schult.
- Möglichst leichten Zugang zur Modellierung ermöglichen.
- Kommt drauf an, wie bedienerfreundlich das Tool zur Erstellung von Anwendungen ist.

Welche Funktionen/Eigenschaften müssen Low Code Plattform noch bekommen damit es noch häufiger eingesetzt wird und einen größeren Mehrwert bietet?

- Unternehmen unterschätzen die Einarbeitungszeit zur Modellierung mit Low Code Plattformen.
- IT-Affinität ist in den meisten Fällen eine Voraussetzung zur Bedienung von Low Code Plattformen. Für Personen, die keinen Bezug zu IT-Themen haben ist daher der Einstieg schwer.
- Wenn eine Low Code Plattform so gestaltet ist, dass sie eine Oberfläche bietet, die am Ende die gleichen Möglichkeiten bietet wie

das klassische Programmieren, ist es nicht unbedingt zugänglicher, da man am Ende denken muss wie ein Programmierer.
- Für viele Mitarbeiter ist es schon schwer das Prinzip von Bool(True/False) zu verstehen.
- Schwierig den Funktionsumfang der Low Code Plattform abzuschätzen. Kann ich alles was ich brauche am Ende auch mit der Low Code Plattform umsetzen. Damit ich am Ende die eine Funktion die ich zwingend brauche nicht umsetzen kann.
- Im schlimmsten Fall braucht man mehrere Low Code Plattformen, um seine Ziele zu erreichen.
- Wichtig zu klären, ob für die Mitarbeiter das Wissen bezüglich Low Code Plattformen nur unternehmensspezifisch anwendbar ist oder ob es dem Mitarbeiter einen langfristigen Mehrwert bietet.

Interview P3

Welche Funktion erfüllt A12 in deinem Projekt? Wie wird A12 eingesetzt?
- A12 hat die Rolle als technologische Basis für das Projekt. Somit starten wir in dem Projekt nicht bei null sondern ca. 1/3 ist schon vorhanden.
- Die Architektur von A12 kann genutzt werden was bedeutet, dass von vornerein lauffähige Applikationen vorhanden sind, die agil weiterentwickelt werden.
- Großer Vorteil ist, dass erste Templates schon während des Ausschreibungsprozesses dem Auftraggeber präsentiert werden konnten, um zu zeigen was möglich ist bzw. was gemacht werden kann.
- Vorteil von A12 die Formulare der Anträge für die Steuerkammer schnell zu modellieren und der Entwickler muss sich nicht um die Erstellung der Formulare kümmern.
- Die Entwickler kümmern sich um das Portal selbst (bilden einen Rahmen) und darin werden die neuesten Modelle gehangen, die ständig aktualisiert werden. Der Entwickler schafft diesen Rahmen und definiert die Funktionen wie z.B. was passiert, wenn man auf abschicken klickt.

Welche Voraussetzungen müssen erfüllt sein, damit der Einsatz von Low Code Plattformen in der öffentlichen Verwaltung erfolgreich und sinnvoll ist?
- Projektleiter muss einschätzen wie viele Business Analysts und Entwickler für das Projekt benötigt werden.
- Für den Projektmanager ist es wichtig die Grundzüge und Grundarchitektur von A12 zu verstehen. Es ist wichtig einschätzen zu können, ob Anforderungen des Kunden standardmäßig verfügbar sind oder ob es separat entwickelt werden muss.
- So gut wie möglich den Kunden dafür zu sensibilisieren, dass er eher die Lösung bevorzugt, die Out of the box kommt und dem Kunde klar machen wie viel Mehraufwand weitere gewünschte Funktionen mit sich bringen und wie viel diese kosten würden.
- Verzahnung von Projektanforderungen und A12-Timeline. Wichtig zu wissen, ob es Anforderungen von Seiten des Kunden gibt, die aktuell nicht mit A12 umsetzbar sind und diese rechtzeitig an die A12-Pipline (Entwickler) weitergeben, damit die Funktionen rechtzeitig verfügbar

sind. Wichtig dabei nicht selbst anzufangen zu entwickeln, sondern es über A12 zu bekommen, damit es auch zentral gewartet wird.

Ist die Softwareentwicklung seit der Einführung von Low Code nun schneller, einfacher und günstiger?

- Ja auf jeden Fall ist der Entwicklungsprozess schneller. Man muss nicht bei null beginnen, da Grundthemen schon vorgedacht wurden wie z.B. responisve Design oder Barrierefreiheit.
- Modellieren, Abstimmen und Kundenfeedback kann sehr schnell vorangetrieben werden.
- Bei A12 können Designpattern, welche die Benutzeroberfläche managen, wiederverwendet werden.
- Bei A12 können Softwaremodule wie z.B. Authentifizierungsmechanismus wiederverwendet werden.
- Bei A12 können Widgets (Vorgefertigte Elemente wie Listen, die man nicht separat entwickeln muss) verwendet werden.

Gibt es weitere Anpassungen, die sie sich im Entwicklungsprozess wünschen?

- A12 besitzt grundsätzlich einen guten Reifegrad.
- Hilfreich wären zusätzliche Add-ons wie z.B. Export als PDF oder als Excel-Datei.
- Grundsätzlich sind Add-ons mit Funktionen, die so binärisch sind, dass sie jedes Projekt braucht hilfreich.

Konnte der Auslastungsgrad der Softwareentwickler durch Low Code Plattformen gesenkt werden?

- Die absolute Zeit, die Softwareentwickler für ein Projekt aufbringen reduziert sich, da sich die Aufgaben verschieben.
- Ab Tag 1 ist es möglich Funktionalitäten für den Kunden zu entwickeln.

Schatten-IT

- Voraussetzung, um Schatten-IT zu reduzieren bzw. zu verhindern ist, dass das ganze Projekt mit A12 entwickelt wird und nicht bspw. eine Firma das Frontend und die andere das Backend erstellt. Dann sind keine manuellen Schritte notwendig.
- Grundsätzlich ist es abhängig von dem Projektscope.

Interview 4

Welche Funktion erfüllt A12 in deinem Projekt? Wie wird A12 eingesetzt?

- Wir arbeiten in unserem Projekt nicht direkt mit A12 sondern mit Toko, das mit A12 gebaut wurde.
- Toko wird für die Oberflächenmodellierung für Elsterformulare eingesetzt.
- Datenmodelle bekommen wir vom Kunden geliefert mit Datenstruktur und Regeln.
- Diese Datenmodelle werden genutzt, um mit Toko Oberflächen für Elster zu gestalten.

Kannst du kurz skizzieren, wie der Softwareentwicklungsprozess in deinem Projekt abläuft?

- Ich kann nur über den Teil sprechen in den ich involviert bin.

- Wenn z.B. bei einem Steuerformular eine Gesetzesänderung ansteht, dann machen wir die Jahresanpassung.
- Wir werden beauftragt die Formularanpassungen im Portal zu übernehmen. Dies geschieht in Form eines Jira Ticket.
- Für die interne Bearbeitung wird eine Kopie des Tickets erstellt.

Welche Rolle/Positionen nehmen Sie bei der Softwareentwicklung in Ihrem Projekt ein?

- Business Analystin, die mit Toko Formulare in Elster modelliert.

Welche IT-Kenntnisse sind hierfür erforderlich? Wird ein IT-Affinität vorausgesetzt?

- Benötigt kein Studium oder Kurse in Bezug auf Programmieren bzw. IT.
- IT-Vorkenntnisse sind nicht notwendig.
- Grundsätzlich logisches/strukturiertes Denken hilfreich.

Wie wurde die Software vor der Verwendung von A12 entwickelt?

- Vor der Einführung von A12 gab es ein Sammelsurium an Werkzeugen, bei dem man sich erst zurechtfinden muss.
- Man musste in jedes Formular(XML) einzeln reingehen. Formulardaten liegen in Form eines XML-Dokuments vor, das vorher manuell bearbeitet werden musste.
- Davor sehr mühsam, wenn es in einem Formular mehrere Änderungen gibt, muss man die einzelnen Stellen an denen Änderungen benötigt werden raussuchen. Selbst wenn sich nur die Feldgrößen ändern. Nicht schwierig aber in einem großen XML-Dokument mühsam die richtigen Stellen zu identifizieren.

Gibt es Verbesserungen in der Softwareentwicklung die du dir wünschst?

- Arbeiten mit Toko seit Mitte Januar
- Idee mit Toko das der Kunde selbst entwickeln kann. Impuls das Mitarbeiter geschult werden, um demnächst mit Toko zu arbeiten und eigene Formulare zu modellieren.
- Ziel: Customer Experts sollen noch mehr mit Toko arbeiten.

Können Sie beurteilen, ob die Usability der Software durch den Einsatz von A12 gestiegen ist?

- Schwer zu beurteilen. Änderungen werden umgesetzt nur mussten früher mehr Ressourcen von Seitens mgm eingesetzt werden. Für den Kunden hat sich aber nichts verändert.
- Zukünftige Verbesserung der Usability, da bisher nur drei Mal im Jahr Verbesserungen veröffentlicht werden und in Zukunft sollen die Formulare von den Releases entkoppelt werden.

Welche Voraussetzungen in Bezug auf z.B. Expertise oder Infrastruktur müssen erfüllt sein, damit eine Softwareentwicklung erfolgreich abläuft?

- Rudimentäre Kenntnisse mit dem Computer und eine gewisse Technikaffinität.
- Gewisses Technikverständnis ist hilfreich, um sich mit Metadaten oder Funktionen auszukennen.

- Bei der Modellierung von Formularen ist eine gewisse Pingelichkeit hilfreich, damit am Ende alles ordentlich und systematisch aussieht von Seiten des Modellierers und derjenigen die es verifizieren.
- Grundsätzlich ist für die Modellierer hilfreich einen Blick dafür zu haben, damit alles rund aussieht.
- Wenn die Formulare mehr mit der Zeit gehen sollen und ansprechender wirken sollen damit die Nutzung Spaß macht und es bspw. in Richtung Interviewstil geht, dann macht es natürlich schon Sinn, wenn die Modellierer sich mit dem Design von Benutzeroberflächen auskennen.

Interview P5

Welche Unterschiede sehen Sie zwischen der Softwareentwicklung mit und ohne Low Code Plattformen in der öffentlichen Verwaltung?

- Vorteil von Low Code Plattformen ist, dass man bei der Softwareentwicklung nicht immer wieder bei null beginnt, sondern schon vorgefertigte Bausteine bzw. Module verwendet, die angepasst werden können.
- Man kann somit bei der Softwareentwicklung mit Low Code wesentlich schneller starten und etwas auf die Beine stellen.
- Weiterer Vorteil ist, dass man schon zu Beginn ein Toolset besitzt, mit dem man in den ersten Meetings direkt etwas vorherzeigbares präsentieren kann - man muss nicht erst eine große Landschaft aufbauen.
- Außerdem können sich die späteren Anwender der Software schon direkt zu Beginn einen ersten Eindruck verschaffen und direkt Anmerkungen geben. Somit können sie direkt mitgestalten.

Welche Voraussetzungen müssen erfüllt sein, um Low Code Plattformen für die Softwareentwicklung in der öffentlichen Verwaltung erfolgreich einzusetzen?

- Wenn man bei der Anforderungsermittlung merkt, dass die Anforderungen sich sehr stark anderen Projekten ähneln und somit schon sehr viel aufbereitet werden kann.
- Grundsätzlich: Je mehr sich die Anforderungen des Projektes mit denen weitere Projekte gleichen, desto größer ist der Vorteil von Low Code Plattformen. Somit können Modelle erzeugt werden, die auf Basis der bereits vorhandenen Modelle aufbaut.
- Vorteil: Alles ist gleichartig aufgebaut und die Oberflächengestaltung fühlt sich gleich an.
- Schwierig wenn von Kundenseite ein hoher Bedarf an Sonderfunktionalitäten besteht, die in noch keinem Projekt umgesetzt wurden und mit den standardisierten Funktionalitäten der Low Code Plattform nicht umsetzbar sind. Hierfür ist dann ein umfangreiches Wissen erforderlich, da die Funktionalitäten in die bereits bestehende Plattform integriert werden muss.

Welche Nachteile gibt es beim Einsatz von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung?

- Grundsätzlich kommt es bei einem hohen Individualisierungsgrad zu Nachteilen.

- Die Entwicklung von Software mit Low Code Plattformen funktioniert dann gut, wenn es noch keine funktionsfähige Anwendung auf Kundenseite gibt. Eine umfangreich funktionsfähige Software auf die Weiterentwicklung bzw. Anpassung mit einer Low Code Plattform kann zu Problemen führen. Problematisch wird es gerade dann, wenn Funktionen aus dem bisher laufenden System mit den standardisierten Möglichkeiten der Low Code Plattform nicht umgesetzt werden können und ein hoher Aufwand an Programmierung erforderlich ist.
- Es ist wichtig schon zu Beginn klar zu kommunizieren, welche Funktionen durch den Einsatz der Low Code Plattform gar nicht oder nur mit einem hohen Programmieraufwand umgesetzt werden können.

Welchen generellen Vorteil bietet Low Code Plattformen und speziell A12?

- Die IT-Spezialisten/Softwareentwickler von mgm können sich auf die technischen Aspekte der Softwareerstellung fokussieren und dabei genau das machen, wofür sie ausgebildet wurden und worin sie gut sind.
- Fachexperten (Customer Experts) können ein Datenmodell und ein Oberflächenmodell erstellen. Außerdem kann der Fachexperte Funktionalitäten durch die Modellierung der Anwendung nutzen, die Modelle verarbeiten können. Somit ist es möglich, das Fachwissen direkt in die Softwareentwicklung einzubringen.
- Somit können Missverständnisse und Mehrfachdeutungen bei der Anforderungsermittlung vermieden werden. Kommt häufig vor da bei der Kommunikation generell Missverständnisse und Mehrfachdeutungen auftreten. Speziell wenn die Beteiligten einen komplett unterschiedlichen Background besitzen.
- Die finanzielle Konkurrenzfähigkeit wird durch unser A12 gesteigert, weil wir auf einen Baukasten zurückgreifen können, mit dem wir schnell etwas Funktionierendes aufsetzen können und folglich ein unschlagbares Angebot abgeben können. Im Gegensatz dazu kostet eine herkömmliche Entwicklung viel mehr Zeit und wäre folgerichtig bedeutend teurer.
- Der Erfahrungsaustausch ist wichtig, weil ohne unser gemeinsames A12 in der Firma viele isoliert arbeitende Projekte lauter tolle Ideen und Lösungen entwickeln, ohne dass andere davon profitieren können und ohne den Vorteil nutzen zu können, dass andere Kollegen schon vor gleichen Problemen standen und sie bereits gelöst haben. Durch den Austausch identifiziert man auch tolle Lösungen, die in den gemeinsamen „A12-Topf“ zurückfließen könnten.

Low Code Plattformen Schatten IT reduzieren?

- Grundsätzlich können Low Code Plattformen die Schatten-IT reduzieren.
- Hierfür muss es den Anwendern so leicht wie möglich gemacht werden eigene Modelle zu erstellen wie z.B. ein Formular zu gestalten bzw. anzupassen.
- In einem Projekt ist es für die Citizen Developer möglich, mithilfe des Editors Seiten und Formulare intuitiv anzupassen, indem der Anwender genau sieht was er aktuell anpasst. Dadurch muss er nicht umdenken und sich vorstellen welche Auswirkungen die eigenen Änderungen besitzen. Anforderungen, die in Form von Excel-Listen und Tickets an die klassischen Entwickler weitergeleitet werden, fallen dadurch weg.

Interview P6

Können Sie sich kurz vorstellen und Ihre Tätigkeit beschreiben?

- Business Analystin, die prüft, ob es Sinn macht Projekte mit A12 umzusetzen oder ob eine individuelle Programmierung besser geeignet ist.

Was ist der Vorteil von A12 gegenüber der klassischen Softwareentwicklung?

- Vorteil ist dass es Formulare als Grundlage gibt die individuell angepasst werden können.
- Super einfach mit A12 Formulare zu erstellen.
- Ein weiterer Vorteil ist, dass A12 mit den Projekten wächst und mit der Zeit immer mehr fertige Funktionen im Angebot hat. Dabei ist allerdings zu prüfen welche zu entwickelnde Funktionen auch für andere Projekte relevant sein können, um diese in das Produkt A12 mit aufzunehmen (Möglichkeiten von change requests).

Welche Vorrausetzungen müssen erfüllt sein, damit A12 erfolgreich eingesetzt werden kann?

- Wichtige Voraussetzung von vornerein zu prüfen, ob es Sinn macht das Projekt mit A12 umzusetzen oder nicht. Dabei ist es relevant zu prüfen wie viele Anforderungen standardmäßig mit A12 umgesetzt werden können. Wenn z.B. 70% der Anforderungen direkt umgesetzt werden können dann macht A12 sinn. Sind es weniger als 50% dann macht es mehr Sinn spezifisch zu entwickeln. Diese Entscheidung wird häufig auf einer hohen Ebene wie dem Sales getroffen. Ich bin als BA auch Berater für den Sales.
- Organisatorisch: Expertengruppe von A12 für jeden Bereich hilfreich, an den sich die Business Analysten wenden können, wenn sie eine spezielle Frage haben. Viele mgm-Mitarbeiter lernen sich gerade in A12 ein und dabei ist es sehr hilfreich, wenn sie bei Fragen wissen an wen sie sich wenden müssen. Oder die Business Analysten erhalten die Info, dass man nichts anpassen muss, sondern dass es schon Standards in A12 gibt.
- Ausführliche A12-Schulungen der Business Analysten sehr wichtig, damit wenige Nachfragen gibt. Somit können bessere Entscheidungen getroffen werden, ob welche Anforderungen standardmäßig in A12 umgesetzt werden können und welche nicht.
- Grundsätzlich versucht man so nahe wie möglich am Standard zu bleiben, damit so wenige individuelle Anpassungen wie möglich notwendig sind. Gegebenenfalls werden dann doch noch Features benötigt, die aktuell noch nicht in A12 verfügbar sind.

Fallen Ihnen weitere Einsatzmöglichkeiten ein, die bisher nicht zum Einsatz kommen?

Im E-Commerce Bereich kann der Einsatz von Low Code Plattformen ebenfalls sinnvoll sein. Hierfür muss aber z.B. die Funktionalitäten von A12 ausgebaut werden mit Internationalisierung z.B. Produktdetailansicht mit Feldern/Bildern und die Features.

Kann die IT-Sicherheit durch den Einsatz von A12 in der Softwareentwicklung erhöht werden (Schatten-IT)?

Der Scope von den meisten Projekten ist so groß, dass es schwer ist alle Funktionen mit Low Code abzudecken und somit die Schatten-IT zu reduzieren. Es ist unrealistisch zu denken, dass eines Tages die Schatten-IT komplett aufgelöst werden kann. Schatten-IT kann allerdings durch A12 reduziert werden.

Interview P7

Wie würdest du deine eigene Tätigkeit bei adesso beschreiben?

- Unterstützt diverse Kunden bei der Auswahl von Low-Code Plattformen, der Implementierung eines Citizen Developer Framework und der Delivery von Low-Code Projekten.

Was ist für dich der größte Vorteil von Low Code Plattformen in der öffentlichen Verwaltung?

- Persönlich ist Low Code ein Ansatz und keine Technologie. Low Code ist ein sehr breiter Begriff worunter z.B. Wordpress fällt oder Power BI für Dashboards.
- Größter Mehrwert ist die Widerverwendbarkeit. Das Rad muss nicht neu erfunden werden sondern wichtig ist erst einmal zu schauen was für Bausteine/Frameworks bereits schon existieren und diese dann abhängig von den eigenen Anforderungen anzupassen. Mit Low Code muss dies nicht mehr mühselig aus dem Internet wie z.B. bei Stack Overflow zusammengesucht werden, sondern die meisten Plattformen bieten out of the box schon viele Frameworks. Somit muss man nicht zum tausendsten Mal TextBoxen mit allen Eigenschaften definieren.
- Durch die Wiederverwendbarkeit wird eine unglaubliche Geschwindigkeit geschaffen und eine echte Standardisierung ermöglicht. Einzelne Module aus unterschiedlichen Projekten werden nach dem gleichen Standards entwickelt.
- Bei High Code bekommt man Anwendungen die stand alone funktionieren, es wird sich nicht darum gekümmert Teile wiederzuverwenden.
- Bei der Entwicklung mit Low Code entstehen automatisch wiederverwendbare Bausteine, die sehr leicht zu strukturieren sind und dann in anderen Projekten eingesetzt werden können.
- Somit wächst ein Werkzeugkasten nicht nur mit den standardmäßig vorhandenen Tools sondern der Werkzeugkasten wird um die eigens entwickelten Module erweitert. Daraus entsteht ein Schneeballsystem das sehr standardisierte Module anbieten, die auf die eigene Firma angepasst sind.

Welche zukünftigen möglichen Anwendungsgebiete sehen Sie beim Einsatz von Low Code Plattformen, die aktuell noch nicht oder selten umgesetzt werden?

- Riesen Potenzial aufgrund der Standardisierung. Dadurch könnte eine KI antrainiert werden, die per Sprachinterface Applikationen baut.
- Da bei Low Code nicht nur User Interfaces entwickelt werden können, sondern auch das komplette Backend integriert ist, wird ist es für KI wesentlich leichter auf die Bausteine zurückzugreifen und diese dann wiederverwenden, um neue Applikationen bauen zu können.

- Viel größere Bausteine, die zusammengefügt werden, sind für eine KI leichter zu verarbeiten als individuelle High Code Anwendungen.
- Der Einsatz von Low Code in der RPA kann kritisch gesehen werden. Besitzt aber enormes Potenzial.
- Weiterer Bereich mit viel Potenzial ist Business Process Management für Fachverfahren, um den kompletten Prozess zu digitalisieren und nicht nur einen Teilbereich. Alle Aufgaben in einem Flow zu sehen und Potenziale und Engpässe in den Prozessen zu ermitteln. Viele Beschwerern sich über zu viel Arbeit aber können es trotzdem in einem Tag erledigen. Diese Sichtbarkeit ermöglicht es auf Kapazitätsengpässe dynamisch zu reagieren.

Welche Voraussetzungen müssen erfüllt sein, um Low Code Plattformen für die Softwareentwicklung in der öffentlichen Verwaltung erfolgreich einzusetzen (Infrastruktur, Expertise, Fachlichkeit)?

- Schwierig zu sagen, da sich die Plattformen und deren Fokus stark unterscheiden.
- Die richtige Low Code Plattform ist sehr wichtig, damit die Fachexperten je nach Usecase sehr schnell eigene Anwendungen entwickeln können.
- Müssen anspruchsvollere Anwendungen entwickelt werden, dann ist eine andere Low Code Plattform geeignet, bei der man gegebenenfalls Programmiererfahrung für die Entwicklung eigener Anwendungen mitbringen muss.

Welche Herausforderungen müssen bewältigt werden, damit Low Code noch häufiger eingesetzt wird?

- Wichtiges Thema dazu ist das Citizen Developer Framework. Ein Ansatz wie man Low Code einführt, indem man die Plattform nicht nur in der IT-Abteilung nutzt. Ohne das Citizen Developer Framework kann das volle Potenzial nicht ausgenutzt werden.
- Governance/Strategie wie man Low Code im Unternehmen einführt sehr wichtig mit Center of Excellence, mit Voraussetzungen wie man Anwendungen in den Fachabteilungen und Center of Excellence entwickelt. Hierfür ist ein gewisses Mindset sehr wichtig.
- Pareto Prinzip ebenfalls sehr wichtig bei der Softwareentwicklung. Ist gerade für Deutschland nicht so einfach zu verinnerlichen nicht alles von Anfang bis Ende durchzuplanen, bevor man mit der Umsetzung beginnt (Mindset Change).
- Ausreichender Reifegrad von Enterprise Low Code Plattformen erst seit ca. 3 Jahren erreicht. Deswegen gibt es aktuell so einen großen Hype. Endlich möglich, dass Entwickler und Fachexperte an einem Tisch sitzen und gemeinsam arbeiten, indem die Anforderungen ermittelt und schnell umgesetzt werden können. Programmierer müssen nicht auf die Definierung der Anforderungen warten ==> großer Mindset change.
- Die Bereitstellung von ersten Prototypen dauert aufgrund von Low Code nicht mehr mehrere Wochen, sondern wenige Stunden. Das Ändert das komplette Requirementengineering.
- Durch die direkte Bereitstellung von Konzepten entsteht Vertrauen und Vertrauen ist bei Projekten sehr wichtig. Je mehr Vertrauen besteht, desto weniger muss dokumentiert werden und desto mehr

wird miteinander geredet und dadurch gewinnt man Geschwindigkeit und senkt die Kosten.

Welche Schritte sind notwendig, um das Potenzial zur Reduzierung der Schatten-IT maximal auszunutzen?

- Die Schatten-IT lässt sich nur vollständig verhindern, wenn alle Mitarbeiter ohne Computer arbeiten. Da dies nicht realistisch ist, wird es immer Hilfslösungen mit Excel, Access oder anderen Tools, entwickelt.
- Die zentrale IT muss akzeptieren, dass IT-Anwendungen auch aus Fachabteilungen heraus entstehen dürfen. Allerdings müssen dabei bestimmte Regeln strikt befolgt werden.
- Die Fachabteilung muss akzeptieren, dass die zentrale IT die Verantwortung für (fast) alle Anwendungen trägt. Regeln der IT-Abteilung sind erforderlich, die von Fachabteilungen eingehalten werden. Die hohe Kunst liegt darin, die Freiheiten und Einschränkungen abzuwägen, um einerseits den Fortschrittsgeist der Mitarbeitenden möglichst stark auszuschöpfen und ihnen andererseits eine sichere Umgebung für solche Innovationen zu schaffen. Hierfür ist ein Citizen Developer Framework erforderlich. Im Unternehmen sollte episodisch ein Center of Excellence (CoE) für Low Code etabliert werden, das Best Practices, Naming Conventions und/oder Security Guidelines definiert. Zusätzlich ist es wichtig im Center of Excellence Know-how aufzubauen, um entscheiden zu können welche Anwendungen in den Fachabteilungen bleiben (zum Beispiel kleine Anwendungen, die keine kritischen Prozesse abbilden, oder Proof of Concepts [PoC]). Zusätzlich gilt es zu klären was durch ein CoE betreut werden muss (zum Beispiel mittelgroße Anwendungen, die keine kritischen nichtfunktionalen Anforderungen haben) und was an die zentrale IT übergeben werden muss.
- Grundsätzlich ist die Überwachung des Applikationsflusses von Citizen Developers zu CoE bis hin zu der zentralen IT die Aufgabe eines Solution Architect oder des CTOs. Durch LCDPs werden alle Applikationen sichtbar. Diese Transparenz ist der Schlüssel zum Erfolg, um die Schatten-IT unter Kontrolle zu bekommen. Die Vielfalt an Anwendungsfällen ist enorm, daher muss die Einführung des Citizen Developer Frameworks stufenweise und zielgerichtet erfolgen.

Welche Nachteile gibt es beim Einsatz von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung?

- Es ist immer eine wichtige Entscheidung zwischen make or buy zu treffen, bei der man sich selbst fragen muss wie individuell muss meine Lösung sein und welche bestehenden Lösungen gibt es. Macht keinen Sinn ein eigenes CRM- oder ERP-System zu bauen. Höchstwahrscheinlich gibt es schon ein passendes System.
- Macht auch wenig Sinn auf Desktopapplikationen zu gehen denn Vorteil besteht in der Webentwicklung.
- Im Bereich Embeded Systems in Naher Zukunft ist der Einsatz von Low Code eher unwahrscheinlich. Es gibt für High Code Entwicklung noch genug zu tun für große, komplexe Anwendungen können dann die Senior Entwickler ihre volle Zeit reinstecken.

Interview P8

Wie würdest du deine eigene Tätigkeit bei hansevision beschreiben?

- Tätig im Kompetenzzentrum Low Code bei der hansevision. Kundenberatung zur Bestimmung der passenden Low Code Plattform je nach Bedürfnissen des Kunden.

Was ist für dich der größte Vorteil von Low Code Plattformen?

- Ein riesiger Vorteil von Low Code Plattformen besteht darin, dass sich der Anwender um die Systeme darunter nicht mehr kümmern muss.
- Bei der klassischen Programmierung muss ich bei einer neuen Version des Endcode oder professional Code schauen ob die Kommunikation zu den Datenbanken und Betriebssystemen weiterhin funktioniert und darüber muss ich mich als Kunde oder Anwender bei Low Code Plattformen nicht kümmern.
- Der Kunde kann sich somit um die Fachlichkeit kümmern.

Welche zukünftigen möglichen Anwendungsgebiete sehen Sie beim Einsatz von Low Code Plattformen, die aktuell noch nicht oder selten umgesetzt werden

- Public Sector ist ein Bereich mit einem hohen Nachholbedarf, da schnell eine erste Lösung in den Go Live Betrieb überführt werden kann.
- Modularer Aufbau von Low Code Plattformen ermöglicht es neue Funktionen zu entwickeln, ohne den genauen Aufbau zu kennen und ohne zu wissen in welchen Klassen Änderungen vorgenommen werden müssen.

Welche Voraussetzungen müssen erfüllt sein, um Low Code Plattformen für die Softwareentwicklung in der öffentlichen Verwaltung erfolgreich einzusetzen (Infrastruktur, Expertise, Fachlichkeit)?

- Commitment von allen Beteiligten und Vorgesetzten in Bezug auf Zeit ==> läuft nicht einfach so nebenher.
- Richtige Zeitabschätzung der Power User.
- Wenn Personen für ein Projekt brennen, dann funktioniert es immer. Wenn es solch eine Person nicht gibt dann funktioniert selbst das kleinste Projekt nicht.
- Bei der Entscheidung für eine Low Code Plattform, sollte der Zeithorizont 4-10 Jahre betragen.

Bei welcher Art von Software/Anwendungen sind Low Code Plattformen besonders geeignet?

- Für Anwendungen und Workflow-Prozessen gibt es passende Low Code Plattformen.
- Bei der Anwendungsentwicklung (Webanwendungen, mobile Anwendungen, Dashboard, Websiten) wird Low Code der Standard werden.
- Bei Embeded Software bei denen man im Millisekundenbereich unterwegs ist wie z.B. Maschinensteuerung wird die klassische Programmierung weiterhin der Standard bleiben.

Welche Herausforderungen müssen bewältigt werden, damit Low Code noch häufiger eingesetzt wird?

- Konsolidierung von dem stark wachsenden Low-Code-Markt. Jeder schwemmt auf den Markt mit dem Buzzword im Gepäck und die Anwender wissen gar nicht was sich dahinter verbirgt.
- Analysten bringen Licht in das dunkle und unterteilen den Oberbegriff Low Code in Unterkategorien wie RPA oder Mulit Experience Plattform damit es am Ende besser zugordnet werden kann.

Welche Voraussetzungen müssen erfüllt sein, um Potenzial zur Reduzierung der Schatte-IT mithilfe von Low Code Plattformen voll auszuschöpfen?

- Ganzheitliche Betrachtung der Anforderungen und Auftragserfüllung. Schatten-IT entsteht, wenn Mitarbeiter eine Aufgabe übertragen bekommen die sie irgendwie erledigen müssen ohne ein Tool vorzugeben. Denn dann nehmen sie das was schon vorhanden ist wie bspw. Excel oder Access.
- Nur wenn ich mich um die Belange der Mitarbeiter kümmere und sage hier habt ihr die Aufgabe und frage was benötigt ihr, um diese Aufgabe zu erfüllen, wie können wir das in unsere Infrastruktur einbringen? Sonst wird sich die Schatten-IT nicht verhindern lassen.
- Organisatorisches herangehen grundsätzlich wichtig. Wenn etwas gebraucht wird, wie z.B. eine spezielle Aufgabenverwaltung die es so nicht gibt dann muss dies aufgenommen werden und eine Lösung gefunden werden, die in das Gesamtkonzept passt.
- Wenn der Vorgesetzte sagt "mach wie du magst Hauptsache um 5 bist du fertig" dann wird die Schatten-IT nicht verschwinden.
- Kommunikation und Bewusstsein muss immer da sein und ist die Grundvoraussetzung.

Wird ein IT-Affinität für die Nutzung von Low Code Plattformen vorausgesetzt?

- Hängt sehr stark von der Low Code Plattform ab. Je nachdem welche Zielgruppe durch die Low Code Plattform angesprochen wird. Bei Plattformen, die es jedem versuchen zu ermöglichen eigene Anwendungen zu entwickeln haben ein sehr niedrigschwelligen Einstieg und setzen somit eine geringere Affinität voraus.
- Bei Plattformen mit stärkerer Spezialisierung, die für einen bestimmten Zweck entwickelt wurden bzw. daraus entstanden sind ist der Einstieg mit einem größeren Aufwand verbunden.

Welche Nachteile gibt es beim Einsatz von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung?

- Für betriebssystemnahe Anwendungen ist Low Code eher ungeeignet. Wenn z.B. bei der Autosteuerung Low Code Plattformen eingesetzt werden würde, dann würde ich mich persönlich nicht so sicher fühlen.
- Da sollte der Code ganz sauber ohne overhead geprüft und sehr zielgerichtet sein. Versuch der Low Code Plattform zu standardisieren ist in diesem Fall vielleicht nicht ganz so gut geeignet.
- Je sicherer die Anwendung sein muss desto wichtiger ist eine individuelle Lösung.

Interview P9

Was ist der Vorteil von A12 gegenüber der klassischen Softwareentwicklung?

- Vorteil von A12 liegt darin, dass die Modellierung von Fachexperten durchgeführt werden kann und sich die mgm-Mitarbeiter nicht mit der Fachlichkeit beschäftigen müssen.
- Lotta Projekt: Digitalisierung der Investmentsteuer.
- Die Investmentsteuer ist sehr komplex und die Fachexperte mit dem Fachwissen modellieren alles was sie benötigen wie z.B. Formulare, eigene Validierungen, Prüfrechnungen usw.
- Der Sachbearbeiter kann eigene Änderungen in dem Formular durchführen.
- Modelle müssen nur einmal modelliert werden und können an verschiedenen Stellen wiederverwendet werden.
- Viele Funktionalitäten für das Projekt waren schon mit A12 out of the box vorhanden. Somit musste nicht komplett von null begonnen werden.

Welche Vorrausetzungen müssen erfüllt sein, damit A12 erfolgreich eingesetzt werden kann?

- Modellierer müssen Schulungen mit A12 durchgeführt haben und sich sehr gut mit dem Programm insgesamt auskennen. Außerdem müssen sie auf dem neuesten Stand bleiben, wenn neue Funktionen releast werden.
- Regelmäßiges Upgrade von A12 notwendig.
- Anforderungen des Kunden prüfen und schauen ob diese Funktionen auch für andere Projekte relevant sein können bzw. von der Funktion profitiert. Wenn die Antwort ja ist dann soll die Funktion in das Produkt hineinfließen.

Fallen Ihnen weitere Einsatzmöglichkeiten ein, die bisher nicht zum Einsatz kommen?

- Fachexperten modellieren noch nicht die Benutzeroberfläche, was wünschenswert wäre.
- Mit Toko gestalten Fachexperten schon die Benutzeroberfläche. Vielleicht ist dies auch bei A12 möglich.

Kann die IT-Sicherheit durch den Einsatz von A12 in der Softwareentwicklung erhöht werden?

- Anwendungen, die mit A12 entwickelt werden können leichter von einem Kollegen übernommen werden der die Anwendung nicht selbst entwickelt hat, da der Aufbau aller Anwendungen sich ähneln und die Struktur die gleiche ist und alles dokumentiert wird.

Welchen Nachteil besitzt A12 gegenüber der klassischen Softwareentwicklung?

- Mit High Code manchmal schneller zu programmieren da es mit A12 generalisierbar sein muss.

Interview P10

Kannst du dich kurz vorstellen und deine Rolle im Modul-F Projekt erklären?

- Ich bin in der Staatskanzlei Hamburg im Bereich IT und Digitalisierung im Referat Steuerung von Fachverfahren und neue Technologien tätig. Komplett zuständig für das Projekt Modul-F.

Wie soll A12 bei Modul-F eingesetzt werden?

- Die A12 Plattform als Basis, um die Funktionalitäten die den Kern von Modul-F ausmachen sollen zu erstellen.
- Dies geschieht in Form von Editoren in denen verwaltungsspezifische Module zusammengefügt werden können.
- Module bestehen aus den Modellen von A12 die mithilfe des Editors zusammengesetzt werden können.

Welchen Vorteil erhoffst du dir durch den Einsatz von Low Code im Modul-F-Projekt?

- Grundsätzlich besteht ein Bedarf in der Verwaltung nach Softwareanwendungen. Es ist aber unwirtschaftlich aufgrund des Bedarfs ein Softwareprojekt einzuleiten, da es sehr selten gebraucht wird (In der Verwaltung gibt es viele Fachverfahren, die selten zum Einsatz kommen). Mit Low Code kann der Prozess erleichtert bzw. beschleunigt werden.
- Es gibt eine Vielzahl an Bedarfe. An ganz unterschiedlichen Stellen in der Verwaltung besteht ein riesen Loch an Softwareunterstützung (Fachkräftemangel).
- Bedarf nicht nur für Hamburg sondern auch bei anderen Bundesländern ==> Mithilfe von Low Code ist es möglich das Einer-für-allle-Prinzip leichter umzusetzen, da individuelle Präferenzen leichter umgesetzt werden können.
- Wir sind nach einer Analyse zu dem Entschluss gekommen, dass es keinen Sinn macht eine eigene Low Code Plattform zu entwickeln oder komplett klassisch zu entwickeln, da das Ziel von Modul-F schon von Beginn ist schnell voran zu kommen und es somit zeitlich nicht realistisch gewesen wäre und auch nicht wirtschaftlich eine eigene Plattform zu entwickeln bzw. den Weg der klassischen Softwareentwicklung zu gehen.
- Der Vorteil von A12 gegenüber anderen Low Code Plattformen ist die Kooperation von verschiedenen Komponenten, die wir nutzen können, um sie weiterzuentwickeln. Bei anderen Anbietern war es deutlich monolitischer wo es schwerer ist in den Code einzusehen bzw. anzupassen.
- Bei A12 gibt es mehr Möglichkeiten der individuellen Anpassung und mehr Möglichkeiten bei der Softwareentwicklung.
- Funktionalitäten, die für viele Projekte benötigt werden wie z.B. das 4-Augen-Prinzip, werden in vorgedachten Modulen standardmäßig dem User zur Verfügung gestellt.
- Wir haben das Ziel, dass die Fachverfahrensentwickler sich mit No-Code die benötigten Fachverfahren zusammenklicken können. Allerdings gibt es unserer Meinung nach keine passende Lösung. Daher sind wir zu dem Entschluss gekommen das ganze selbst zu programmieren und dafür ist A12 unserer Meinung am besten geeignet.

Wie verändert sich dadurch die Arbeit der klassischen Entwickler im Vergleich zu ohne Low Code?

- Der klassische Entwickler entwickelt die Module in A12. Ist ein Stück Fachlichkeit und das Technische.
- Der Fachverfahrensentwickler bedient den Editor und muss den Prozess mit den Modulen abbilden. Somit näher bei der Fachlichkeit.

Welche Voraussetzungen müssen erfüllt sein, damit A12 erfolgreich eingesetzt werden kann z.B. Schulung der Mitarbeiter oder Abschätzung der Kapazitäten des Projektleiters?

- Die Voraussetzung für Modul-F ist, dass es eine ausreichende Anzahl an Modulen gibt, die die Bedarfe der Verwaltung erfüllen.
- Wichtig, dass die Erstellung der Module relativ aufwandsarm möglich ist.
- Standardmäßige Verfügbarkeit von ansprechenden Oberflächen (Templates) ist als Grundlage sehr wichtig, damit die Anwendungen am Ende leicht verständlich sind und ansprechend aussehen. Für den User muss es am Ende möglich sein, sich leicht zu orientieren, indem alles selbsterklärend bzw. barrierefrei ist.

Was sind für dich die größten Herausforderungen bei dem Einsatz von A12 bei Modul-F?

- Herausforderungen der Finanzierung und die richtige Lösung am Markt und das Personal zu finden.
- Herausforderung ob die gewünschten Funktionalitäten mit der existierender Basis umgesetzt werden können und wie viel manuelle Programmierung notwendig ist.

Reduzierung Schatten-IT

- Dafür müssen die problematischen Prozesse vollkommen mit A12 abgebildet werden.
- Es ist aufwendig die Behörden dazu zu bringen, sich die Prozesse genau anzuschauen, um dann das Ganze auch abbilden zu können, wenn dies nicht alles einer Software umsetzbar ist.
- Der Verwaltung eine Software anzubieten die sie nutzen dürfen, um alle Regularien einzuhalten wie z.B. Barrierefreiheit.

Nachteile von Low Code Plattformen/A12?

- Limitierungen die mit den Modellen von A12 kommen. Hat Vorteile und Nachteile mit solchen Modellen zu arbeiten, da gewisse Funktionalitäten direkt genutzt werden können aber individuelle Anforderungen aufwendiger umzusetzen sind als bei der klassischen Entwicklung.
- Komplexität bzw. Möglichkeiten der Tools von mgm zu vielseitig für unseren Anwendungsfall. Für unser Szenario noch mehr Einfachheit wichtig. Diese Einfachheit schaffen wir nun selbst, da wir die Komplexität von A12 nicht in vollem Umfang benötigen.
- Herausforderung zu schauen, wie die bestehenden Prozesse/Workflows mit den Modulen von A12 abgedeckt/nachgestellt werden können. Wichtig die Fachlichkeit in die Modelle zu integrieren bzw. reinzupressen.
- Möglichkeiten der Software zur Modellierung von z.B. Workflows nicht so vielseitig, wie die reellen Prozesse in der Verwaltung.

Interview P11

Können Sie sich kurz vorstellen und Ihre Tätigkeit beschreiben?

- Im Landesamt für Steuern in München tätig in der Elster Entwicklung bin fachlicher Chefarchitekt.
- Bindeglied zwischen Verfahrensmanagement und der Gesamtsteuerung aller Projekte

Wie wird A12 bzw. Toko in deinem Projekt eingesetzt?

- A12 wird in Form von Toko und Tukan in verschiedenen Projekten eingesetzt.
- Bei Tukan geht es um die Modellierung von Formularen. Die Fachlichkeit der Steuerformulare wird vom Bundesfinanzministerium modelliert.
- Bei Toko geht es darum, aus den fachlichen Vorgaben, die aus Tukan erfasst werden eine Oberfläche in myElster zu bauen.

Wie hat sich deine Tätigkeit durch die Modellierung mit Toko und Tukan verändert?

- Früher wurden Formulare manuell mit dem PDF-Editor erstellt und dann im Anschluss abgeschrieben. Diesen Medienbruch gibt es heute nicht mehr.
- Durch den Einsatz von A12 konnte die Modellierung die normalerweise 5 Monate dauert um ca. 6 Wochen verkürzt werden.
- Ein großer Vorteil ist, dass die Mitarbeiter ihre Kompetenzen besser einbringen können und sich nicht mehr so viel mit Fleißarbeiten beschäftigen müssen.
- Insgesamt können durch den Einsatz von A12 sich die Mitarbeiter der Finanzbehörde früher mit den fachlichen Themen beschäftigen.

Wird deiner Meinung nach IT-Kenntnisse oder eine IT-Affinität für die Nutzung vorausgesetzt?

- Grundsätzlich müssen die Mitarbeiter offen und neugierig für neues sein.
- Bei Toko ist eine Interesse für UI/UX wichtig. Allerdings sind keine IT-Vorkenntnisse oder IT-Studium erforderlich.
- Bei Tukan sind keine Kenntnisse erforderlich, da die Bedienung sehr einfach ist und von allen Mitarbeiter der Finanzbehörde genutzt werden soll.

Gibt es für dich Voraussetzungen die erfüllt sein müssen, um A12 erfolgreiche einzusetzen? (Schulungen, Fähigkeiten der Projektbeteiligten)

- Grundsätzlich ist es sehr hilfreich, wenn die Entwickler und Ansprechpartner für die Behörde ein Verständnis für die Abläufe der öffentlichen Verwaltung mitbringen. Je höher die Hierarchie desto komplizierter ist die Struktur.
- Die Mitarbeiter in der öffentlichen Verwaltung müssen die Veränderung akzeptieren bzw. positiv mitgestalten wollen und sich nicht dagegenstellen.
- Grundsätzlich ist es auf jeden Fall hilfreich die kritischen Mitarbeiter noch stärker fachlich abzuholen und die persönlichen Vorteile bei der Anwendung noch klarer herauszustellen.

Gibt es Verbesserungen die Sie sich wünschen?

- Ein strukturelles Problem, das schwer zu verbessern ist, ist die Abhängigkeit vom Gesetzgeber. In der Regel werden gesetzliche Änderungen in 7 Monaten umgesetzt. Nur werden kurzfristige Änderungen von Seiten des Gesetzgebers immer häufiger und sind dementsprechend schwer umzusetzen. Mit dem Einsatz von Low Code ist es überhaupt möglich die Änderungen schneller durchzuführen.
- Grundsätzlich muss die Bereitschaft vorhanden sein Prozesse neu zu denken. Mithilfe von Low Code Plattformen bzw. A12 Vordrucke und Anpassungen iterativ und somit agil durchgeführt werden. Die Bearbeitung erfolgt immer noch in statischen Sitzungen, die in einem vordefinierten Turnus stattfinden.

Reduzierung Schatten-IT?

- Teilweise kann die Schatten-IT abgelöst werden. Allerdings gibt es noch ein riesiges Potenzial weitere Schatten-IT abzulösen.
- Intern gibt es eine klare Trennung zwischen Elster und den internen Fachverfahren, die eine weitere Reduzierung der Schatten-IT verhindert. Diese Trennung gilt es nach dem Vorbild der OZG-Plattform aufzulösen.

Interview P12

Was ist für dich der größte Vorteil von Low Code Plattformen in der öffentlichen Verwaltung bzw. welche Probleme können gelöst werden?

- Der Vorteil von Low Code ist, dass man sehr schnell Ergebnisse erzielen kann
- Der Einarbeitungsaufwand und das Verständnis ist bei klassischen Entwicklern deutlich kürzer als bei Fachexperten, da die meisten Low Code Plattformen nicht so intuitiv sind wie vom Hersteller angegeben.
- Der größte Vorteil besteht für mich in der schnellen Entwicklung.
- Besonders ist auch, dass ich alles vollintegrierbar an einem Ort habe und ich nicht das ganze extra deplployen muss. Außerdem gibt es immer einen stabilen Code der läuft.

Welche Nachteile gibt es beim Einsatz von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung?

- Lizenzen können zu einem Problem werden, da dadurch die laufenden Kosten höher sind als bei der klassischen Entwicklung.
- Grundsätzlich ist die Softwareinitiierung mit Low Code günstiger aber die laufenden Kosten höher
- Einige Low Code Anbieter bieten keine On-Premise-Lösung an, was bei Kunden teilweise Voraussetzung ist.
- Durch die Nutzung einer Low Code Plattform ist es schwieriger zu prüfen, ob die eigenen Sicherheitsstandards eingehalten werden. Daher ist es auch schwieriger eine Fehlersuche zu betreiben und man ist abhängig von den Sicherheitspatches des Herstellers.

Interview P13

Bei welcher Art von Software/Anwendungen sind Low Code Plattformen besonders geeignet? Wann ist es eher unpassend?

- A12 bietet sich bei komplexen Anwendungen an, die in eine komplexe IT-Infrastruktur integriert werden muss. Eher nicht geeignet für eine App oder ein in sich geschlossenes System.
- Alles fachliche lässt sich konfigurativ durch die Editoren anpassen und erfordert keine Entwicklung. Es treten regelmäßig komplexe Änderungen bei der Steuerbehörde auf.

In wie weit müssen sich Low Code Plattformen weiterentwickeln, damit sich das Einsatzgebiet in der öffentlichen Verwaltung erweitert?

- A12 ist eine offene Low Code Plattform, bei der es viele Konfigurationsmöglichkeiten gibt.
- Die Anwenderfreundlichkeit von A12 kann gesteigert werden, wenn die einfache Gesamtnutzung verbessert wird, damit keine Entwickler mehr benötigt werden, um ein Projekt aufzusetzen. Fachanwender kann eigenständig einfache Anwendung konfigurieren, ohne auf den Entwickler angewiesen zu sein.

Welche Nachteile gibt es beim Einsatz von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung?

- Wenn man zu Beginn des Projektes nicht alle Anforderungen an die Software kennt und man noch nicht ganz genau weiß was man möchte, dann kann der Einsatz einer Low Code Plattform zu Problemen führen. Während der Konfiguration kann die Erkenntnis auftreten, dass die eigenen Anforderungen mit den Funktionen der Low Code Plattform nicht abgedeckt werden können. Die Zeit ist verschwendet, wenn die Low Code Plattform eine Black Box ist, bei der man nur die standardmäßig verfügbaren Funktionen nutzen kann und keine eigene Entwicklung möglich ist bspw., wenn kein Drittsystem angebunden werden kann oder bestimmte Daten nicht erfasst werden können. Im schlimmsten Fall muss bei der Softwareentwicklung noch einmal von vorne begonnen werden.

Welcher Nachteil besteht bei der Softwareentwicklung mit A12 gegenüber der klassischen Softwareentwicklung?

- Bei der Entwicklung von kleinen Tools bzw. Anwendungen dauert die Softwareentwicklung mit A12 länger als bei anderen Low Code Plattformen oder länger als bei der klassischen Softwareentwicklung.
- A12 ist eine zu große Lösung für ein kleines Problem.

Interview P14

Was sind für dich persönlich die größten Vorteile von Low Code Plattformen?

- Größter Vorteil ist die modellbasierte Softwareentwicklung.
- Mithilfe der Modelle können Fachexperten die eigene Fachlichkeit selbst erfassen.
- Reduzierung der Fertigungstiefe in der Softwareentwicklung. Das bedeutet, dass man nicht immer von null starten muss und vieles Vorkonfigurierte nutzen kann. Als Beispiel ist der Aufwand sehr hoch die Schnittstellen (APIs) von A12 gegen Cyberangriffe zu sichern.

Securtiy Experten untersuchen die Schnittstellen und gewährleisten die Sicherheit.
- Ohne eine Plattform werden die Aufwände der Securtiy Experten bei jedem Projekt aufs neue benötigt, um die Sicherheit zu gewährleisten.

Können Low Code Plattformen die Schatten-IT reduzieren? Und wenn ja was sind die Voraussetzungen?
- Kommt grundsätzlich auf die Funktionsfähigkeit und Leistungsfähigkeit der Low Code Plattform an.
- Jede Software eines Unternehmen das die Fachlichkeit oder Fachanwendungen abbildet muss in eine Anwendungslandschaft integriert werden. Kein großer Vorteil wenn ein Excel-Sheet durch eine Low Code Lösung abgelöst wird, die in sich geschlossen ist. Somit nicht wirklich was geschafft da die Datenintegration fehlt.
- Datenaustausch ist notwendig, damit die Daten nicht in eine Excel exportiert und an anderer Stelle wieder importiert werden. Dadurch gibt es trotzdem Insellösungen.
- Wichtig ist dass die Low Code Plattform von vornherein dafür geschaffen ist, die einzelnen entwickelten Anwendungen in die Low Code Plattform zu integrieren.
- Ziel ist eine Anwendungslandschaft mit automatischem Datenaustausch. Damit die Vorteile der Low Code Plattform genutzt werden können.

Welche Herausforderungen in der öffentlichen Verwaltung können durch Low Code gelöst werden? Und wie lassen sich diese lösen?
- A12 zeigt, dass eine der komplexesten Domänen die Steuerlogik modular angegangen werden kann. Immense Bedeutung für den öffentlichen Bereich.
- Ziel kann es sein unterschiedliche Institutionen eine Plattform zur Verfügung zu stellen, auf der die IT-Dienstleister ihre eigene Lösungen auf Basis der Plattform neu aufbauen.
- Abteilung mit der Gesetzgebung vernetzen. Während der Gesetzgebung kann die Abteilung schon überlegen, wie das ganze digital umgesetzt werden kann.
- Lösungen die auf einer Low Code Plattform aufbauen sind auf andere Verfahrensarten übertragbar wie z.B. dass ein Bescheid automatisch verschickt wird.

Zukünftige Anwendungsgebiete von Low Code Plattformen
- Geht eher darum die Softwareentwicklungsprozesse noch stärker zu unterstützen.
- Möglichkeiten verbessern schneller erste Prototypen der Software zu testen und zu sehen wie sieht meine Produktionsumgebung aus? Wie sieht meine Test und Entwicklungsumgebung aus?
- Für breiteren Einsatz von Low Code Plattformen soll im Idealfall die Möglichkeit bestehen, all diese Prozesse zu optimieren und zu schulen.
- Sonst passiert die Optimierung nur im Rahmen von Großprojekten und man braucht ein Großprojekt, um die Qualität und Umgebung verfügbar zu machen. Durch die Optimierung der Low Code Plattform kann die ganze Produktionsstraße vorgedacht werden
- Zusätzlich Vielfalt in Aspekte der Anwendung. Wie persistiere ich Daten? Produktionsstraße optimieren.

Nachteile von Low Code Plattformen?

- Wenn die Plattform dort eingesetzt wird wofür sie nicht geschaffen ist bspw. Im Data Science Bereich wo schnell Daten aufbereitet werden und daher Low Code Plattformen ungeeignet ist.
- Bei hardwarenaher Programmierung ist Low Code auch nicht geeignet.
- Muss sich auf das konzentrieren wofür es geschaffen wurde und das sind meistens Fachanwendungen.

A12 wird meines Wissens hauptsächlich zur Formularmodellierung eingesetzt. Welche Schritte oder vlt. auch Funktionen sind notwendig, um dies zu erreichen?

- Immer eine Abwägungssache. Je weniger konfigurierbar eine Plattform gestaltet ist desto weniger Training benötigt man. Gleichzeitig gibt es eine größere Einschränkung der Einsatzmöglichkeiten.
- Daher immer eine Abwägungssache wie viel Konfiguration möchte ich erlauben wie einfach sollte die Einarbeitung sein. Werden von den Business Analysten dazu gedrängt die Möglichkeiten von A12 zu erweitern.
- Verschiedene Low Code Plattformen konzentrieren sich auf unterschiedliche Anwendungsgebiete.

Bei welchen Projekten sich die Verwendung und A12 lohnt und bei welchen es unvorteilhaft ist?

- A12 ist nicht für kleine Anwendungen geeignet sondern spezialisiert auf Enterprise Anwendungen.

Aktuelle Herausforderungen von A12 die es zu bewältigen gibt?

- Bei A12 ist aktuell eine Herausforderung die Persistenzmöglichkeiten zu verbessern. Die Möglichkeiten zum persistieren von Daten verbessern. Reicht nicht für alle Konstellationen die in der fachlichen Welt auftreten. Persistieren: Wie speichert man Daten ab.
- Aktuell muss man zu schnell in die Individualentwicklung abbiegen.
- Aktuell nur technische Antworten auf Integration mit Umsystemen und das ganze Modulbasiert umsetzen.
- Kollaboration spielt immer mehr eine Rolle. Wichtig bei einer erstellten Fachanwendung sicherstellen, dass ich mit dem Bürger kommunizieren kann. Auch möglich moderne Kanäle wie Chat oder Videotelefonie anbieten mit Einhaltung des Datenschutzes.
- Neben den Modellieren und Enterprise-Entwicklern eine mittlere Ebene dazwischen zu finden um auch an Anwendungsentwickler herantreten zu können.
- Verwendung von komplexe Abläufe andere Prototyping-Tools langfristiges Ziel die Visualisierung in A12 möglich.

Langfristig Low Code durch No Code abgelöst?

- Aus der Erfahrung kann gesagt werden, dass 80% standardmäßig umgesetzt werden kann und 20% individuell. Sind Dinge die andere

Unternehmen nicht brauchen können. Bei einem geschlossenen System erheblich aufwendiger die 20% umzusetzen.

Interview P15

Welche Vorteile bieten Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung?

- Neue Ideen umsetzen, die Wert generieren. Gewisse Basis vorhanden und man kann selbst etwas neues ausprobieren. Keine aufwendige Entwicklungsumgebung jedes mal aufs neue bereitstellen um etwas auszuprobieren.
- Geschwindigkeit der Entwicklung nimmt zu in Bezug auf die Zeit.
- Standardisierung durch Low Code. Skalierung mit gleichzeitiger Wartung.
- Schatten-IT kann durch Low Code reduziert werden. Passendes Framework wird geschaffen.
- Möglichkeit die Kernprozesse wie die Schadensermittlung mit eigens entwickelten Anwendungen zu erweitern. Muss nicht alles in dem Kernprozess direkt entwickelt werden - was häufig sehr aufwendig ist.

Welche Voraussetzungen müssen erfüllt sein, um Low Code Plattformen für die Softwareentwicklung in der öffentlichen Verwaltung erfolgreich einzusetzen (Infrastruktur, Expertise, Fachlichkeit)?

- Man kann nicht davon ausgehen, dass jeder Projektbeteiligte begeistert von IT ist. Vorfeld gemeinsames Verständnis mit dem Kunden zu schaffen was mit Low Code möglich ist. Anwender müssen begeistert werden. Brauche die Menschen die es dann umsetzen und diese von den Vorteilen zu überzeugen.
- Grundverständnis wie Datenbanken und Software funktioniert ist auch eine Voraussetzung
- Bewusstsein für den Change Prozess schaffen und welche Arbeitsschritte damit einherkommen.

Welche Nachteile gibt es beim Einsatz von Low Code Plattformen bei der Softwareentwicklung in der öffentlichen Verwaltung?

- Umfangreiche Kernprozesse, die schon seit vielen Jahren bestehen sind mit den aktuellen Möglichkeiten von Low Code Plattformen nicht abzubilden. Dort lohnt es sich weiterhin klassisch zu entwickeln.
- Schwer langfristig abzuschätzen bei der Anschaffung von Low Code Plattformen welche Kosten und welcher Aufwand auf das Unternehmen zukommt und ob sich tatsächlich der Einsatz der Low Code Plattform finanziell lohnt. Intransparenz von Kosten für bestimmte Anwendungsfälle.
- Limitierung der Möglichkeiten der Low Code Plattform wie z.B. wo kann ich meine Daten speichern.

Welche Schritte sind notwendig, um das Einsatzgebiet von Low Code Plattformen in der öffentlichen Verwaltung zu erweitern?

- Positionierung und Vermarktung was mit Low Code Plattformen möglich ist kann noch verbessert werden, da die meisten noch gar nicht wissen welche Möglichkeiten ich mit Low Code Plattformen habe.