



UAG DIGITALISIERUNG

WHITEPAPER  
ZUM EINSATZ VON LOW-CODE-SYSTEMEN  
IM ÖFFENTLICHEN GESUNDHEITSDIENST

**Projekt:**

Inhaltliche Erweiterung der Geschäftsstelle  
der UAG Digitalisierung der AG ÖGD der AOLG

**Projektnummer:**

BE-P512LM20E51

**Programm:**

Förderung von Maßnahmen zur Steigerung  
und Weiterentwicklung des digitalen Reifegrades  
des Öffentlichen Gesundheitsdienstes in  
Deutschland



Funded by  
the European Union  
NextGenerationEU



ELFA  
EIN LAND FÜR ALLE

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	4
1.1	Zielsetzung des Whitepaper	4
1.2	Relevanz des Themas für den ÖGD	4
1.2.1	Demographischer Wandel	4
1.2.2	Komplexität der Anwendungslandschaft	5
<b>2</b>	<b>Grundlagen zu Low-Code-Plattformen</b>	6
2.1	Begriffserklärung und theoretische Grundlagen	6
2.1.1	Vorteile von Low-Code-Anwendungen	6
2.1.2	Einsatzzwecke	7
2.1.3	Vorgehen in der herkömmlichen Softwareentwicklung	8
2.1.4	Schnellvergleich	8
<b>3</b>	<b>Einsatzszenarien (Use Cases) und Best Practices</b>	9
3.1	Einsatzszenarien im ÖGD	9
3.1.1	Vorgehen bei den Interviews	9
3.1.2	Anwendungsfall Infektionsschutz	9
3.1.3	Anwendungsfall Hygienekontrollen	10
3.1.4	Anwendungsfall Beseitigung von Medienbrüchen	10
3.1.5	Anwendungsfall Schulungen	11
3.1.6	Hauptprobleme von Interviewpartnern	11
3.2	Best Practices	12
3.2.1	Success Story Intrexx Allianz	12
<b>4</b>	<b>Herausforderungen und Rahmenbedingungen</b>	14
4.1	Herausforderungen beim Einsatz von Low-Code-Plattformen	14
4.2	Zusätzliche Aufwände durch unzureichende Prozessdokumentation	14
4.3	Datenschutz und IT-Sicherheit	14
4.4	Aufbewahrung und Löschung	15
4.5	Personalmitbestimmung (Beteiligung)	15
4.6	Betrieb und Verwaltung von Low-Code-Anwendungen	16

4.6.1	Wildwuchs verhindern	16
4.6.2	Regeln für Änderungen an Fachverfahren	16
4.6.3	Organisation und Best Practices	16
4.7	Integration in die bestehende IT-Landschaft	17
4.8	Anpassbarkeit und Flexibilität	17
4.9	Skalierbarkeit	17
4.10	Akzeptanz und Schulung der Mitarbeiter	17
4.11	Lieferantenabhängigkeit	18
4.12	Rechtliche Rahmenbedingungen und Compliance	19
5	Leitfaden für die Einführung von Low-Code-Plattformen	20
5.1	Planung	20
5.1.1	Vorbereitungsphase	21
5.1.2	Beschaffungsphase	22
5.1.3	Implementierungsphase	23
5.2	Auswahl der richtigen Plattform	24
5.2.1	Bewertungskriterien bei der Auswahl der richtigen Low-Code-Plattform	25
5.3	Implementierung (Anwendungsentwicklung)	30
5.3.1	Anforderungen bei der Anwendungsentwicklung	30
6	Zusammenfassung der Erkenntnisse (FAZIT)	33
7	Anhang	35
7.1	Quellen	35
7.2	Glossar	36
	Impressum	40

## 1.1 Zielsetzung des Whitepaper

Dieses Whitepaper zeigt praxisorientiert auf, wie Low-Code-Plattformen den ÖGD unterstützen können. Gleichzeitig beleuchtet das Papier auch die Herausforderungen, die mit der Einführung von Low-Code-Plattformen einhergehen und gibt Handlungsempfehlungen, um diese zu bewältigen. Ziel ist es, Gesundheitsämtern und weiteren gesundheitsbezogenen Verwaltungen in Deutschland eine klare Orientierung zu bieten, die es ihnen ermöglicht, Digitalisierungsvorhaben erfolgreich umzusetzen.

Low-Code-Plattformen unterscheiden sich von der klassischen Softwareentwicklung durch die Nutzung vorgefertigter Bausteine und Werkzeuge, die den Entwicklungsprozess für Nicht-Programmier-Experten einfacher gestalten. Mit Low-Code-Plattformen können Anwendungen erstellt und angepasst werden, ohne dass umfassende Programmierkenntnisse erforderlich sind. Mitarbeitende außerhalb der IT können so selbst Lösungen entwickeln. Die Abhängigkeit von IT-Fachkräften innerhalb der Verwaltung bzw. von externen Dienstleistern wird verringert.

Dieses Papier soll im Rahmen der Digitalisierung als Entscheidungsgrundlage zur Bewertung der Einführung von Low-Code-Plattformen fungieren. Es soll insbesondere helfen, die Frage nach „Make or Buy“, d. h. nach der Beschaffung von Individualsoftware oder der Beschaffung von Standardlösungen, im Kontext einer strategischen Entscheidung zu beantworten, die die digitale Transformation der gesamten Organisation langfristig prägt. In beiden Fällen könnten Low-Code-Plattformen die Antwort sein. Es dient als praxisnaher Leitfaden um Low-Code-Plattformen besser zu verstehen, ihre Potenziale zu erkennen und gleichzeitig Herausforderungen realistisch einschätzen zu können. Um die Anwendbarkeit von Low-Code-Plattformen im Kontext der öffentlichen Gesundheitsdienste besser zu verstehen, wurden Interviews mit Bedarfstragenden aus verschiedenen Bereichen des ÖGD durchgeführt. Ziel dieser Gespräche war es, konkrete Anwendungsfälle (Use Cases) zu identifizieren, in denen Low-Code-Plattformen einen Mehrwert bieten können.

## 1.2 Relevanz des Themas für den ÖGD

### 1.2.1 Demographischer Wandel

Die Struktur der in Deutschland lebenden Bevölkerung verändert sich im Hinblick auf die Altersstruktur, die kulturelle Zusammensetzung und die Verteilung der Bevölkerung zwischen Regionen bzw. Stadt und Land. Der demographische Wandel hat somit Einfluss auf die Leistungen und Aufgaben im öffentlichen Gesundheitsdienst.

Das steigende Durchschnittsalter der Bevölkerung in Deutschland verändert die Anforderungen an die Gesundheitsförderung und -versorgung. Durch die individuelle Erfahrung mit Technologien steigt in allen Bevölkerungsgruppen der Anspruch, auch im öffentlichen Gesundheitsdienst digitale Leistungen zu erhalten. Bürgern soll ein einfacher Zugang zu gesundheitsrelevanten Informationen (zum Beispiel Trinkbrunnen oder kühle Räume) und Gesundheitsdaten über digitale Kanäle bereitgestellt werden.

Weiterhin kann die Qualität der gesundheitlichen Betreuung durch digitale Hilfsmittel in Diagnose und Behandlung erhöht werden. Dabei spielt die digitale Vernetzung staatlicher und privater Gesundheitsleistungen eine bedeutende Rolle.

Zuwanderung ist ein Faktor des demographischen Wandels, welcher einen Einfluss auf die Arbeit in den Gesundheitsämtern hat. Mehrsprachige Zugänglichkeit bei Informationsangeboten und die Übernahme weiterer subsidiärer Aufgaben in der Gesundheitsversorgung erhalten einen höheren Stellenwert.

Nicht zuletzt betreffen die Alterungsprozesse der Gesellschaft auch die Gesundheitsämter selbst. Die Knappheit an Gesundheits-Fachkräften macht es notwendig Dienste und Aufgaben des öffentlichen Gesundheitsdienstes effizienter und effektiver zu erledigen.

Außerdem wird es im Wettbewerb mit der Privatwirtschaft immer schwieriger, ausreichend IT-Fachkräfte zu finden, um die Digitalisierung voranzutreiben. Der demographische Wandel führt daher indirekt zu einem Digitalisierungsdruck in den Gesundheitsämtern, welcher mit den bestehenden IT-Ressourcen und Entwicklungsmustern nur schwer kompensiert werden kann.

Hier können Low-Code-Plattformen eine wichtige Rolle spielen, da sie die Entwicklung von Lösungen ohne tiefgehende IT-Kenntnisse ermöglichen. Dies versetzt die Mitarbeitenden im ÖGD in die Lage, unabhängig von ihrem technischen Know-how an der Gestaltung von Digitalisierungsmaßnahmen teilzunehmen, die Bedarfe der alternden und diversifizierten Bevölkerung zu erfüllen und gleichzeitig den Mangel an IT-Fachkräften im ÖGD auszugleichen. Außerdem kann bei der Gestaltung und Anpassung digitaler Prozesse die Belastung für bestehendes Fachpersonal durch Einsatz von Mitarbeitenden ohne technische Vorkenntnisse reduziert werden.

### **1.2.2 Komplexität der Anwendungslandschaft**

Die Aufgaben des öffentlichen Gesundheitsdienstes werden durch zusätzliche rechtliche Vorgaben und die Übernahme subsidiärer Leistungen komplexer. Viele Aufgaben weisen einen direkten Personenbezug auf und können nur digital gestützt, jedoch nicht vollständig digitalisiert werden. Die Komplexität der Aufgaben, der Grad der Datenerhebung sowie -analyse und die Vernetzung der Anwendungen und Akteure untereinander spiegelt sich auch im Anwendungsportfolio wider. Neben den bestehenden großen IT-Fachverfahren kommen vermehrt kleinere IT-Anwendungen zur Erledigung einzelner Aufgaben zum Einsatz. Dies sind häufig alleinstehende IT-Anwendungen, die durch das Gesundheitspersonal selbst entworfen wurden.

Hinzu kommt, dass sich die Änderungsdichte für rechtlichen Vorgaben des Verwaltungshandeln in Gesundheitsämtern erhöht. Entsprechend müssen sowohl größere Fachverfahren als auch kleinere Anwendungen entsprechend in kürzeren Abständen schneller angepasst werden. Der öffentliche Gesundheitsdienst steht vor der Herausforderung diese Änderungszyklen bei Dienstleistungen bis zu den in den Gesetzesänderungen dafür vorgegebene Fristen durchzusetzen oder die Fähigkeiten zur Anpassung selbst zu entwickeln.

Sowohl die Komplexität der Anwendungslandschaft als auch deren Entwicklungstempo erfordern Fach- und Steuerungskompetenzen innerhalb der Verwaltung. Hierzu zählen nicht nur die technischen und gesundheitsfachlichen Kompetenzen, auch die Einhaltung hoher Datenschutzstandards und die Abbildung geeigneter IT-Sicherheitsmaßnahmen erfordern neue Lösungen. Low-Code-Plattformen können in diesem Kontext eine Möglichkeit bieten die Komplexität zu bewältigen, indem sie einen kontrollierten Rahmen für die Abbildung der fachlichen Anforderungen vorgeben und dem Fachpersonal den Freiraum zur Gestaltung ihrer Lösung bieten.

# 2

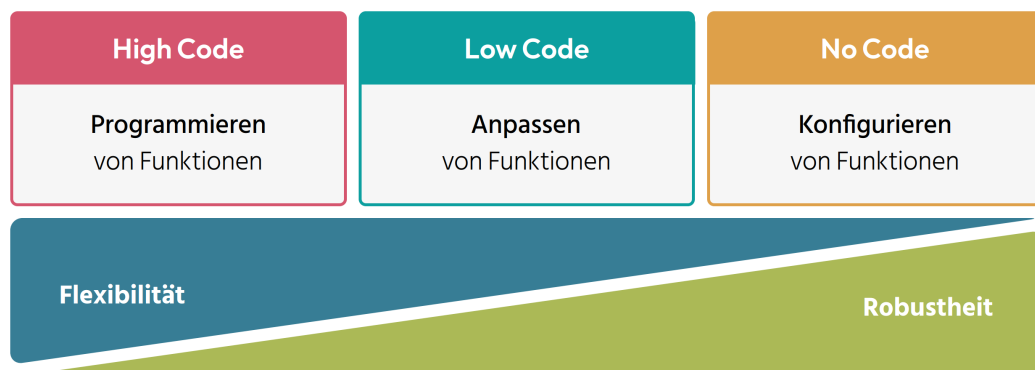
## GRUNDLAGEN ZU LOW-CODE-PLATTFORMEN

### 2.1 Begriffserklärung und theoretische Grundlagen

Die klassische Anwendungsentwicklung ist dadurch gekennzeichnet, dass Programmierende, die meist mehrere Programmiersprachen beherrschen, völlig flexibel Softwarelösungen entwickeln. Diese werden als High-Code-Anwendungen bezeichnet. Low-Code- und No-Code-Anwendungen sind im Vergleich dazu Programme, mit denen intuitiv auf einer grafischen Oberfläche die Softwarekomponenten und -funktionen zu einem Workflow zusammengestellt werden können.

Der Unterschied zwischen den beiden Plattformen ist, dass bei Low-Code zusätzliche Funktionen entwickelt und angepasst werden können. Bei No-Code hingegen kann das System nicht um eigene Funktionen erweitert werden, es bietet dadurch aber eine hohe Robustheit, da alle Funktionen aufeinander abgestimmt sind.

Im Rahmen dieses Whitepaper wurde die Entscheidung getroffen, sich auf die Betrachtung von Low-Code-Systemen zu konzentrieren, da diese Systeme erweiterbar sind und so die Möglichkeit bieten, schnell auf unerwartete Anforderungen mit individuellen Prozessen reagieren zu können.



#### 2.1.1 Vorteile von Low-Code-Anwendungen

Wiederkehrende, standardisierte Aufgaben können mit dem Low-Code-Ansatz mit geringem Zeitaufwand automatisiert werden und es können Robotic Process Automation (RPA) oder KI-Komponenten eingebunden werden.

##### 2.1.1.a Mitarbeitende im Workflow

Mit vorgefertigten Bausteinen, also modularen, wiederverwendbaren Komponenten innerhalb einer Low-Code-Plattform, ist der Einstieg in die Anwendungsentwicklung schnell und effizient umgesetzt. Diese Bausteine können vordefinierte UI-Elemente, Datenbankverbindungen oder Workflow-Komponenten umfassen, die ohne tiefgehende Programmierkenntnisse kombiniert und angepasst werden können. Einfache Anwendungen unterschiedlicher Anwendungsfälle lassen sich somit schnell und ohne umfangreiche Kenntnisse in Programmiersprachen oder der Methoden des Software-Engineerings erstellen. Fachliche Anforderungen können häufig über

Standardkomponenten direkt durch Mitarbeitende umgesetzt werden, ohne dass Dienstleister hinzugezogen werden müssen. Zum Aufbau und Bedienen des Workflows sind keine Programmierkenntnisse erforderlich. Individuell gefertigte Lösungen einzelner Mitarbeitenden werden vereinheitlicht und auf eine gemeinsame, leichter wartbare Basis gebracht.

#### 2.1.1.b    **Standardisierung**

Bei Bedarf und komplexeren Fragestellungen können eigene Bausteine mit Standardkomponenten oder eigenem Code entwickelt werden, die anschließend der gesamten Organisation zur Verfügung stehen. Durch die Verwendung von Standardkomponenten des Plattformanbieters, wie zum Beispiel Dateien hochladen, Dokumente verakten, Bescheide erstellen oder Bescheid versenden, können Entwickler den Entwicklungsprozess beschleunigen, wiederkehrende technische Fragestellungen durch den Plattformanbieter einheitlich gelöst und die Ergebnisse in den Anwendungen nachgenutzt werden. Das hat zur Folge, dass Anwendungen und Code reduziert werden, was die Wartbarkeit, Zuverlässigkeit und Sicherheit von Software erhöht und Fehler in Konfigurationen verringert. Testen und Debuggen, Bereitstellen ist in standardisierten und bereits geprüften Komponenten viel einfacher und schneller ausgeführt. Zusätzlich zu den vorgefertigten Komponenten haben Entwickler die Möglichkeit, individuellen Code in ihre Low-Code-Anwendungen zu integrieren. So wird die Flexibilität für Erweiterungen und Anpassungen beibehalten.

#### 2.1.1.c    **Anpassbarkeit**

Auch die Anpassung bestehender Arbeitsabläufe wird vereinfacht. Es gibt hier keine Wartezeiten und Kosten für externe Dienstleister, denn die Mitarbeitenden können die Anpassungen selbst vornehmen. Die Erfahrungen in der Arbeit mit der selbst erstellten Anwendung fließen direkt in ihre Weiterentwicklung ein, ohne dass umfassende Anforderungsdefinitionen für Dienstleister vorgenommen und kommuniziert werden müssen.

### 2.1.2    **Einsatzzwecke**

Mit einer Low-Code-Plattform können Standardprozesse der Verwaltung, wie Antrags- und Genehmigungsprozesse, zum Beispiel Antrag auf Ausstellung eines Gesundheitszeugnisses nach §43 Infektionsschutzgesetz, oder etwa die Dokumentation von Patientenkontakten umgesetzt werden. Sind derartige Umsetzungen bereits einmal erfolgt bei einem Gesundheitsamt, ist es mit einer Low-Code-Plattform einfacher, diese in einem anderen Gesundheitsamt nach zu nutzen und an die dortigen Rahmenbedingungen (z. B. Corporate Design oder leicht andere Felder in Formularen) anzupassen.

Es können häufig genutzte Einsatzszenarien als vorgefertigte Bausteine mit ausgeliefert werden. Das sind z. B. vorgefertigte Komponenten zur Stammdaten- oder Adresserfassung, Datensätze zusammenführen, Formulare erstellen oder automatisiert Benachrichtigungen verschicken. Dazu können standardisierte Funktionalitäten, wie Filteroptionen oder eine umfassende Suche, sind direkt integriert sein und bei jeder Anwendung einfach eingebunden werden. Dies beschleunigt zum einen den Erstellungsprozess und reduziert zum anderen die Komplexität im Vergleich zur herkömmlichen Programmerstellung.

Fügen Sie Daten, Prozesse und Anwendungen einfach via Drag-and-Drop zusammen, oder entwerfen Sie Ihre eigenen Bausteine hierbei unterstützen Drag-and-Drop-UI-Builder, sowie Report-Builder.

Low-Code-Plattformen bieten ein großes Potenzial, um Interoperabilität in heterogenen Systemlandschaften herzustellen – wie sie besonders häufig im öffentlichen Gesundheitswesen anzutreffen sind. Ein wesentlicher Vorteil liegt darin, dass viele dieser Plattformen bereits über eine Vielzahl standardisierter Schnittstellen verfügen. Diese sind oft vorkonfiguriert für gängige

Systeme wie E-Akten, Datenbanken oder umfassende IT-Ökosysteme wie Microsoft 365. Darüber hinaus ermöglichen Low-Code-Plattformen auch die Entwicklung individueller Schnittstellen – beispielsweise zur Integration von Altsystemen (Legacy-Systemen) oder zur Umsetzung spezifischer Anforderungen in der Gesundheitsverwaltung, etwa XÖV-Schnittstellen oder Anbindungen an die Telematikinfrastruktur (TI): FHIR, HL7. Auch Standards wie IHE, DICOM oder CDISC für den medizinischen Datenaustausch lassen sich integrieren. So können unterschiedliche Datenquellen effizient harmonisiert und standardisiert werden.

### 2.1.3 Vorgehen in der herkömmlichen Softwareentwicklung

In der herkömmlichen Softwareentwicklung werden genaue Anweisungen für den Computer erstellt, damit eine Anwendung funktioniert. Diese Steuerung im Hintergrund nennt man Backend. Anpassungen und Erweiterungen erfordern hierbei tiefgehende Programmierkenntnisse. Um die Bedienung solcher Anwendungen einfacher zu gestalten, werden grafische Nutzeroberflächen entwickelt, das Frontend. Diesem liegt ebenfalls Code zu Grunde und für Anpassungen braucht es auch hier Programmierkenntnisse. Diese Kombination aus Backend und Frontend ist den meisten Menschen aus der täglichen Nutzung bekannt, die wenigsten können jedoch Änderungen daran vornehmen. Meist ist dies auch von Unternehmen und Entwicklern gar nicht gewünscht, so dass der Code gar nicht eingesehen und verändert werden kann. Hier kann nur das genutzt werden, was ausgeliefert wird, eine Anpassung im Workflow ist immer mit einer Kontaktaufnahme, der Aufnahme der Anforderungen und entsprechenden Entwicklungsaufwand möglich.

### 2.1.4 Schnellvergleich

	High-Code	Low-Code	No-Code
Intuitive Bedienung	– Nein	+ Ja	+ Ja
Grafische Oberfläche	+ / – Teilweise	+ Ja	+ Ja
Schnelles Ändern von Prozessen	+ / – Teilweise	+ Ja	+ Ja
Individuelle Anpassungen von Parametern	+ Ja	+ Ja	– Nein
Erweiterbar mit eigenen Anforderungen	+ Ja	+ Ja	– Nein
Programmierkenntnisse erforderlich	+ Ja	+ / – Teilweise	– Nein
Hohe Robustheit	+ / – Nur mit entsprechenden Test- und Deploymentzyklen	+ / – Abhängig von individuellen Erweiterungen	+ Ja

Tabelle 1: Schnellvergleich High-, Low- und No-Code

# 3

## EINSATZSZENARIEN (USE CASES) UND BEST PRACTICES

### 3.1 Einsatzszenarien im ÖGD

Im Folgenden werden nun die Einsatzszenarien und Anwendungsfälle für Low-Code-Plattformen im ÖGD beschreiben. Diese basieren unter anderem auf den Ergebnissen von Interviews mit Bedarfsträger\*innen aus Gesundheitsämtern und veranschaulichen, wie konkrete Herausforderungen adressiert werden können. Sie umfassen typische Aufgaben und Fachverfahren die in Gesundheitsämtern auftreten.

#### 3.1.1 Vorgehen bei den Interviews

Die Grundlage für die Einsatzszenarien und Anwendungsfälle sind von den Autoren durchgeführte Interviews mit Bedarfsträger\*innen aus dem ÖGD die ihr Interesse an einer Mitarbeit nach einem ersten Workshop im September 2024 angemeldet hatten. Die Durchführung der Interviews erfolgte auf der Basis eines Interviewleitfadens und umfasst folgende Punkte:

- Vorstellung der Person und Ihrer Rolle
- Aktuelle Herausforderungen in der täglichen Arbeit
- Erfassung konkreter Anwendungsfälle (Use Cases)

#### 3.1.2 Anwendungsfall Infektionsschutz

Die stellvertretende Leitung eines Gesundheitsamts, das an ein Landratsamt angegliedert ist, beschrieb ihre Situation und Herausforderungen. Seit zehn Jahren im ÖGD tätig, leitet die Interviewpartnerin ein Team von 30 Mitarbeitenden. Dem Amt steht lediglich eine IT-Fachkraft für zwei Tage die Woche zur Verfügung. Ansonsten gibt es nur eine IT-affine Person im Team. Eine IT-Integration in die Struktur des Landratsamtes fehlt.

Das Gesundheitsamt verfügt über eine Digitalisierungsstrategie, deren Umsetzung teilweise eine große Herausforderung darstellt und somit in deren Umsetzung einige Kompromisse gemacht werden. Während einige Fachverfahren wie Schuleingangsuntersuchungen und die amtsärztlichen Untersuchungen bereits digitalisiert sind, gibt es in anderen wie der Krankenhausbegehung noch keine Digitalisierungsansätze.

Das Fachverfahren zur Datenverarbeitung im Infektionsschutz gestaltet sich wie folgt:

- Labore melden Infektionsfälle über das System DEMIS während Meldungen von Ärzt\*innen meist per Fax in einem digitalen Postfach eingehen. In seltenen Fällen werden Papierunterlagen eingereicht.
- Die Mitarbeitenden im Gesundheitsamt prüfen die verschiedenen Eingangskanäle um die Informationen zusammenzuführen.
- Anschließend werden die Fälle mit teilweiser Unterstützung durch das System SurvNet erfasst, sodass bestimmte Schritte automatisch ablaufen, während andere manuell ergänzt werden müssen. Hygienekontrolleur\*innen können danach direkt darauf zugreifen.
- Schriftsätze der Ärzt\*innen werden in SurvNet abgespeichert, bevor sie in KomXwork, der digitalen Akte des Landratsamts archiviert werden.

Low-Code-Plattformen könnten Medienbrüche in diesem Prozess adressieren. Eine Anwendung könnte die Eingangskanäle wie DEMIS und das Funktionspostfach zusammenführen und die automatische Zuordnung der Daten ermöglichen. Halbautomatische Schritte wie die Erfassung in SurvNet, könnten durch zusätzliche Automatisierung ergänzt werden. Die Anbindung an bestehende Anwendungen wie SurvNet und KomXwork könnte durch Low-Code-Plattformen vereinfacht werden um redundante Arbeitsschritte zu vermeiden.

Low-Code-Plattformen bieten Potenziale jedoch auch klare Herausforderungen. Eine der größten Hürden sind die komplexen Schnittstellen bestehender Systeme wie SurvNet und KomXwork.

### **3.1.3 Anwendungsfall Hygienekontrollen**

Ein Beispiel für einen ressourcenintensiven Prozess im Öffentlichen Gesundheitsdienst ist die Durchführung und Dokumentation von Hygienekontrollen. Diese finden regelmäßig in Einrichtungen wie Restaurants, Pflegeheimen oder Arztpraxen statt. Bisher verlaufen viele dieser Abläufe noch papierbasiert: Kontrolleure erfassen Mängel mit Stift und Papier, schreiben anschließend Berichte am PC und leiten die Ergebnisse manuell weiter – ein Prozess, der anfällig für Medienbrüche, Verzögerungen und Übertragungsfehler ist.

Hier können Low-Code-Plattformen gezielt zur Digitalisierung beitragen. So können z. B. mobile Erfassungs-Apps erstellt werden, mit denen Hygienekontrolleure Mängel direkt vor Ort über ein Tablet oder Smartphone dokumentieren. Ein Beispiel liefert das Gesundheitsamt Dortmund, das einen solchen Prozess mit einer modular aufgebauten Anwendung digitalisiert hat: Die Plattform schlägt automatisch neue Kontrolltermine vor, Einrichtungen erhalten im Vorfeld digitale Checklisten, und das Prüfpersonal dokumentiert Auffälligkeiten medienbruchfrei im System. Die Ergebnisse fließen direkt in das Fachverfahren ein, und die abschließende Mängelliste wird automatisch als PDF erzeugt und per E-Mail an die geprüfte Einrichtung versendet ohne Zwischenschritte oder Papierformulare. Durch den Einsatz einer Low-Code-Plattform kann dieser Digitalisierungsprozess umgesetzt werden.

Anpassungen – etwa bei gesetzlichen Änderungen oder neuen Formularfeldern – lassen sich kurzfristig realisieren, ohne eine externe IT beauftragen zu müssen. Für die Betriebe vor Ort bedeutet die Digitalisierung wiederum schnellere Rückmeldungen und eine höhere Nachvollziehbarkeit der Kontrollergebnisse.

### **3.1.4 Anwendungsfall Beseitigung von Medienbrüchen**

Die IT-Koordinatorin eines Gesundheitsamts schilderte ihre Situation und ein zentrales Problem, das in der Befragung hervorgehoben wurde, ist das Change-Management. Die Akzeptanz einiger Parteien für Digitalisierung ist sehr gering was die Einführung neuer Technologien erschwert.

Die IT-Koordinatorin ist zuständig für die Implementierung von Software wie Sormas und die Betreuung von Projekten die durch die Fördermittel unterstützt werden.

Die Herausforderungen mit denen die IT-Koordinatorin konfrontiert ist, betreffen technische sowie organisatorische Aspekte. Einer der zentralen Punkte sind Medienbrüche. So wird der zahnärztliche Dienst noch mit Stift und Papier durchgeführt, wobei die erhobenen Daten manuell in die Fachanwendung ISGA übertragen werden müssen. Diese Herausforderung wird jedoch nicht allein durch die Einführung von Low-Code-Plattformen gelöst, wenn es an der nötigen mobilen Hardware oder einem verschlüsselten Zugang zum eigenen Netzwerk mangelt. Um die Vorteile von Low-Code-Technologien voll ausschöpfen zu können, müssen parallel auch die infrastrukturellen Voraussetzungen geschaffen werden.

Ein weiteres Problem ergibt sich bei Trinkwasseruntersuchungen bei denen eine Offline-Funktionalität erforderlich ist, da die Daten oft ohne Internetverbindung erfasst und später synchronisiert werden müssen. Zudem fehlt in der Kreisverwaltung ein etabliertes Mobile Device Management wodurch der Einsatz von Smartphones und Tablets erschwert wird.

Das Change-Management stellt die größte Herausforderung dar. Akzeptanzprobleme bestehen insbesondere bei Ärzt\*innen und medizinischen Fachangestellten. Einige Ärzt\*innen bevorzugen weiterhin analoge Arbeitsweisen wie das Diktieren auf Kassetten anstelle von Speech-to-Text Lösungen. Auch jüngere Mitarbeitende wie Sozialarbeiter\*innen zeigen oft wenig Interesse an digitalen Arbeitsweisen.

Low-Code-Plattformen bieten das Potential die im Anwendungsfall genannten Medienbrüche zu beseitigen indem sie die die Entwicklung von Anwendungen ermöglichen welche die manuelle Übertragung von Daten automatisieren. Allerdings zeigt sich das die geringe Akzeptanz für selbst einfache Digitalisierungsaufgaben unter den Mitarbeitenden den Aufwand im Change-Management erheblich erhöht. Es wäre hier notwendig gezielte Schulungsmaßnahmen zu nutzen um die Einführung zu erleichtern und die Akzeptanz zu steigern. Zudem gibt es für diesen Anwendungsfall eine OZG-Leistung, die ebenfalls berücksichtigt werden muss.

### **3.1.5 Anwendungsfall Schulungen**

Die Interviewpartnerin ist eine ehemalige Radiologin mit technischer Affinität, die nun für das Gesundheitsamt zuständig ist. Das Gesundheitsamt steht vor Herausforderungen in der Schulungsorganisation für IT-Anwendungen. Der aktuelle Personalbestand umfasst vier Ärzte, 1,5 Hygienekontrolleure sowie 13 medizinische Fachkräfte, jedoch keine Verwaltungskraft. Der bestehende Fachkräftemangel und der hohe Personalwechsel erschweren die Wissensvermittlung erheblich.

Die derzeitige Schulungspraxis ist ineffizient, da Mitarbeitende je nach Dienstverhältnis (Kommunalbedienstete vs. staatlich Bedienstete) an unterschiedlichen Standorten geschult werden. Dies verursacht einen hohen organisatorischen Aufwand, erschwert die Teilnahme und führt zu einer inkonsistenten Wissensvermittlung. Ein Wiki wurde als Ersatz für zentrale Schulungen erstellt, ist jedoch pflegeintensiv und wird nicht effizient genutzt. Zudem verschärft die fehlende Unterstützung durch GA-Lotsen die Problematik.

Im Zusammenhang mit der Einführung von Low-Code-Plattformen, ergeben sich zusätzliche Anforderungen an die Schulung der Mitarbeitenden. Sie bieten Potenzial zur Prozessoptimierung, erfordern jedoch gezielte Qualifizierungsmaßnahmen, insbesondere für Fachkräfte ohne IT-Hintergrund.

Eine Schulungsstrategie ist notwendig, um die Einführung und Nutzung zu gestalten und die Akzeptanz bei den Mitarbeitenden zu fördern.

### **3.1.6 Hauptprobleme von Interviewpartnern**

Die Hauptprobleme der Interviewpartner konnten zusammengetragen werden und stellen einen guten Querschnitt durch alle Bereiche dar.

- Zu wenige IT-Fachkräfte
- IT-affine Personen in anderer Beschäftigung, deshalb können sie sich nicht hauptberuflich um die IT kümmern
- Inkonsequente Umsetzung von IT-Strategien
- Dateneingang über verschiedene Kanäle und Medien
- Dateneingänge sind nicht vollständig oder sehr individualisiert
- Händische Aufbereitung der eingegangenen Daten
- Hohe Kosten durch manuelles ausfüllen und Übertragen von Papierdokumenten, sowie hohe Transportkosten
- Softwareanwendungen müssen offline funktionieren, da nicht überall eine ausreichende Netzabdeckung vorhanden ist
- Hoher Schulungsaufwand durch unterschiedlich eingesetzte Softwarelösungen und Arbeitsworkflows

## 3.2 Best Practices

Dieses Kapitel stellt bewährte Praxisbeispiele vor, in denen Low-Code-Plattformen erfolgreich zur Digitalisierung von Verwaltungsabläufen eingesetzt wurden. Die dargestellten Success Stories zeigen konkrete Anwendungsfälle auf, die sowohl den Mehrwert als auch die Herausforderungen bei der Einführung und Nutzung solcher Technologien beleuchten. Sie dienen als Orientierungshilfe für Entscheidungsträger, die Low-Code als Werkzeug zur Modernisierung der Verwaltung in Betracht ziehen.

### 3.2.1 Success Story Intrexx Allianz

Die Senatsverwaltung für Finanzen (SenFin) und die Senatsverwaltung für Arbeit, Soziales, Gleichstellung, Integration, Vielfalt und Antidiskriminierung (SenASGIVA) haben in den letzten Jahren verschiedene interne Prozesse mithilfe der Low-Code-Plattform Intrexx digitalisiert. Dazu gehören Anwendungen wie Zeiterfassung, Platzbuchung, Abwesenheitsanträge, Mobile-Arbeit-Anträge und Ressourcenreservierung.

Aufgrund des steigenden Interesses anderer Behörden an diesen Lösungen sowie der Möglichkeit, kleinere Fachverfahren mit Intrexx selbstständig zu digitalisieren, wurden die SenFin und SenASGIVA regelmäßig von Berliner und bundesweiten Verwaltungen kontaktiert. Die hohe Nachfrage führte zu zahlreichen Einzelgesprächen und Beratungen. Um diesen Austausch zu bündeln und institutionell zu verankern, wurde im November 2022 die Intrexx-Allianz gegründet.

Die Intrexx-Allianz ist ein Netzwerk aller Berliner Behörden, die Low-Code auf Basis von Intrexx nutzen, einführen oder eine Einführung planen. Die Allianz verfolgt einen kooperativen und dezentralen Ansatz:

- Anwendungen werden gemeinsam entwickelt und über eine Austauschplattform bereitgestellt
- Behörden können bestehende Lösungen nachnutzen, weiterentwickeln und an eigene Anforderungen anpassen
- Wissen, Dokumentationen und Erfahrungen werden ohne Zuständigkeitsbarrieren geteilt.
- Einheitliche Prozessstandards werden gemeinsam erarbeitet

Ziel ist es, die Chancen der Low-Code-Technologie für Digitalisierung zu nutzen und den Entwicklungsaufwand durch behördenübergreifende Zusammenarbeit zu minimieren.

#### 3.2.1.a Umgesetzte Anwendungen und Prozesse

Im Rahmen der Intrexx-Allianz wurden verschiedene digitale Lösungen entwickelt, die von anderen Behörden übernommen und angepasst werden können:

- Digitale Zeiterfassung: Automatisierte Erfassung von Arbeitszeiten ohne Medienbrüche
- Platzbuchungs-App: Flexibles Buchen von Arbeitsplätzen, insbesondere im hybriden Arbeitsmodell
- Abwesenheitsmanagement: Vereinfachte Genehmigung und Dokumentation von Urlaubs- und Krankmeldungen
- Mobile Arbeit- und Telearbeitsanträge: Digitale Antrags- und Genehmigungsprozesse für flexible Arbeitsmodelle
- Raum- und Ressourcenbuchung: Zentralisierte Reservierung von Besprechungsräumen und technischen Geräten
- Kanban-Board für Projektmanagement: Optimierte Aufgabensteuerung für agile Teams

Diese Anwendungen wurden zunächst bei SenFin und SenASGIVA eingeführt und stehen nun anderen Behörden zur Verfügung.

### 3.2.1.b Meilensteine der Intrexx-Allianz

Zwei zentrale Meilensteine markieren die Weiterentwicklung der Intrexx-Allianz:

- Aufbau des Plattform-Hostings durch das ITDZ, das ab 2024 allen interessierten Behörden ein Hosting-Angebot unterbreitet.
- Beschaffung einer Landeslizenz, die eine flächendeckende Nutzung von Intrexx ermöglichen soll.

### 3.2.1.c Akteure und Beteiligte

An der Intrexx-Allianz sind zahlreiche Behörden beteiligt oder betroffen, darunter:

- Senatsverwaltungen: SenFin, SenASGIVA, Senatskanzlei, SenInnSport, SenBJF, SenKultGZ, SenStadt, SenMVKU.
- Bezirksämter: Charlottenburg-Wilmersdorf, Neukölln, Pankow, Reinickendorf, Lichtenberg, Marzahn-Hellersdorf.
- Weitere Akteure: ITDZ, Rechnungshof von Berlin, Abgeordnetenhaus, LAF, LAGeSo, LEA, LAGetSi.

Die Intrexx-Allianz gehört zur Kategorie „Prozess- und Qualitätsmanagement sowie ressortübergreifende Zusammenarbeit“.

Die Intrexx-Allianz ist ein Beispiel für eine erfolgreiche ressortübergreifende Zusammenarbeit in der öffentlichen Verwaltung. Durch den gemeinsamen Ansatz bei der Nutzung von Low-Code-Plattformen werden Digitalisierungskosten gesenkt, Entwicklungszeiten verkürzt und standardisierte Prozesse etabliert.

# 4

## HERAUSFORDERUNGEN UND RAHMENBEDINGUNGEN

### 4.1 Herausforderungen beim Einsatz von Low-Code-Plattformen

Der Einsatz von Low-Code-Systemen in der öffentlichen Verwaltung bietet viele Vorteile, wie eine schnellere Digitalisierung und eine einfachere Anpassung von Anwendungen. Nichtsdestotrotz sollten auch die Herausforderungen, die sie mit sich bringen, berücksichtigt werden.

### 4.2 Zusätzliche Aufwände durch unzureichende Prozessdokumentation

Die Einführung von Low-Code-Plattformen verspricht eine schnellere und effizientere Entwicklung von Anwendungen. Doch diese Vorteile können schnell durch zusätzliche Aufwände überschattet werden, wenn die internen Prozesse eines Unternehmens nicht ausreichend dokumentiert sind. Ohne eine klare und umfassende Dokumentation der bestehenden Prozesse kann es zu Missverständnissen und Fehlern kommen, die den Entwicklungsprozess verlangsamen. Entwickler und Anwender der Low-Code-Plattform könnten Schwierigkeiten haben, die genauen Anforderungen und Abläufe zu verstehen, was zu ineffizienten Lösungen und erhöhtem Nacharbeitsaufwand führt.

Zudem kann die mangelnde Dokumentation dazu führen, dass wichtige Prozessschritte übersehen werden, was die Qualität der entwickelten Anwendungen beeinträchtigen kann. Dies kann wiederum zu einer geringeren Akzeptanz der neuen Plattform im Unternehmen führen und den erhofften Nutzen der Low-Code-Technologie schmälern.

Um diese Herausforderungen zu bewältigen, ist es entscheidend, vor der Einführung einer Low-Code-Plattform eine gründliche Bestandsaufnahme und Dokumentation der bestehenden Prozesse durchzuführen. Dies stellt sicher, dass alle Beteiligten ein gemeinsames Verständnis der Anforderungen und Abläufe haben und die Low-Code-Plattform effektiv nutzen können.

### 4.3 Datenschutz und IT-Sicherheit

Die Datenschutz-Grundverordnung (DSGVO) stellt hohe Anforderungen an den Schutz personenbezogener Daten. Plattformbetreiber müssen sicherstellen, dass alle Datenverarbeitungsprozesse den gesetzlichen Vorgaben entsprechen. Dies umfasst die Einholung der Einwilligung der betroffenen Personen, die Sicherstellung der Datenminimierung und die Implementierung geeigneter Sicherheitsmaßnahmen. Die Einwilligung der betroffenen Personen ist dabei von zentraler Bedeutung, da sie sicherstellen muss, dass die Datenverarbeitung auf einer rechtlichen Grundlage erfolgt. Die betroffenen Personen müssen daher darüber informiert werden, welche Daten zu welchem Zweck verarbeitet werden. Die Datenminimierung bedeutet, dass nur die Daten erhoben und verarbeitet werden dürfen, die für den jeweiligen Zweck notwendig sind. Geeignete technische und organisatorische Maßnahmen müssen implementiert werden, um die Sicherheit der Daten zu gewährleisten und sie vor unbefugtem Zugriff zu schützen.

Es muss klar definiert sein, wer für die Datenverarbeitung verantwortlich ist. Dies betrifft sowohl den Plattformbetreiber als auch die öffentliche Verwaltung, die die Plattform nutzt. Es muss eindeutig festgelegt sein, wer die Verantwortung für die Einhaltung der Datenschutzvorschriften trägt. Dies ist besonders wichtig, um sicherzustellen, dass alle Beteiligten ihre Pflichten und Verantwortlichkeiten kennen und erfüllen.

Bei der Nutzung solcher Plattformen können personenbezogene Daten auf Servern außerhalb der EU gespeichert oder verarbeitet werden. Dies erfordert besondere Vorsichtsmaßnahmen, um den Datenschutz zu gewährleisten. Eine Möglichkeit, dies zu erreichen, ist die Implementierung von Standardvertragsklauseln, die eine rechtliche Grundlage für die Datenübertragung in Drittländer bieten und sicherstellen, dass ein angemessenes Datenschutzniveau eingehalten wird. Zusätzlich sollten weitere Sicherheitsmaßnahmen wie die Verschlüsselung und Pseudonymisierung der Daten implementiert werden, um sie vor unbefugtem Zugriff zu schützen.

## 4.4 Aufbewahrung und Löschung

Information dürfen nur so lange gespeichert und verarbeitet werden, wie sie für den jeweils definierten Zweck benötigt werden. Wenn der Zweck nicht (mehr) besteht, müssen sie gelöscht werden, es sei denn, es besteht eine rechtliche oder vertragliche Verpflichtung zur Aufbewahrung.

Die Plattform muss daher Mechanismen bereitstellen, um den Betroffenen ihre Rechte gemäß DSGVO zu ermöglichen. Dazu gehören das Recht auf Auskunft, Berichtigung, Löschung und Widerspruch.

Das Recht auf Auskunft bedeutet, dass die betroffenen Personen das Recht haben zu erfahren, welche Daten über sie gespeichert sind und wie diese verwendet werden. Das Recht auf Berichtigung ermöglicht es den betroffenen Personen, die Korrektur unrichtiger Daten zu verlangen. Das Recht auf Löschung erlaubt es den betroffenen Personen, unter bestimmten Bedingungen die Löschung ihrer Daten zu verlangen. Das Recht auf Widerspruch gibt den betroffenen Personen die Möglichkeit, der Verarbeitung ihrer Daten zu widersprechen. Für die Verarbeitung personenbezogener Daten gelten die Standardlöschfristen von Protokollierungsdaten 30 bis 90 Tage. Bei Services die auf der Low-Code-Plattform aufsetzen gelten die Bestimmungen der jeweiligen Verantwortlichen.

## 4.5 Personalmitbestimmung (Beteiligung)

Ein weiterer wesentlicher Punkt ist die Beteiligung der verschiedenen Akteure innerhalb der öffentlichen Verwaltung. Die Einführung von Low-Code-Plattformen erfordert eine enge Zusammenarbeit zwischen verschiedenen Abteilungen und Ebenen der Verwaltung. Dies kann eine Herausforderung darstellen, da unterschiedliche Interessen und Prioritäten berücksichtigt werden müssen. Eine erfolgreiche Implementierung erfordert daher eine klare Kommunikation und Koordination zwischen den Beteiligten. Es ist wichtig, dass alle relevanten Akteure frühzeitig in den Prozess einbezogen werden, um ihre Bedürfnisse und Bedenken zu berücksichtigen und eine breite Akzeptanz zu gewährleisten.

Beteiligungsmanagement spielt hierbei eine zentrale Rolle, um sicherzustellen, dass die verschiedenen Interessen ausgewogen berücksichtigt werden und die Zusammenarbeit reibungslos verläuft.

In die Auswahl einer Low-Code Plattform sollten unbedingt die Bedarfsträger aus den Gesundheitsämtern mit einbezogen werden. Entscheidende Rollen sind hierbei z. B.

- Stellvertretende Leitung eines Gesundheitsamtes
- Teamleitung
- IT-affine Personen in den Teams
- IT-Koordination eines Gesundheitsamtes
- Fachpersonal, welches mit der Anwendung später arbeiten soll

Personen abseits des alltäglichen Arbeitsworkflows kennen die Abläufe zu wenig, um bei der Softwareauswahl schnell Potentiale zu sehen. Weniger technisch-affine Mitarbeitende sollten intensiv in das Changemanagement und die Darstellung der Vorteile für ihre Arbeitsweise einbezogen werden, jedoch nicht in die Auswahl der Software.

Angesichts der üblichen Beteiligungsverfahren bei der Einführung neuer Softwarelösungen ist es wichtig, bei der Entwicklung neuer Lösungen innerhalb der Low-Code-Plattformen differenzierter vorzugehen. Da die kontinuierliche Entwicklung von Low-Code-Lösungen den klassischen Beteiligungsprozess bei jeder einzelnen Entwicklung erheblich verzögern und somit einen der Hauptvorteile von Low-Code zunichtemachen würde, sollten alternative Beteiligungsmethoden in Betracht gezogen werden. Zum Beispiel könnte der Personalrat bei Bedarf über ein Inspektionsrecht Einblick in bestimmte Module bzw. Entwicklungen erhalten.

## **4.6** **Betrieb und Verwaltung von Low-Code-Anwendungen**

Der Betrieb und die Verwaltung von Low-Code-Plattformen bringen einige Herausforderungen mit sich, insbesondere wenn es darum geht, einen Wildwuchs zu verhindern und klare Regeln für Änderungen an Fachverfahren zu schaffen.

### **4.6.1** **Wildwuchs verhindern**

Um die Gefahr eines Wildwuchses durch die kontinuierliche Entwicklung auf Low-Code-Plattformen durch verschiedene Nutzende zu vermeiden, sind klare Richtlinien und Governance-Strukturen notwendig. Dazu gehört eine zentrale Steuerung durch die IT-Abteilung oder ein Governance-Team, das sicherstellt, dass alle Entwicklungen den festgelegten Standards entsprechen. Richtlinien und Standards sollten Best Practices, Sicherheitsanforderungen und Qualitätsstandards umfassen. Es muss klar geregelt sein, wer was verändern darf. Zudem ist es wichtig, dass alle Nutzenden der Plattform ausreichend geschult sind und die notwendigen Fähigkeiten besitzen, um qualitativ hochwertige Anwendungen zu entwickeln. Außerdem sollte transparent festgehalten werden, wer für den Support der entwickelten Anwendungen zuständig ist.

### **4.6.2** **Regeln für Änderungen an Fachverfahren**

Ein strukturiertes Änderungsmanagement hilft dabei, festzulegen, wer wann Änderungen an Fachverfahren vornehmen darf. Klare Rollen und Zugriffsrechte für die Nutzer der Low-Code-Plattform sind essenziell. Nur autorisierte Personen sollten Änderungen an kritischen Fachverfahren vornehmen dürfen. Ein System für Änderungsanträge, bei dem alle Änderungen dokumentiert und von einem Gremium oder einer verantwortlichen Person genehmigt werden müssen, ist ebenfalls hilfreich. Versionierungssysteme ermöglichen es, Änderungen nachzuverfolgen und bei Bedarf rückgängig zu machen, was sicherstellt, dass alle Änderungen nachvollziehbar und dokumentiert sind.

### **4.6.3** **Organisation und Best Practices**

Die Organisation der Low-Code-Plattform und die Implementierung von Best Practices können durch agile Methoden wie Scrum oder Kanban unterstützt werden. Regelmäßige Reviews und Audits der Low-Code-Anwendungen stellen sicher, dass alle Entwicklungen den festgelegten Standards entsprechen und keine unautorisierten Änderungen vorgenommen wurden. Eine Erfahrungsaustauschgruppe, in der sich Entwickler und Nutzer der Low-Code-Plattform regelmäßig austauschen und Best Practices teilen können, fördert die kontinuierliche Verbesserung.

## 4.7 Integration in die bestehende IT-Landschaft

Die Einbindung einer neuen Low-Code-Plattform in die bestehende IT-Landschaft ist ein anspruchsvoller Vorgang, der eine gründliche Planung und enge Abstimmung erfordert. Ein zentraler Aspekt ist die Sicherstellung der Kompatibilität und Interoperabilität mit den vorhandenen Systemen und Anwendungen. Dies umfasst die Analyse der bestehenden IT-Infrastruktur und die Identifikation von Integrationspunkten, um eine nahtlose Kommunikation zwischen der neuen Plattform und den bestehenden Systemen zu gewährleisten. Dazu zählt auch die Sicherstellung einer reibungslosen Datenmigration und -synchronisation. Es ist entscheidend, dass bei der Nutzung der Low-Code-Plattform die Datenintegrität gewahrt bleibt und keine Datenverluste auftreten.

Hierbei können ETL-Tools (Extract, Transform, Load) und Middleware-Lösungen hilfreich sein, um den Datentransfer zu erleichtern und sicherzustellen, dass die Daten konsistent und aktuell bleiben. ETL-Tools (Extract, Transform, Load) sind Softwarelösungen, die den Prozess der Datenintegration unterstützen. Sie bestehen aus drei Hauptschritten: Daten werden aus verschiedenen Quellen wie Datenbanken, Anwendungen oder Dateien entnommen (Extract), die extrahierten Daten werden in ein geeignetes Format umgewandelt, bereinigt und angereichert, um den Anforderungen der Zielsysteme zu entsprechen (Transform), und schließlich werden die transformierten Daten in das Zielsystem, wie z. B. eine Datenbank oder eine Low-Code-Plattform, geladen (Load). Es ist jedoch (vor allem bei sensiblen Daten) wichtig, dass die Sicherheitsfunktionen der ETL-Tools richtig implementiert und regelmäßig überprüft werden, um sicherzustellen, dass sie den Datenschutzanforderungen entsprechen. Dazu zählen die Datenverschlüsselung, Zugriffskontrollen und Protokollierung bzw. Überwachung der Aktivitäten und Datenflüsse.

## 4.8 Anpassbarkeit und Flexibilität

Um eine nahtlose Integration einer neuen Low-Code-Plattform in die vorhandenen Systeme und Anwendungen zu gewährleisten, ist es entscheidend, dass die Plattform flexibel und anpassbar ist. Die Anpassbarkeit einer Low-Code-Plattform ermöglicht es Unternehmen, spezifische Geschäftsanforderungen und Prozesse effizient zu berücksichtigen. Durch die Möglichkeit, individuelle Module und Funktionen zu erstellen, kann die Plattform optimal an die bestehenden Systeme und Arbeitsabläufe angepasst werden, was die Integration erleichtert.

## 4.9 Skalierbarkeit

Auch das Thema Skalierbarkeit kann ggf. zu einem späteren Zeitpunkt zu einer großen Herausforderung werden, sofern sie nicht im Vorfeld mitberücksichtigt wird. Die Plattform muss in der Lage sein, mit dem Wachstum des Unternehmens Schritt zu halten. Dies bedeutet, dass sie skalierbar sein muss, um steigende Datenmengen und Benutzerzahlen zu bewältigen, ohne an Leistung zu verlieren.

## 4.10 Akzeptanz und Schulung der Mitarbeiter

Die Akzeptanz von Low-Code-Plattformen bei den Mitarbeitern kann eine komplexe Herausforderung darstellen. Ein wesentlicher Faktor ist der Widerstand gegen Veränderungen. Mitarbeiter könnten skeptisch gegenüber neuen Technologien sein, insbesondere, wenn sie befürchten, dass ihre bisherigen Fähigkeiten und Kenntnisse an Bedeutung verlieren könnten. Diese Skepsis kann zu einer geringeren Bereitschaft führen, sich auf die neue Plattform einzulassen.

Die Anpassung an neue Arbeitsweisen könnte ebenfalls ein Grund für Widerstand sein, da Mitarbeiter Zeit und Geduld benötigen, um sich an die neuen Abläufe zu gewöhnen und diese effizient zu nutzen. Die Einführung einer neuen Plattform kann bedeuten, dass bestehende Arbeitsprozesse und -methoden überarbeitet und angepasst werden müssen. Dies erfordert Zeit und Geduld, bis sich die Mitarbeiter an die neuen Abläufe gewöhnt haben und diese effizient nutzen können. Die Umstellung kann anfänglich zu einer Verringerung der Produktivität führen, bis die neuen Prozesse vollständig integriert sind. Nach dieser anfänglichen Phase, wenn sich die Mitarbeiter an die neuen Arbeitsweisen gewöhnt haben und die Prozesse reibungslos funktionieren, kann die Produktivität jedoch deutlich steigen. Die Mitarbeiter werden effizienter arbeiten können, da die Low-Code-Plattformen viele manuelle und zeitaufwändige Aufgaben automatisieren. Dies führt zu einer höheren Effizienz und ermöglicht es den Mitarbeitern, sich auf strategische und wertschöpfende Aufgaben zu konzentrieren. Langfristig kann dies zu einer verbesserten Arbeitszufriedenheit und einer gesteigerten Innovationsfähigkeit des Unternehmens beitragen.

Ein weiterer wichtiger Aspekt, der nicht unterschätzt werden darf, ist der Schulungsbedarf. Obwohl Low-Code-Plattformen darauf ausgelegt sind, die Komplexität der Anwendungsentwicklung zu reduzieren, erfordert ihre effektive Nutzung dennoch eine umfassende Schulung. Mitarbeiter müssen lernen, wie sie die Plattform optimal einsetzen können, um die gewünschten Ergebnisse zu erzielen. Dies kann zeitaufwendig sein und erfordert oft spezielle Schulungsprogramme und kontinuierliche Unterstützung.

## **4.11** **Lieferantenabhängigkeit**

Die Lieferantenabhängigkeit (Vendor Lock-in) bei Anbietern kann verschiedene Herausforderungen mit sich bringen. Eine der größten Herausforderungen sind die hohen Umstellungskosten. Denn der Wechsel zu einem anderen Anbieter kann erhebliche finanzielle und zeitliche Investitionen erfordern, was ein Hindernis darstellen kann, zu einem möglicherweise besseren oder kostengünstigeren Anbieter zu wechseln. Zudem können langfristige Verträge mit Anbietern Unternehmen an bestimmte Produkte oder Dienstleistungen binden. Diese Verträge beinhalten oft hohe Kündigungsfristen und -kosten, was die Flexibilität einschränkt.

Ein weiteres Problem sind technische Abhängigkeiten. Es kann zu technischen Abhängigkeiten kommen, wenn stark angepasste oder exklusive Lösungen eines Anbieters genutzt werden. Dies kann die Migration zu anderen Systemen erschweren und zusätzliche Kosten verursachen. Ein zu starker Fokus auf einen Anbieter kann auch dazu führen, dass weniger innovativ gearbeitet wird und neue Technologien oder Lösungen nicht in Betracht gezogen werden, was langfristig die Wettbewerbsfähigkeit beeinträchtigen kann. Darüber hinaus kann die Lieferantenabhängigkeit auch zu einer sogenannten Kompetenzbindung führen. Wenn Mitarbeiter speziell auf die Nutzung einer bestimmten Plattform geschult sind, kann dies zu einem Mangel an Flexibilität führen. Der Aufbau von Know-how für alternative Lösungen kann zeitaufwendig und kostspielig sein.

Um diese Herausforderungen zu bewältigen, sollten Unternehmen eine strategische Analyse ihrer Lösungen durchführen und sicherstellen, dass sie nicht zu stark von einem einzelnen Anbieter abhängig sind. Dies kann durch die Nutzung offener Standards, die Vermeidung langfristiger Verträge und die kontinuierliche Schulung der Mitarbeiter auf verschiedene Technologien erreicht werden.

Bei der Einführung einer Low-Code-Plattform müssen die bestehenden Sicherheitsaspekte und Compliance-Anforderungen berücksichtigt werden. Die neue Plattform sollte den bestehenden Sicherheitsstandards entsprechen und Maßnahmen wie (rollenbasierte) Zugriffskontrollen und Verschlüsselungstechnologien implementieren. Regelmäßige Sicherheitsüberprüfungen sind notwendig, um potenzielle Schwachstellen zu identifizieren und zu beheben. Daher ist im Vorfeld der Auswahl der richtigen Plattform wichtig, zu prüfen, ob die jeweilige Plattform die Rahmenbedingungen erfüllt.

Die Entwicklung auf der Low-Code-Plattform muss außerdem stets unter Einhaltung der rechtlichen Vorgaben erfolgen. Zu den wichtigsten Vorschriften bei Datenverarbeitung im ÖGD gehören das Gesundheitsdienst-Gesetz (GDG), den Landeskrankenhausgesetzen oder den Vorschriften kirchlicher Einrichtungen, länderspezifische Datenschutzgesetze, Regelungen in den Sozialgesetzbüchern und allgemeine Vorschriften wie die Datenschutzgrundverordnung (DSGVO) oder Bundesdatenschutzgesetz (BDSG, insbesondere Paragraph 4 zur Einwilligung in die Datenverarbeitung).

Um diese Herausforderung trotz sich stetig verändernder Vorschriften zu meistern, wird die Implementierung eines Compliance-Management-Systems (CMS) empfohlen. Das CMS überwacht und stellt die Einhaltung gesetzlicher Vorgaben sicher. Dies umfasst die Entwicklung von Compliance-Richtlinien, regelmäßige Schulungen und interne Audits. Regelmäßige Schulungen und Sensibilisierungsmaßnahmen für Entwickler und andere Beteiligte sind notwendig, um sicherzustellen, dass sie die rechtlichen Anforderungen und Best Practices verstehen und umsetzen können. Die Durchführung regelmäßiger Audits und Überprüfungen der entwickelten Anwendungen sind eine wichtige Maßnahme zur kontinuierlicher Einhaltung gesetzlicher Anforderungen. Dies umfasst auch die Analyse und Bewertung bestehender Compliance-Richtlinien und -Verfahren.

# 5

## LEITFADEN FÜR DIE EINFÜHRUNG VON LOW-CODE-PLATTFORMEN

Die Einführung von Low-Code-Plattformen in die öffentliche Verwaltung bietet zahlreiche Vorteile, erfordert jedoch eine sorgfältige Planung und Umsetzung. Mit den richtigen Strategien und Maßnahmen können die Herausforderungen gemeistert und die Digitalisierung erfolgreich vorangetrieben werden. Der folgende Leitfaden soll den Digitalisierungsverantwortlichen innerhalb der öffentlichen Verwaltung als umfassende Orientierungs- und Entscheidungshilfe bei der Einführung neuer Low-Code-Plattformen dienen. Ziel dieses Leitfadens ist es, eine fundierte Grundlage zu schaffen, die sowohl strategische als auch praktische Aspekte der Einführung bzw. Implementierung abdeckt. In diesem Leitfaden werden detaillierte Tipps zur Auswahl und Nutzung von Low-Code-Plattformen gegeben, der Umgang mit typischen Herausforderungen beschrieben und weitere wichtige Punkte behandelt, die für eine erfolgreiche Einführung und Nutzung solcher Plattformen in der öffentlichen Verwaltung notwendig sind.

### 5.1 Planung

Bei der Einführung einer neuen Low-Code-Entwicklungsplattform benötigt es einer sorgfältigen und strukturierten Planung. Die Einführung lässt sich allgemein in drei aufeinanderfolgende Phasen unterteilen: Vorbereitung, Beschaffung und Implementierung.

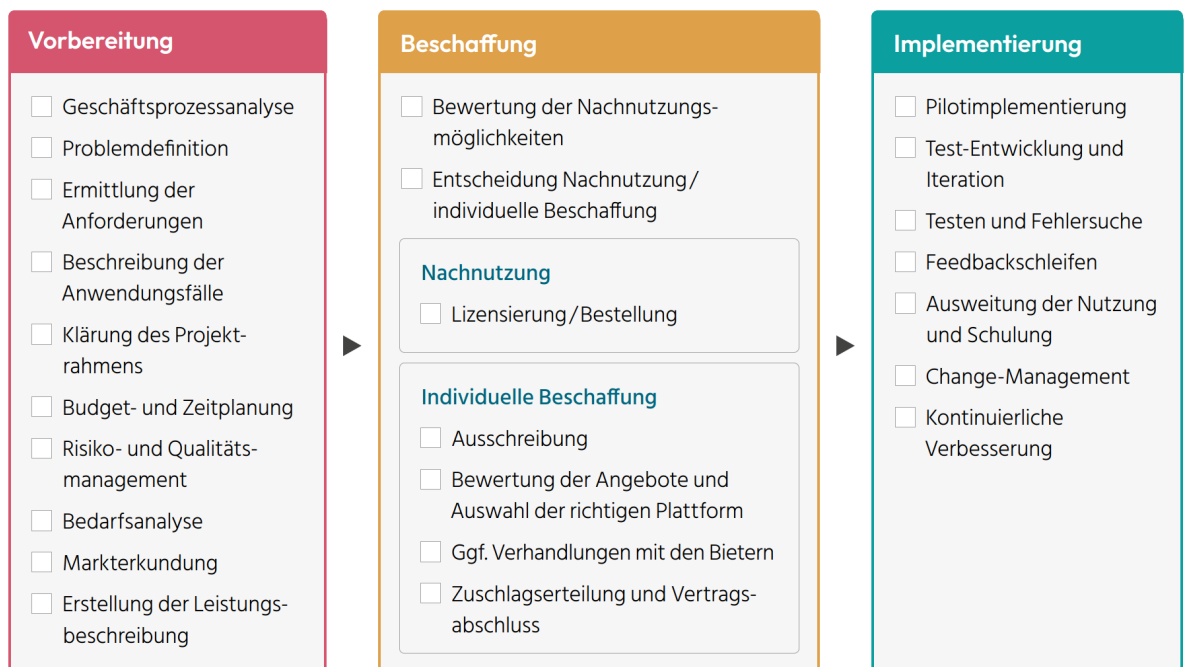


Abbildung 1: Einführungsphasen einer Low-Code-Plattform

### 5.1.1 Vorbereitungsphase

In der Vorbereitungsphase verschafft man sich ein detailliertes Bild darüber, was genau man sich durch den Einsatz einer Low-Code-Plattform erhofft. Das Ziel dieser Phase besteht darin, das Problem zu definieren, den Bedarf zu ermitteln und die potenziellen Anwendungsfälle (Use-Cases) und Geschäftsprozesse zu identifizieren, die verbessert, automatisiert oder effizienter gestaltet werden können. Der Fokus liegt somit auf die Identifizierung belastbarer Use-Cases, die die Effizienz und Durchgängigkeit der Aufgaben im ÖGD erhöhen.

Der Aufwand dieser Phase hängt stark davon ab, wie gut die Geschäftsprozesse dokumentiert sind. Bei unzureichender Dokumentation wird eine Geschäftsprozessanalyse der relevanten Prozesse empfohlen, da dies die spätere Entwicklung auf der Plattform beeinflusst.

Gut dokumentierte Prozesse vereinfachen außerdem die Festlegung des Schutzbedarfs der für die Arbeit auf der Low-Code-Plattform geplanten Daten. Die Schutzbedarfskategorie ist zur späteren Auswahl der geeigneten Plattform sehr wichtig, da sie die Anforderungen an die IT-Sicherheit beeinflusst und somit einen direkten Einfluss auf die Bewertung zur Auswahl der richtigen Plattform haben kann. Zur Schutzbedarfsfeststellung kann es je nach Art der Daten und den spezifischen Anforderungen sinnvoll sein, eine Kombination verschiedener Normen zu verwenden. Beispielsweise könnte die DSGVO für personenbezogene Daten und Gesundheitsdaten besonders wichtig sein, während ISO / IEC 27001 und BSI-Standard 200-2 für die allgemeine Informationssicherheit und Schutzbedarfsfeststellung relevant sind.

Es ist außerdem zu klären, wo die Risiken liegen, wie viel Budget zur Verfügung steht und welcher Zeitrahmen für die Einführung vorgesehen bzw. realistisch ist.

Eine Markterkundung ist hier eine gute Methode, realistische Anforderungen zu sammeln und sich ein Bild über die unterschiedlichen Funktionen verschiedener Plattformen sowie deren Kosten zu verschaffen. Zum Schluss der Vorbereitungsphase muss man in der Lage sein, die Anforderungen in Form einer Leistungsbeschreibung zu dokumentieren und die Genehmigung für die Beschaffung einzuholen.

Um den maximalen Nutzen aus einer Low-Code-Plattform zu ziehen und sich für die richtige Plattform zu entscheiden, ist es entscheidend, im Vorfeld Anwendungsfälle mit hohem Nutzenpotenzial zu identifizieren. Denn die Vorteile von Low-Code-Plattformen können sich bei einer strategisch langfristig geplanten Umsetzung deutlich stärker bemerkbar machen. Die wiederholte Nutzung der Plattform für verschiedene Projekte, im Vergleich zu einzelnen Anwendungen, erzeugt Skaleneffekte, fördert kontinuierliche Innovation, steigert die Effizienz, verbessert die Wartungsfreundlichkeit und bietet umfassende Integrationsmöglichkeiten. Der erste Schritt zur Identifikation geeigneter Anwendungsfälle ist die Analyse bestehender Geschäftsprozesse. Dieser Prozess kann durch eine systematische Analyse und Bewertung verschiedener Faktoren unterstützt werden.

Hierbei sollten Prozesse identifiziert werden, die durch Automatisierung und Digitalisierung erheblich verbessert werden können. Es ist wichtig, Prozesse zu wählen, die sowohl technisch machbar als auch wirtschaftlich sinnvoll sind. Dies bedeutet, dass die Prozesse nicht nur automatisiert werden können, sondern dass die Automatisierung auch einen signifikanten Mehrwert bietet. In diesem Zusammenhang werden die funktionalen und nicht-funktionalen Anforderungen ermittelt und dokumentiert. Eine Kosten-Nutzen-Analyse ist ein wichtiges Instrument, um die Wirtschaftlichkeit der ausgewählten Use-Cases zu bewerten. Sie hilft dabei, den finanziellen Aufwand (Kosten) und den erwarteten Nutzen gegenüberzustellen, um fundierte Entscheidungen zu treffen. So lassen sich mögliche Anwendungsfälle identifizieren, die von einer Low-Code-Plattform am meisten profitieren könnten.

Es ist auch möglich, durch die Einführung einer Low-Code-Plattform existierende Probleme in den Geschäftsprozessen zu beseitigen. Dazu werden die potenziellen Probleme dokumentiert, die durch den Einsatz der Low-Code-Plattform beseitigt werden könnten. Damit dies gelingt, sollten relevante Stakeholder durch eine Stakeholder-Analyse identifiziert und deren Einbindung gefördert werden. Es ist wichtig, dass am Ende der Vorbereitungsphase alle relevanten Anforderungen identifiziert und dokumentiert sind. Ist dies der Fall kann mit der Beschaffungsphase begonnen werden.

### 5.1.2 Beschaffungsphase

Zu Beginn der Beschaffungsphase ist es ratsam eine Marktsichtung anhand der vorher dokumentierten Anforderungen durchzuführen. Dadurch können ggf. bestehende Plattformen (z. B. über OZG) oder Rahmenverträge (z. B. über ITZBund) identifiziert werden, die kostengünstiger als eine Neubeschaffung nachgenutzt werden könnten. Ein Beispiel für eine bereits nachnutzbare Plattform ist Modul F vom Bundesministerium des Innern und für Heimat (BMI). Ebenfalls empfohlen wird die Sichtung der Angebote über Marktplätze zur Nachnutzung der EfA-Leistungen (z. B. FITKO und GovDigital).

Bei einer eventuellen Nachnutzung sollte besonders die individuelle Anpassbarkeit der Bestandsplattform geprüft werden. Hier sind die Lösungen in der Regel weniger anpassbar. Auch das Thema Datenschutz und -sicherheit müssen bedacht werden, da die Bestandsplattform in der Regel zentral betrieben wird. Auch der Funktionsumfang kann gegenüber der individuellen Beschaffung beschränkt sein. Das Thema Support kann sowohl positiv als auch negativ bewertet werden, da einerseits Support- und Wartungsaufwand bei der Nachnutzungsmethode viel geringer sind, andererseits jedoch die Abhängigkeit von anderen Organisationen bzw. Verwaltungen die Flexibilität weiter einschränken. Die größten Vorteile der Nachnutzung liegen bei den niedrigeren Kosten sowie eine schnelle Implementierung.

Die Vor- und Nachteile der jeweiligen Beschaffungsmethode werden in Tabelle 2 aufgeführt.

	Vorteile	Nachteile
<b>Eigenbeschaffung</b>	<ul style="list-style-type: none"> <li>+ Maßgeschneiderte Lösungen</li> <li>+ unabhängige Kontrolle über die Plattform</li> <li>+ Eine bessere Kontrolle über die Datensicherheit und den Datenschutz</li> <li>+ Flexiblere Anpassungen und Erweiterungen</li> </ul>	<ul style="list-style-type: none"> <li>– Die Einführung kann lange dauern</li> <li>– Hohe Anfangsinvestitionen</li> <li>– Höherer Aufwand bei Wartung und Support</li> </ul>
<b>Nachnutzung einer Bestandsplattform</b>	<ul style="list-style-type: none"> <li>+ Schnelle Implementierung</li> <li>+ kostengünstiger</li> <li>+ Umfassender externer Support</li> <li>+ Austausch vorhandener Lösungen, Erfahrungen und Best Practices</li> <li>+ Standardisierte Verträge</li> </ul>	<ul style="list-style-type: none"> <li>– Eingeschränkte Anpassungsmöglichkeiten</li> <li>– Abhängigkeit von anderen Verwaltungen</li> <li>– Ggf. komplexere Integration in eigene Systeme</li> <li>– Datenschutzrisiken</li> </ul>

Tabelle 2: Vor- und Nachteile der Beschaffungsmethoden

Falls keine geeignete Lösung am Markt gefunden wird, kann per Ausschreibung die passende Lösung beschafft werden. Bei einer Ausschreibung müssen die Vergabeunterlagen alle relevanten Informationen und Anforderungen enthalten, die bereits in der Vorbereitungsphase gesammelt wurden und die potenzielle Anbieter erfüllen müssen. Dazu gehören technische und funktionale Kriterien, Betriebs- und Qualitätskriterien, Leistungsbeschreibungen, Vertragsbedingungen und Bewertungskriterien.

Welche Kriterien und Spezifikationen bei der Beschaffung berücksichtigt werden sollten, wird im Kapitel 5.2.1 beschrieben.

### **5.1.3 Implementierungsphase**

Um sicherzustellen, dass der Implementierungsprozess strukturiert und erfolgreich verläuft, sollte die Einführung in mehreren Phasen geplant und vollzogen werden. Die Festlegung der Rechte und Pflichten der entwickelnden Mitarbeitenden im Vorfeld anhand von definierten Rollen ist ein wichtiger Schritt, um klare Verantwortlichkeiten zu schaffen und die Effizienz sowie die Qualität der Entwicklungsprozesse zu gewährleisten.

Auch die Festlegung klarer Regeln und Prozesse gehört zu den maßgeblichen Vorbereitungsschritten der Implementierungsphase. Es muss allen Beteiligten klar sein, wer etwas entwickeln darf bzw. wer etwas an den Entwicklungen ändern darf. Ein wesentlicher Aspekt dabei ist die Formalisierung des Freigabeprozesses für die entwickelten Anwendungen. Durch die Definition eines strukturierten Freigabeprozesses wird sichergestellt, dass alle Anwendungen vor ihrer Einführung gründlich geprüft werden. Dies umfasst die Überprüfung der Konformität mit rechtlichen Regelungen und internen Richtlinien.

Ein geregeltes Change-Management von Beginn an kann bei der Einführung einer Low-Code-Plattform für dessen Erfolg und Akzeptanz sehr entscheidend sein. Neue Technologien stoßen oft auf Widerstand, insbesondere, wenn sie bestehende Arbeitsweisen verändern. Ein effektives Change-Management hilft, die Mitarbeiter von den Vorteilen der Low-Code-Plattform zu überzeugen und ihre Bedenken zu adressieren. Außerdem minimiert es die Störungen im Tagesgeschäft, sorgt dafür, dass die Einführung der neuen Plattform reibungslos verläuft und stellt sicher, dass alle Mitarbeiter die notwendige Schulung und Unterstützung erhalten, um die neue Plattform effektiv zu nutzen. Dies reduziert die Lernkurve und erhöht die Produktivität.

Wenn diese Rahmenbedingungen realisiert sind, empfiehlt es sich zuerst mit einer Pilotimplementierung der Plattform anzufangen, um sie in einem kontrollierten Umfeld zu testen, die ersten Nutzenden zu schulen und erste Erfahrungen zu sammeln. Dies ermöglicht es, potenzielle Herausforderungen frühzeitig zu identifizieren und entsprechende Maßnahmen zu ergreifen. So können die ersten Anwendungen mithilfe der implementierten Plattform entwickelt werden.

Die Entwicklung sollte stets iterativ erfolgen und kontinuierlich angepasst und verbessert werden. Dazu müssen laufend Feedbacks sowohl in der Testphase als auch später in der produktiven Phase von den Nutzenden eingeholt und berücksichtigt werden.

Nach dem erfolgreichen Abschluss der Pilotphase erfolgt die schrittweise Ausweitung der Nutzung der Plattform auf weitere Vorhaben. Die in der Vorbereitungsphase identifizierten Use-Cases eignen sich hervorragend als Einstiegspunkt in die weitere Entwicklung. Dabei ist es wichtig, weiterhin kontinuierlich Feedback von den Nutzenden zu sammeln und die Plattform sowie die zugehörigen Prozesse entsprechend anzupassen. Um sicherzustellen, dass alle Nutzenden die notwendigen Kenntnisse und Fähigkeiten besitzen, sollten regelmäßige Schulungen und Workshops angeboten werden, um die Plattform effektiv zu nutzen.

## 5.2 Auswahl der richtigen Plattform

Die Auswahl der richtigen Plattform ist entscheidend für den Erfolg von Entwicklungsprojekten. Angesichts der Vielzahl von verfügbaren Optionen ist es wichtig, eine Plattform zu wählen, die den spezifischen Anforderungen und Zielen entspricht. In diesem Abschnitt werden die wesentlichen Kriterien und Überlegungen erläutert, die helfen, die beste Low-Code-Plattform für die jeweiligen Bedürfnisse zu identifizieren und auszuwählen.

Leistungskriterien	Funktionale Kriterien	Qualitätskriterien	Betriebskriterien
<ul style="list-style-type: none"> <li>• Leistungsfähigkeit</li> <li>• Skalierungsmöglichkeit</li> <li>• Schnittstellen</li> <li>• Unterstützung von APIs, Middleware und anderen Integrationslösungen</li> </ul>	<ul style="list-style-type: none"> <li>• Anzahl der Nutzenden</li> <li>• Testverfahren</li> <li>• Multi-User-Fähigkeit</li> <li>• Funktionalitäten</li> <li>• Anpassbarkeit und Flexibilität</li> <li>• Mobile Unterstützung (bei Bedarf)</li> </ul>	<ul style="list-style-type: none"> <li>• Datenschutz und -sicherheit</li> <li>• Schulungen und Dokumentation</li> <li>• Art des Projekts</li> <li>• User Stories</li> <li>• Benutzerfreundlichkeit</li> <li>• Barrierefreiheit</li> <li>• Kosten/Kostenmodell</li> </ul>	<ul style="list-style-type: none"> <li>• Anpassungsfähigkeit</li> <li>• Support</li> <li>• Wartbarkeit</li> <li>• Bereitstellungsmodell</li> <li>• Interoperabilität: Kompatibilität mit bestehenden Systemen und Protokollen</li> <li>• Mandantenfähigkeit (bei Bedarf)</li> </ul>

Abbildung 2: Kriterien zur Auswahl der richtigen Low-Code-Plattform

Im Vorfeld der Beschaffung einer neuen Low-Code-Plattform sollten die zu entwickelnden Anwendungen und die daraus resultierenden Anforderungen bestimmt werden. Zur Unterstützung kann hierbei die Norm ISO 25010 herangezogen werden. Sie enthält die Hauptkategorien Übertragbarkeit, Wartbarkeit, Effizienz, Funktionalität, Zuverlässigkeit und Benutzbarkeit, Kompatibilität und IT-Sicherheit.

Kategorien	Unterkategorien
<b>Funktionale Eignung</b>	Funktionale Vollständigkeit, Funktionale Korrektheit, Funktionale Angemessenheit
<b>Leistungseffizienz</b>	Zeitverhalten, Ressourcenverbrauch, Kapazität
<b>Kompatibilität</b>	Koexistenz, Interoperabilität
<b>Benutzbarkeit</b>	Erlernbarkeit, Bedienbarkeit, Benutzerfehlerkontrolle, Benutzeroberflächenästhetik, Zugänglichkeit
<b>Zuverlässigkeit</b>	Reife, Verfügbarkeit, Fehlertoleranz, Wiederherstellbarkeit
<b>IT-Sicherheit</b>	Vertraulichkeit, Integrität, Nichtabstreitbarkeit, Authentizität
<b>Wartbarkeit</b>	Modularität, Wiederverwendbarkeit, Analysierbarkeit, Modifizierbarkeit, Prüfbarkeit
<b>Übertragbarkeit</b>	Anpassbarkeit, Installierbarkeit, Austauschbarkeit

Tabelle 3: Kategorien und Unterkategorien des ISO 25010 Qualitätsmodells

Durch Definieren der Anforderungen können die relevanten Bewertungskriterien und die den eigenen Bedürfnissen entsprechenden Funktionen bei der Auswahl der richtigen Plattform berücksichtigt werden. Bietet z. B. die Plattform reaktive Designoptionen, einfache Integration von nativen Funktionen und vielleicht sogar Offline-Fähigkeiten? Ist sie mit den eigenen Betriebssystemen kompatibel? (z. B. MacOS, Android oder Linux). Auch der geplante Anwendungsbereich kann die Auswahl der richtigen Plattform beeinflussen. Für Webanwendungen sind andere Faktoren entscheidend, darunter Server-Performance, Datenbankintegration und vielleicht Auswertungsmöglichkeiten. Wenn es um die Automatisierung von Geschäftsprozessen geht, dann spielen Aspekte wie die einfache Modellierung von Prozessen, die Integration mit bestehenden Systemen und die Möglichkeit, komplexe Logik abzubilden, eine entscheidende Rolle.

Eine gute Methode, um verschiedene Low-Code-Plattformen miteinander zu vergleichen und die beste auszuwählen, ist die Durchführung einer Nutzwert-Analyse. So besteht die Möglichkeit, die festgelegten Kriterien zu berücksichtigen und sie hilft, die verschiedenen Plattformen systematisch zu bewerten. Die Kriterien werden dazu je nach Wichtigkeit mit einer Gewichtung versehen. Nach der Durchführung der Nutzwert-Analyse werden die Daten gesammelt und analysiert. Die Summe der gewichteten Bewertungen in einer Bewertungsmatrix kann, unter Berücksichtigung der Einhaltung der Ausschlusskriterien und Mindestanforderungen, ein geeigneter Messwert zum Vergleich der einzelnen Plattformen sein.

Folgenutzen sollten dabei ebenfalls betrachtet werden. Dies können z. B. Einsparungen durch schnellere Entwicklungszyklen und die Reduktion von Outsourcing-Kosten oder aber auch verbesserte Mitarbeiterzufriedenheit, erhöhte Flexibilität und eine schnellere Markteinführung neuer Produkte sein.

In der Nutzwert-Analyse sollten auch die aus der Risikobewertung resultierenden Anforderungen, die während der Vorbereitungsphase ermittelt wurden, einbezogen werden. Technische Risiken wie die Kompatibilität mit bestehenden Systemen und Sicherheitsaspekte sowie organisatorische Risiken wie die Akzeptanz durch die Mitarbeiter und notwendige Schulungen sollten berücksichtigt werden.

## **5.2.1 Bewertungskriterien bei der Auswahl der richtigen Low-Code-Plattform**

Die Auswahl der richtigen Low-Code-Plattform kann eine Herausforderung sein. Durch die individuelle Berücksichtigung folgender Kriterien kann je nach Anforderung eine fundierte Entscheidung getroffen und eine geeignete Low-Code-Plattform ausgewählt werden, die am besten zu den Bedürfnissen des Unternehmens passt:

### **5.2.1.a Benutzerfreundlichkeit**

Die Benutzerfreundlichkeit ein wichtiger Faktor bei der Auswahl der richtigen Plattform. Eine benutzerfreundliche Plattform sollte eine intuitive und leicht verständliche Benutzeroberfläche bieten, die es sowohl technischen als auch nicht-technischen Benutzern ermöglicht, Anwendungen schnell und effizient zu erstellen. Wichtige Aspekte sind Drag-and-Drop-Funktionalitäten, vorgefertigte Vorlagen und eine klare Navigation. Zudem sollte die Plattform umfassende Schulungs- und Supportmöglichkeiten bieten, um den Einstieg zu erleichtern und die Produktivität zu steigern. Eine gute Benutzerfreundlichkeit trägt dazu bei, die Akzeptanz der Plattform im Unternehmen zu erhöhen und die Entwicklungszeit zu verkürzen.

### 5.2.1.b **Skalierbarkeit**

Eine skalierbare Plattform sollte in der Lage sein, mit dem Wachstum des Unternehmens mitzuwachsen und steigende Anforderungen zu bewältigen. Dies umfasst die Fähigkeit, eine zunehmende Anzahl von Benutzern und Anwendungen zu unterstützen, ohne dass die Leistung beeinträchtigt wird. Zudem sollte die Plattform flexible Integrationsmöglichkeiten bieten, um nahtlos mit bestehenden und zukünftigen Systemen zusammenzuarbeiten. Eine gute Skalierbarkeit stellt sicher, dass die Plattform langfristig genutzt werden kann und Investitionen in die Technologie nachhaltig sind. Auch die Möglichkeit, zusätzliche Funktionen und Module hinzuzufügen, ohne die bestehende Infrastruktur zu überlasten, ist wichtig. So wird die Flexibilität gewährleistet und kann auf Veränderungen schnell reagiert werden.

### 5.2.1.c **Support und Community**

Ein guter Kundensupport ist unerlässlich, um bei Problemen und Fragen schnell Hilfe zu erhalten. Eine große und aktive Community bietet zahlreiche Vorteile. Sie stellt eine Fülle an Ressourcen wie Tutorials, Foren und Dokumentationen bereit, die den Lernprozess erleichtern und die Produktivität steigern können. Zudem ermöglicht eine große Community den Austausch von Best Practices und die schnelle Lösung von Problemen durch den Zugang zu einem breiten Netzwerk von Anwendern und Experten. Eine starke Community fördert auch die kontinuierliche Weiterentwicklung der Plattform durch Feedback und Beiträge der Community-Mitglieder. Dadurch wird die Implementierung und Nutzung der Low-Code-Plattform erheblich unterstützt und verbessert.

### 5.2.1.d **Art des Projekts**

Auch die Art des Projekts spielt bei der Auswahl der richtigen Plattform bzw. Anbieters eine wichtige Rolle. Für die Einführung einer Low-Code-Plattform ist ein agiles Projekt oft sinnvoller, da die Plattform selbst Flexibilität und schnelle Anpassung ermöglicht. Agile Methoden unterstützen die iterative Entwicklung und Anpassung, was besonders bei der Implementierung neuer Technologien vorteilhaft ist.

### 5.2.1.e **Schnittstellen und Integrierbarkeit in bestehenden Systemen**

Eine Plattform, die nahtlos mit den vorhandenen Systemen und Datenbanken eines Unternehmens integriert werden kann, erleichtert die Datenmigration und den Datenaustausch erheblich. Dies reduziert nicht nur den Implementierungsaufwand, sondern minimiert auch das Risiko von Dateninkonsistenzen und Systemausfällen.

Zudem ist es wichtig, dass die Plattform eine breite Palette von Schnittstellen unterstützt, um verschiedene Anwendungsfälle abzudecken. Eine Plattform, die eine Vielzahl von Schnittstellen unterstützt, ist zukunftssicherer, da sie sich leichter an neue Technologien und Anforderungen anpassen lässt. Durch die Unterstützung zahlreicher Schnittstellen können Anwendungen schneller entwickelt und implementiert werden, da weniger Zeit für die Integration und Anpassung benötigt wird. Schließlich sollten die Schnittstellen der Plattform sicher sein, um den Schutz sensibler Daten zu gewährleisten und Compliance-Anforderungen zu erfüllen.

Die Möglichkeit, Daten und Anwendungen einfach zu importieren und zu exportieren, ist ebenfalls wichtig für die Integration und Migration. Dies erleichtert den Wechsel von einer Plattform zur anderen und die Integration in bestehende Systeme.

### 5.2.1.f Anpassungsfähigkeit

Die Plattform sollte an spezifische Bedürfnisse und Anforderungen angepasst werden können. Dies umfasst die Möglichkeit, benutzerdefinierte Funktionen und Designs zu erstellen. Anpassungsfähigkeit ermöglicht es, die Plattform optimal an die Geschäftsprozesse anzupassen. Dies erhöht die Effizienz und Effektivität der entwickelten Anwendungen. Eine flexible Plattform kann besser auf Veränderungen reagieren.

### 5.2.1.g Bereitstellungsmodell der Plattform

Auch das Bereitstellungsmodell der Plattform sollte bei der Auswahl berücksichtigt werden. Die gängigsten Modelle sind On-Premises (Installation auf lokalen Rechnern oder Servern erforderlich), webbasiert (Nutzung über einen Webbrowser), cloudbasiert (ähnlich wie webbasiert, aber sie kann auch zusätzliche Dienste und Speicherlösungen in der Cloud umfassen) und hybrid (eine Kombination aus On-Premises- und cloudbasierten Lösungen).

Bei On-Premises-Modellen hat man die volle Kontrolle über die Daten und Systeme, was besonders für Branchen mit strengen Datenschutzanforderungen wichtig ist und können ohne Internetverbindung benutzt werden. Dafür gestalten sich die Einrichtung und Wartung kostenintensiver als die anderen Modelle, da man dafür eigene Hardware und Personal benötigt.

Cloudmodelle können ohne große Investitionen relativ flexibel eingesetzt werden, sind wartungsarm und sind skalierbar. Die Nachteile dabei sind die Abhängigkeit von einer stabilen Internetverbindung und eine meist eingeschränkte Anpassbarkeit der Plattform. Da die Daten bei diesem Modell extern in der Cloud gespeichert werden, könnte es bei sensiblen Daten zu Sicherheitsbedenken führen. Webbasierte Modelle ähneln sich im Prinzip den Cloudmodellen, da man auch hier auf eine Internetverbindung angewiesen ist. Jedoch, besteht hier die Möglichkeit, die Daten lokal (z. B. auf einem Unternehmensserver) zu hosten. Sowohl bei cloud- als auch webbasierten Modellen sollten die eventuell langfristig hohen Kosten (Abo-Gebühren) berücksichtigt werden, obwohl die anfänglichen Kosten niedriger sein können.

Hybride Low-Code-Modelle bieten Flexibilität und schnelle Entwicklung, da sie die Vorteile von Low-Code und Full-Code kombinieren. Sie sind oft kosteneffizient und leicht skalierbar, da sie sowohl Cloud- als auch lokale Ressourcen nutzen. Allerdings können sie komplex in der Verwaltung sein und zusätzliche Sicherheitsrisiken mit sich bringen. Zudem können die anfänglichen Implementierungskosten höher und die Wartung aufwändiger sein.

	Vorteile	Nachteile
On-Premises	<ul style="list-style-type: none"><li>+ Bessere Kontrolle und Sicherheit</li><li>+ Gute Anpassbarkeit</li><li>+ Keine Internetverbindung benötigt</li></ul>	<ul style="list-style-type: none"><li>- Hohe Anfangsinvestitionen</li><li>- Wartungsaufwand</li><li>- komplexe/teure Skalierbarkeit</li></ul>
Cloudbasiert/Webbasiert	<ul style="list-style-type: none"><li>+ Geringe Anfangsinvestitionen</li><li>+ Einfache Skalierung</li><li>+ Wartungsarm</li><li>+ lokaler Betrieb möglich (webbasiert auf eigenem Server)</li></ul>	<ul style="list-style-type: none"><li>- Internetverbindung notwendig</li><li>- Sicherheitsbedenken</li><li>- Eingeschränkte Anpassbarkeit</li><li>- Leistungsprobleme bei hoher Serverauslastung</li></ul>
Hybrid	<ul style="list-style-type: none"><li>+ Flexibilität</li><li>+ Skalierbarkeit und Kontrolle</li><li>+ Kostenoptimierung</li></ul>	<ul style="list-style-type: none"><li>- Komplexität</li><li>- Sicherheitsrisiken</li><li>- Hohe Anfangsinvestitionen</li></ul>

Tabelle 4: Vor- und Nachteile der Bereitstellungsmodelle

### 5.2.1.h **Bereitstellungsoptionen der entwickelten Anwendungen**

Die Bereitstellungsmöglichkeiten für Anwendungen, die auf Low-Code-Plattformen entwickelt wurden, sind vielfältig und bieten Unternehmen Flexibilität und Anpassungsfähigkeit. Ähnlich zu den Bereitstellungsmodellen der Plattformumgebung gibt es auch Unterschiede bei den Bereitstellungsoptionen der entwickelten Anwendungen. Anwendungen können entweder innerhalb der Plattform veröffentlicht oder separat bereitgestellt werden. Darüber hinaus stehen verschiedene Hosting-Optionen zur Verfügung, darunter cloudbasierte Bereitstellung, Betrieb auf hybriden Systemen oder Hosting auf eigenen Servern. Eine Plattform kann durchaus auch mehrere der Bereitstellungsoptionen anbieten.

### 5.2.1.i **Kosten und Preismodell**

Die Preisstruktur der Plattform sollte transparent sein. Es ist wichtig, sowohl die anfänglichen Kosten als auch laufende Gebühren zu berücksichtigen. Versteckte Kosten, wie zusätzliche Gebühren für bestimmte Funktionen oder Benutzer. Ein gutes Preis-Leistungs-Verhältnis ist entscheidend für die langfristige Nutzung der Plattform. Kosten sollten immer im Verhältnis zum gebotenen Nutzen stehen.

Low-Code-Plattformen bieten eine Vielzahl von Preismodellen, die sich je nach Anbieter und den spezifischen Anforderungen der Nutzer unterscheiden. Ein gängiges Modell ist das abonnementbasierte Modell, bei dem monatliche oder jährliche Gebühren anfallen. Diese Modelle bieten unterschiedliche Stufen, die sich in den angebotenen Funktionen und der Anzahl der Benutzer unterscheiden.

Ein weiteres Modell ist das nutzungsbasierte Modell, bei dem die Kosten basierend auf der tatsächlichen Nutzung der Plattform berechnet werden. Dieses Modell ermöglicht eine flexible Skalierung der Kosten entsprechend der tatsächlichen Nutzung. Einige Anbieter bieten auch die Möglichkeit, eine einmalige Lizenzgebühr zu zahlen, die den Zugang zur Plattform und deren Funktionen auf unbestimmte Zeit ermöglicht. Dies kann besonders für Unternehmen attraktiv sein, die langfristige Projekte planen.

Viele Low-Code-Plattformen bieten zudem ein „Freemium“-Modell an, bei dem eine kostenlose Basisversion mit grundlegenden Funktionen zur Verfügung steht. Dieses Modell ermöglicht es Nutzenden, die Plattform auszuprobieren, bevor sie sich für ein kostenpflichtiges Upgrade entscheiden.

Für große Unternehmen bieten viele Plattformen maßgeschneiderte Enterprise-Modelle an, die auf die spezifischen Bedürfnisse und Anforderungen des Unternehmens zugeschnitten sind. Diese Modelle können verschiedene Varianten umfassen, wie benutzerbasierte Abrechnung, bei der die Kosten basierend auf der Anzahl der Benutzer berechnet werden, oder ressourcenbasierte Abrechnung, bei der die Nutzung von Ressourcen wie Rechenleistung und Speicherplatz berücksichtigt wird. Diese Modelle sind oft für große Unternehmen attraktiv, die eine umfassende Lösung benötigen.

### 5.2.1.j **Datenschutz und -sicherheit**

Eine Plattform muss sicherstellen, dass sensible Daten geschützt und die Datenschutzbestimmungen eingehalten werden. Dies umfasst die Implementierung von Verschlüsselungstechnologien, um Daten während der Übertragung und Speicherung zu schützen. Zudem sollte die Plattform regelmäßige Sicherheitsupdates und -prüfungen durchführen, um potenzielle Schwachstellen zu identifizieren und zu beheben. Ein weiterer wichtiger Punkt ist die Einhaltung der Datenschutz-Grundverordnung (DSGVO) und anderer relevanter Datenschutzgesetze, um rechtliche Risiken zu minimieren. Schließlich sollte die Plattform Mechanismen zur Verwaltung von Zugriffsrechten bieten, um sicherzustellen, dass nur autorisierte Personen auf sensible Daten zugreifen können. Die Security-Funktionen und Security-Bausteine, die die Low-Code-Plattform anbietet, sollten daher von Experten des Unternehmens geprüft werden.

Auch das Thema Datenbankmanagement-Funktionen spielt für Datenschutz und -Sicherheit eine wichtige Rolle. Damit dies gewährleistet ist, sollte die Plattform die Möglichkeit bieten, Datenbanken zu trennen und zu verwalten. Bei der Datenbearbeitung auf der Low-Code-Plattform muss zwischen der Art der betroffenen Daten differenziert werden. Gesundheitsdaten benötigen z. B. einen anderen Schutzbedarf als Daten zur Arbeitsorganisation (z. B. Raumbuchung). Dies würde auch Auswirkungen auf das Bereitstellungsmodell haben, da Gesundheitsdaten in der Cloud sehr hohe Sicherheitsanforderungen erfordern.

#### 5.2.1.k **Multi-User-Fähigkeit**

Die Plattform sollte die Zusammenarbeit mehrerer Benutzer gleichzeitig unterstützen. Dies ist besonders wichtig für Teams, die gemeinsam an Projekten arbeiten. Funktionen wie Versionskontrolle und Benutzerrechteverwaltung sind hierbei entscheidend. Multi-User-Fähigkeit fördert die Zusammenarbeit und Effizienz. Sie ermöglicht es, Projekte schneller und koordinierter abzuschließen.

#### 5.2.1.l **Funktionalität**

Die Plattform sollte eine breite Palette an Funktionen bieten, die den gesamten Entwicklungsprozess unterstützen. Dazu gehören Drag-and-Drop-Builder, vorgefertigte Vorlagen und umfangreiche Integrationsmöglichkeiten. Es ist wichtig, dass die Plattform sowohl einfache als auch komplexe Anwendungen unterstützen kann. Erweiterte Funktionen wie Workflow-Automatisierung und Datenvisualisierung sind ebenfalls von Vorteil. Eine umfassende Funktionalität ermöglicht es, vielfältige Geschäftsanforderungen abzudecken.

#### 5.2.1.m **Mobile Unterstützung**

Die Plattform sollte die Entwicklung von mobilen Anwendungen unterstützen. Mobile Unterstützung ermöglicht es, Anwendungen für eine breite Benutzerbasis zugänglich zu machen. Dies ist besonders relevant in einer zunehmend mobilen Welt. Eine gute mobile Unterstützung erhöht die Reichweite und Flexibilität der Anwendungen.

#### 5.2.1.n **Leistungsfähigkeit**

Die Performance der Plattform ist entscheidend, insbesondere bei der Verarbeitung großer Datenmengen und komplexer Anwendungen. Eine hohe Leistungsfähigkeit stellt sicher, dass Anwendungen schnell und zuverlässig laufen. Dies ist wichtig für die Benutzerzufriedenheit und die Effizienz der Geschäftsprozesse. Leistungsprobleme können zu Frustration und Produktivitätsverlust führen. Eine leistungsstarke Plattform bietet eine stabile und zuverlässige Basis für die Entwicklung.

#### 5.2.1.o **Flexibilität**

Die Plattform sollte flexibel genug sein, um verschiedene Arten von Anwendungen zu unterstützen. Dies umfasst einfache Formulare bis hin zu komplexen Workflows. Flexibilität ermöglicht es, die Plattform für unterschiedliche Geschäftsanforderungen zu nutzen. Dies erhöht den Wert und die Vielseitigkeit der Plattform. Eine flexible Plattform kann besser auf sich ändernde Anforderungen reagieren.

#### 5.2.1.p **Schulung**

Bei der Auswahl einer Low-Code-Plattform spielt die Schulung der Mitarbeiter eine entscheidende Rolle. Eine umfassende und effektive Schulung ist unerlässlich, um sicherzustellen, dass die Mitarbeiter die Plattform optimal nutzen können und die gewünschten Ergebnisse erzielt werden.

Ein wichtiger Aspekt bei der Bewertung einer Low-Code-Plattform ist daher das Schulungsangebot des Anbieters. Dies umfasst nicht nur die Qualität und Umfang der Schulungsmaterialien, sondern auch die Verfügbarkeit von praxisorientierten Trainings und Support. Die Technologie entwickelt sich ständig weiter, und regelmäßige Updates und Schulungen sind notwendig, um sicherzustellen, dass die Mitarbeiter immer auf dem neuesten Stand sind und die Plattform optimal nutzen können. Daher sollte die Weiterbildung kontinuierlich angeboten werden.

Darüber hinaus sollte die Schulung darauf abzielen, den Mitarbeitern nicht nur die technischen Fähigkeiten zu vermitteln, sondern auch ein tiefes Verständnis für die Anwendung der Plattform in ihren spezifischen Arbeitsprozessen. Dies hilft, die Effizienz und Produktivität zu steigern und Fehler zu minimieren.

## **5.3 Implementierung (Anwendungsentwicklung)**

Die Etablierung eines Governance-Frameworks, das klare Richtlinien und Standards für die Nutzung der Low-Code-Plattformen definiert, ist von großer Bedeutung. Dieses Framework sollte Mechanismen zur Überwachung und Kontrolle der Plattformnutzung beinhalten, um sicherzustellen, dass die entwickelten Anwendungen den Qualitäts- und Datenschutz-Sicherheitsanforderungen entsprechen.

Das Governance-Framework sollte auch klare Vorgaben zu dem Ablauf der Freigabe von Entwicklungen beinhalten. Jede Entwicklung muss von einem Gremium, z. B. Personalrat o. ä., geprüft und freigegeben werden, bevor sie veröffentlicht wird.

Bevor eine entwickelte Anwendung produktiv benutzt werden kann, sollte sie gründlich getestet werden. Dazu braucht es eines umfassenden Testplans. Manche Plattformen bieten auch automatisierte Validierungsmechanismen des erzeugten Codes. Durch gründliches Testen können Probleme im Vorfeld rechtzeitig identifiziert und behoben werden.

### **5.3.1 Anforderungen bei der Anwendungsentwicklung**

Bei der Entwicklung der Anwendungen auf einer Low-Code-Plattform gibt es Anforderungen, die beachtet werden sollten, um sicherzustellen, dass die Anwendung robust, sicher und skalierbar ist.

Die wichtigsten Anforderungen sind:

#### **5.3.1.a Mandantenfähigkeit**

Die Mandantenfähigkeit ermöglicht, dass verschiedene Behörden oder Abteilungen innerhalb einer gemeinsamen IT-Infrastruktur arbeiten können, ohne dass ihre Daten vermischt werden. Dies ist besonders relevant, da öffentliche Verwaltungen oft komplexe Strukturen mit unterschiedlichen Zuständigkeiten und sensiblen Daten haben.

Die Mandantenfähigkeit der entwickelten Anwendungen wird benötigt, wenn damit sensible Daten verarbeitet werden sollen. Dies ist bei der Nutzung von Gesundheitsdaten der Fall, da sie strengen Datenschutzbestimmungen unterliegen. Eine mandantenfähige Anwendung stellt sicher, dass die Daten jedes Mandanten isoliert und geschützt sind, wodurch unbefugter Zugriff verhindert wird.

Dies ist entscheidend, um die rechtlichen Anforderungen zu erfüllen und die Sicherheit der Daten zu gewährleisten. Zudem ermöglicht die Mandantenfähigkeit eine flexible und skalierbare Nutzung der Anwendung, sodass unterschiedliche Einrichtungen ihre spezifischen Anforderungen individuell umsetzen können. Durch differenzierte Zugriffsrechte und Rollenmanagement wird sichergestellt, dass nur autorisierte Personen Zugriff auf die sensiblen Gesundheitsdaten haben.

Daher muss vor der Entwicklung einer Anwendung geprüft werden, welche Daten damit verarbeitet werden sollen und ob die Anwendung mandantenfähig sein muss.

### 5.3.1.b    **Datenschutz und -sicherheit**

Hierzu gehören Maßnahmen wie Datenverschlüsselung, Authentifizierung und Autorisierung, um sensible Daten zu schützen und unbefugten Zugriff zu verhindern. Die Nutzung eines einheitlichen Identity Providers zur Speicherung und Verwaltung digitaler Identitäten der Nutzenden und ein robustes Rollen- und Rechtekonzepts können dabei zur Erfüllung der Sicherheitsanforderungen beitragen.

Die Implementierung von Sicherheitsprotokollen und regelmäßige Sicherheitsüberprüfungen sind essenziell, um potenzielle Schwachstellen zu identifizieren und zu beheben. Darüber hinaus ist die Einhaltung von Compliance-Anforderungen wie der Datenschutz-Grundverordnung (DSGVO) von großer Bedeutung, um rechtliche Risiken zu minimieren und den Schutz personenbezogener Daten zu gewährleisten. Dies umfasst auch die regelmäßige Schulung der Mitarbeiter in Bezug auf Sicherheitspraktiken und Datenschutzrichtlinien.

### 5.3.1.c    **Interoperabilität**

Ein weiterer wichtiger Punkt ist die Interoperabilität. Die Anwendung sollte Standardprotokolle und -schnittstellen unterstützen, um eine nahtlose Integration mit anderen Systemen und Plattformen zu ermöglichen. Dies erleichtert die Zusammenarbeit und den Datenaustausch zwischen verschiedenen Anwendungen und stellt sicher, dass die Anwendung in bestehende IT-Landschaften integriert werden kann. Interoperabilität ist besonders wichtig in Umgebungen, in denen verschiedene Systeme und Anwendungen miteinander kommunizieren müssen, um Geschäftsprozesse zu unterstützen.

Im Öffentlichen Gesundheitsdienst gibt es mehrere wichtige Schnittstellen und Standards, die eine effiziente und sichere Kommunikation sowie den Schutz sensibler Gesundheitsdaten gewährleisten. Bei der Anwendungsentwicklung sollten möglichst diese Schnittstellen und Standards berücksichtigt werden. Zu den wichtigsten Standards gehören die X-Standards (z. B. XÖV und XLeitstelle) sowie FHIR-Standards (Fast Healthcare Interoperability Resources). Bei den Schnittstellen gehören DEMIS (Deutsches Elektronisches Melde- und Informationssystem für den Infektionsschutz) und eHealth-Schnittstellen (z. B. Telematik-Infrastruktur) zu den Bekanntesten.

### 5.3.1.d    **Benutzerfreundlichkeit**

Die Benutzerfreundlichkeit der Anwendung darf ebenfalls nicht vernachlässigt werden. Intuitive Benutzeroberflächen und einfache Bedienbarkeit fördern die Akzeptanz und Produktivität der Nutzenden. Zu den Nutzenden können je nach Anwendung interne aber auch externe Nutzenden gehören. Eine gute Benutzererfahrung trägt dazu bei, dass die Anwendung effektiv genutzt wird und die Zufriedenheit der Benutzer steigt. Usability-Tests und Benutzerfeedback sind wichtige Instrumente, um die Benutzerfreundlichkeit kontinuierlich zu verbessern. Beispielsweise können Prototypen und Mockups verwendet werden, um frühzeitig Feedback von Nutzenden zu erhalten und die Benutzeroberfläche entsprechend anzupassen.

Auf die Barrierefreiheit der entwickelten Anwendungen muss in der Öffentlichen Verwaltung und im ÖGD besonders geachtet werden. Dazu gehören Lösungen für seh- und hörbehinderte Menschen.

Damit die Barrierefreiheit und Benutzerfreundlichkeit in allen Entwicklungen berücksichtigt werden, bedarf es klare Leitlinien, die von allen Entwicklern zu befolgen sind und im Freigabeprozess jeder Anwendung geprüft werden sollten.

Bei der Festlegung der Benutzerfreundlichkeitsleitlinien können zur Orientierung die DIN-Normen ISO 9241 sowie ISO 13407 herangezogen werden. ISO 9241 befasst sich mit Aspekten wie Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Erwartungskonformität, Erlernbarkeit, Steuerbarkeit und Robustheit gegen Benutzerfehler. ISO 13407 beschreibt den benutzerorientierten

Gestaltungsprozess für interaktive Systeme und umfasst Grundsätze wie aktive Beteiligung der Nutzenden, Verständnis von Nutzenden- und Aufgabenanforderungen, geeignete Funktionsaufteilung zwischen Nutzenden und Technik, Iteration von Gestaltungslösungen und Multidisziplinäre Gestaltung.

#### 5.3.1.e **Wartbarkeit**

Eine gute Wartbarkeit stellt sicher, dass die Anwendung einfach aktualisiert und gewartet werden kann, um langfristig funktionsfähig zu bleiben. Dies umfasst die Dokumentation des Codes, die Implementierung von Best Practices für die Softwareentwicklung und die regelmäßige Überprüfung und Aktualisierung der Anwendung. Wartungsfreundliche Anwendungen sind leichter zu pflegen und anzupassen, was die Lebensdauer der Software verlängert und die Gesamtbetriebskosten senkt.

#### 5.3.1.f **Dokumentation**

Eine gute Dokumentation verbessert die Zusammenarbeit im Team, da alle Beteiligten ein gemeinsames Verständnis der Projektziele und -anforderungen haben. Sie erleichtert die Wartung und Weiterentwicklung der Anwendung, da Entwickler schneller Fehler beheben und Änderungen vornehmen können. Neue Teammitglieder können sich schneller einarbeiten, was die Produktivität erhöht. Zudem stellt eine gründliche Dokumentation sicher, dass die Anwendung den gesetzlichen und sicherheitstechnischen Anforderungen entspricht und das Wissen über die Anwendung für das gesamte Team zugänglich bleibt.

# 6

## ZUSAMMENFASSUNG DER ERKENNTNISSE (FAZIT)

Low-Code ist ein Ansatz zur Softwareentwicklung, der auch Personen ohne tiefgehende Programmierkenntnisse ermöglicht, Anwendungen mit minimalem manuellem Programmieraufwand zu erstellen. Dies geschieht hauptsächlich durch eine visuelle Benutzeroberfläche, bei der Entwickler vorgefertigte Bausteine per Drag-and-Drop zusammenstellen können.

Low-Code-Plattformen bieten erhebliche Potenziale zur Optimierung von Geschäftsprozessen, Förderung von Innovationen und Skalierung der IT-Kapazitäten. Das ist gerade im Gesundheitswesen von Bedeutung, einem Sektor, der unter steigender Komplexität, Fachkräftemangel und Kostendruck leidet.

Allerdings sollte die Einführung einer Low-Code-Plattform nicht isoliert zur Umsetzung eines Einzelprojekts erfolgen, sondern als strategische Entscheidung für die digitale Transformation der gesamten Behördenlandschaft. Denn nur wenn die Implementierung, Wartung und Betrieb der Plattform langfristig und strategisch geplant werden, lassen sich diese Vorteile auch vollständig ausschöpfen.

Vorteile liegen z. B. darin, dass durch die benutzerfreundliche Oberfläche und die Drag-and-Drop-Funktionalität können auch Mitarbeitende ohne tiefgehende Programmierkenntnisse Applikationen erstellen und anpassen. Ein wesentlicher Vorteil von Low-Code-Plattformen ist die hohe Geschwindigkeit bei der Entwicklung von Anwendungen. Dies ermöglicht es, schneller auf sich ändernde Anforderungen zu reagieren und innovative Lösungen zu implementieren. So können wiederkehrende Prozesse automatisiert und die Mitarbeitenden entlastet werden. Zudem fördern Low-Code-Plattformen die Zusammenarbeit zwischen verschiedenen Abteilungen, da sie das Silo-Denken aufbrechen und eine engere Abstimmung zwischen IT und Fachbereichen ermöglichen. Dies ist besonders im Gesundheitswesen von Vorteil, wo eine enge Zusammenarbeit zwischen medizinischem Personal, Verwaltung und IT-Abteilungen unerlässlich ist.

Ein weiterer Vorteil ist die Möglichkeit, durch den Einsatz von Low-Code-Plattformen die Produktivität zu steigern und die Ausrichtung der IT an den Geschäftsanforderungen zu verbessern. Darüber hinaus können Low-Code-Plattformen zur Digitalisierung und Automatisierung von Prozessen beitragen, was zu einer verbesserten Patientenversorgung und optimierten Abläufen führt. Beispielsweise können administrative Aufgaben wie Terminplanung, Patientenregistrierung und Abrechnung automatisiert werden, wodurch das medizinische Personal entlastet und mehr Zeit für die Patientenversorgung gewonnen wird. Und zuletzt die Möglichkeit der Integration von Code in die Low-Code-Anwendungen macht Low-Code-Plattformen flexibler und vielseitiger als No-Code-Plattformen.

Trotz der vielen Vorteile gibt es auch einige Herausforderungen bei der Nutzung von Low-Code-Plattformen im Gesundheitswesen. Eine der größten Herausforderungen ist die Abhängigkeit von den Plattformanbietern. Dies kann zu Problemen führen, wenn der Anbieter seine Dienste einstellt oder die Preise erhöht. Eine weitere Herausforderung ist die begrenzte Flexibilität und Anpassungsfähigkeit der Plattformen gegenüber der klassisch entwickelten Anwendungen. Obwohl Low-Code-Plattformen viele vorkonfigurierte Komponenten und Vorlagen bieten, können sie nicht immer die spezifischen Anforderungen und komplexen Prozesse vollständig abbilden. Dies kann zu Einschränkungen bei der Implementierung und Nutzung führen. Beispielsweise könnten spezielle Anforderungen an die Datensicherheit und den Datenschutz, die im Gesundheitswesen besonders streng sind, nicht vollständig erfüllt werden.

Die Einführung der Low-Code-Plattform sollte gut strukturiert und nach einer durchdachten Planung durchgeführt werden. Damit das klappt, sollten alle relevanten Stakeholder frühzeitig in den Planungsprozess eingebunden werden, um sicherzustellen, dass alle wichtigen Anforderungen gesammelt und berücksichtigt werden und so ihre Erwartungen erfüllt werden. Der Beschaffungsprozess bedarf eine gründliche Vorbereitung. Während der Vorbereitung werden alle Aspekte detailliert betrachtet und abgewogen. Dazu gehören u. a. Anforderungen, Anwendungsfälle, Stakeholder, Risiken, Kosten und Nutzen. Sollte in der Vorbereitungsphase die Einführung der Low-Code-Plattform als sinnvoll erachtet und genehmigt werden, müssen die Vergabeschritte durchlaufen werden. Zum Schluss wird eine geeignete Low-Code-Plattform ausgewählt und eingeführt. Die schrittweise Einführung und Ausweitung der Nutzung wird wie in diesem Papier beschrieben empfohlen.

Zusammenfassend lässt sich sagen, dass Low-Code-Plattformen im Gesundheitswesen ein großes Potenzial bieten, die Digitalisierung voranzutreiben und die Effizienz zu steigern. Die Vorteile, wie die erhöhte Entwicklungsgeschwindigkeit, die Förderung der Zusammenarbeit und die Verbesserung der Produktivität, überwiegen die Nachteile. Dennoch sollten Gesundheitsorganisationen die Herausforderungen und Risiken sorgfältig abwägen und eine fundierte Entscheidung treffen, ob und wie sie Low-Code-Plattformen in ihre IT-Strategie integrieren möchten. Es ist wichtig, eine umfassende Analyse der spezifischen Anforderungen und Rahmenbedingungen durchzuführen und gegebenenfalls zusätzliche Maßnahmen zur Sicherstellung der Datensicherheit und -integrität zu ergreifen.

## 7.1 Quellen

### **Abgeordnetenhaus von Berlin, Drucksache 19/1633**

28.03.2024; <https://pardok.parlament-berlin.de/starweb/adis/citat/VT/19/DruckSachen/d19-1633.pdf>; S.4 (Stand vom 13.04.2025)

### **Binder, Florian: So wird die Einführung einer Low-Code-Plattform ein Erfolg**

In IT-Daily.net; <https://www.it-daily.net/it-management/business-software/erfolgskriterien-fuer-die-einfuehrung-von-low-code-plattformen>  
[https://www.pd-g.de/assets/PD-Veroeffentlichungen/241031\\_PD-Veroeffentlichung\\_Einsatz\\_von\\_Low-Code-Plattformen\\_zur\\_Optimierung\\_von\\_Foerdermittelmanagement-Prozessen\\_auf\\_kommunaler\\_Ebene.pdf](https://www.pd-g.de/assets/PD-Veroeffentlichungen/241031_PD-Veroeffentlichung_Einsatz_von_Low-Code-Plattformen_zur_Optimierung_von_Foerdermittelmanagement-Prozessen_auf_kommunaler_Ebene.pdf) (Stand vom 13.04.2025)

### **Baumgarten, C., Endl, R. & Stich, S.: Professionelle Softwareentwicklung mit Low Code optimieren – eine Fallstudie.**

HMD 61, 1213–1234 (2024); <https://doi.org/10.1365/s40702-024-01095-y>  
(Stand vom 13.04.2025)

### **CAFM – CAFM-Blog: Risiken und Chancen von Low-Code/No-Code-Plattformen**

<https://www.cafm-blog.de/blog/risiken-und-chancen-von-low-code-no-code-plattformen/>  
(Stand vom 13.04.2025)

### **Der Regierende Bürgermeister – Berliner Verwaltungspreis: Innovationslandkarte**

<https://www.berlin.de/verwaltungspreis/innovationslandkarte/index.php/detail/363>  
(Stand vom 13.04.2025)

### **Elshan, E., Binzer, B.: Mehr als ein Trend?: Wie Low-Code die digitale Transformation unterstützt.**

HMD 61, 1070–1087 (2024); <https://doi.org/10.1365/s40702-024-01107-x>  
(Stand vom 13.04.2025)

### **engomo: The Future is Low-Code**

[https://ressourcen.engomo.com/hubfs/Whitepaper/engomo\\_Whitepaper\\_the\\_future\\_is\\_Low-Code.pdf](https://ressourcen.engomo.com/hubfs/Whitepaper/engomo_Whitepaper_the_future_is_Low-Code.pdf) (Stand vom 13.04.2025)

### **Materna – Der Materna Blog: Low-Code als Schlüssel zur Digitalisierung: Was Sie darüber wissen sollten**

<https://blog.materna.de/low-code-als-schluessel-zur-digitalisierung/> (Stand vom 13.04.2025)

### **Mendix: Was ist Low-Code?**

<https://www.mendix.com/de/low-code-guide/#how-to-pick-a-lowcode-platform>  
(Stand vom 13.04.2025)

### **Ministerium für Arbeit, Soziales, Frauen und Gesundheit Saarland: Resilient, modern, nachhaltig gesichert – der ÖGD der Zukunft**

[https://www.saarland.de/masfg/DE/aktuelles/aktuelle-meldungen/aktuelle-meldungen\\_2024/aktuelle-meldungen\\_2024\\_12/aktuelle-meldungen\\_20241203\\_oegd#:~:text=,ausgerichtet%20an%20den%20spezifischen%20saarl%C3%A4ndischen](https://www.saarland.de/masfg/DE/aktuelles/aktuelle-meldungen/aktuelle-meldungen_2024/aktuelle-meldungen_2024_12/aktuelle-meldungen_20241203_oegd#:~:text=,ausgerichtet%20an%20den%20spezifischen%20saarl%C3%A4ndischen) (Stand vom 13.04.2025)

### Outsystems: Open source low-code: Is it the best choice for your business?

<https://www.outsystems.com/de-de/low-code/understanding-open-source-low-code-platforms/#open-source-low-code-platform-security> (Stand vom 13.04.2025)

### Presse-Service.de: Das Gesundheitsamt wird digitaler: Neue Software erleichtert die Arbeit bei Hygienekontrollen

<https://www.presse-service.de/data.aspx/static/1169314.html#:~:text=Bisher%20protokollierten%20die%20Hygienekontrolleur,und%20%C3%9Cberwachung%20oder%20M%C3%A4ngel%20nachverfolgen> (Stand: 13.04.2025)

### Serwa, Kamila: Handreichung – Wie wähle ich eine passende Low-Code/No-Code-Plattform aus?

Mittelstand-Digital Zentrum Hannover; [https://digitalzentrum-hannover.de/wp-content/uploads/2023/11/Handreichung\\_Low-Code\\_No-Code\\_23-08-07.pdf](https://digitalzentrum-hannover.de/wp-content/uploads/2023/11/Handreichung_Low-Code_No-Code_23-08-07.pdf) (Stand vom 13.04.2025)

### Timetoact Software Consulting: Welches Low Code Tool ist das richtige für mein Unternehmen?

<https://www.timetoact.de/blog/welches-low-code-tool-ist-das-richtige>  
<https://appmaster.io/de/blog/lebenszyklus-eines-low-code-projekts> (Stand vom 13.04.2025)

## 7.2 **Glossar**

Begriff	Definition
Audit	Eine systematische Überprüfung und Bewertung von Prozessen, Systemen oder Organisationen, um deren Konformität mit festgelegten Standards zu überprüfen.
Ausschlusskriterien	Kriterien, die dazu führen, dass bestimmte Alternativen oder Angebote von der weiteren Betrachtung ausgeschlossen werden.
Backend	Das Backend bezieht sich auf die serverseitige Datenverarbeitung und Logik.
BDSG	Bundesdatenschutzgesetz zum Schutz personenbezogener Daten.
Best-Practices	Bewährte Methoden und Verfahren, die sich in der Praxis als effektiv und effizient erwiesen haben.
Blended-Learning-Angebote	Bildungsangebote, die eine Kombination aus Präsenzunterricht und Online-Lernen umfassen.
CDISC	CDISC entwickelt eine Reihe von offenen Standards für den Austausch von Daten aus klinischen Studien. CDISC ist ein in den USA von der Food and Drug Administration (FDA) akzeptierter Standard für die Einreichung klinischer Daten bei der Arzneimittelzulassung. Zur Definition der Standards dienen XML-Schemata.
Change-Management	Prozess zur Steuerung und Unterstützung von Veränderungen innerhalb einer Organisation, insbesondere bei der Einführung neuer Technologien.
Compliance-Anforderungen	Vorschriften und Standards, die eingehalten werden müssen, um rechtliche und regulatorische Anforderungen zu erfüllen.
Compliance-Management-System	Ein integriertes System aus Richtlinien, Verfahren und Prozessen, das Organisationen dabei hilft, gesetzliche Vorschriften, interne Richtlinien und Branchenstandards einzuhalten.
Datenbankintegration	Prozess der Verbindung und Integration von Datenbanken mit anderen Systemen oder Anwendungen, um einen nahtlosen Datenaustausch zu ermöglichen.
Dateninkonsistenz	Zustand, bei dem Daten in verschiedenen Systemen oder Datenbanken nicht übereinstimmen oder widersprüchlich sind.

Datenintegrität	<i>Die Genauigkeit, Konsistenz und Zuverlässigkeit von Daten während ihres gesamten Lebenszyklus.</i>
Datenmigration	Der Prozess der Übertragung von Daten von einem System oder Format in ein anderes.
Datensynchronisation	Der Prozess, bei dem Daten zwischen verschiedenen Systemen oder Geräten abgeglichen und auf dem neuesten Stand gehalten werden.
Deploymentzyklen	Die Phasen, die eine Software durchläuft, von der Entwicklung über das Testen bis hin zur Bereitstellung und Wartung.
DICOM	Digital Imaging and Communications in Medicine (digitale Bildgebung und -kommunikation in der Medizin) ist ein offener Standard zur Speicherung und zum Austausch von Informationen im medizinischen Bilddatenmanagement.
Drag-and-Drop	Funktionalität, die es ermöglicht, Elemente durch Ziehen und Ablegen mit der Maus zu verschieben und zu organisieren.
Drag-and-Drop-UI-Builder	Ein Werkzeug, das es ermöglicht, Benutzeroberflächen durch einfaches Ziehen und Ablegen von Komponenten zu erstellen, ohne umfangreiche Programmierkenntnisse.
DSGVO	Datenschutz-Grundverordnung: Europäische Verordnung zum Schutz personenbezogener Daten.
ELfa-Maßnahme	Ein Land für Alle: Eine länderübergreifende Initiative, bei der mehrere Bundesländer zusammenarbeiten, um digitale Dienste zu entwickeln und bereitzustellen, die für alle beteiligten Länder relevant sind und in diesen angewendet werden sollen
ETL-Tools (Extract, Transform, Load)	Softwarelösungen zur Datenintegration, die Daten aus verschiedenen Quellen extrahieren, transformieren und in Zielsysteme laden.
FHIR	FHIR (Fast Healthcare Interoperability Resources, englisch gesprochen wie „fire“) ist ein Standard zum Datenaustausch zwischen verschiedenen IT Systemen im Gesundheitswesen. Der Standard bietet eine Schnittstelle (API), um die als „Resources“ bezeichneten Daten auszutauschen.
Framework	Struktur oder Gerüst, das als Grundlage für die Entwicklung von Softwareanwendungen dient und häufig vorgefertigte Komponenten und Bibliotheken enthält.
Frontend	Das Frontend stellt die benutzerseitige Präsentationsschicht der Anwendung dar.
Full-Code	Traditioneller Ansatz zur Softwareentwicklung, bei dem umfangreiche Programmierkenntnisse erforderlich sind, um maßgeschneiderte Softwarelösungen zu erstellen.
Governance	Governance definiert den Rahmen und die Prozesse, um sicherzustellen, dass IT-Ressourcenplanung effektiv genutzt werden, die Geschäftsziele unterstützen und Risiken minimiert und die Qualität und Compliance in der IT (und Softwareentwicklung) sichergestellt werden.
High Code	Traditioneller Ansatz zur Softwareentwicklung, bei dem umfangreiche Programmierkenntnisse erforderlich sind, um maßgeschneiderte Softwarelösungen zu erstellen.
HL7	HL7 definiert die Struktur der elektronischen Nachrichten, die zwischen Computersystemen im Gesundheitswesen ausgetauscht werden.
Identity Provider	System zur Verwaltung digitaler Identitäten und zur Authentifizierung von Nutzern.
IHE	IHE® steht für Integrating the Healthcare Enterprise und ist eine weltweite Initiative von Fachleuten des Gesundheitswesens. Ihr Ziel ist es, die Kommunikation zwischen IT-Systemen zu vereinheitlichen und zu verbessern.

Interoperabilität	<i>Die Fähigkeit verschiedener Systeme oder Softwarelösungen, miteinander zu kommunizieren und Daten auszutauschen.</i>
Iterative Entwicklung	Ansatz zur Softwareentwicklung, bei dem ein Projekt durch wiederholte Zyklen (Iterationen) und inkrementelle Verbesserungen fortschreitet. Jede Iteration umfasst Planung, Design, Implementierung und Testen.
Kanban	Ein visuelles System zur Verwaltung von Arbeitsabläufen, das die kontinuierliche Verbesserung und Effizienzsteigerung fördert.
Kosten-Nutzen-Analyse	Verfahren zur vergleichenden Bewertung von Objekten oder Handlungsalternativen, bei dem die zukünftigen Kosten und Nutzen eines Projekts bestimmt und miteinander verglichen werden.
Leistungsbeschreibung	Detaillierte Beschreibung der zu erbringenden Leistungen, die alle Funktions- und Leistungsanforderungen an die zu beschaffende Leistung benennt.
Middleware-Lösungen	Software, die als Vermittler zwischen verschiedenen Anwendungen oder Systemen fungiert, um deren Interoperabilität zu gewährleisten.
Mindestanforderungen	Grundlegende Anforderungen, die erfüllt sein müssen, damit eine Alternative oder ein Angebot in die engere Auswahl kommt.
Mockup	Visuelle Darstellung eines Designs oder einer Benutzeroberfläche, die als Vorlage für die Entwicklung dient und das Layout und die Funktionalität zeigt.
Nutzwert-Analyse	Methode zur Bewertung und Auswahl von Alternativen anhand einer gewichteten Bewertung verschiedener Kriterien.
Offline-Fähigkeiten	Fähigkeit einer Anwendung, auch ohne Internetverbindung zu funktionieren und Daten lokal zu speichern und zu synchronisieren, sobald die Verbindung wiederhergestellt ist.
ÖGD (Öffentlicher Gesundheitsdienst)	Ein Bereich der öffentlichen Verwaltung, der sich mit der Gesundheitsförderung und -versorgung der Bevölkerung beschäftigt.
Outsourcing-Kosten	Kosten, die durch die Auslagerung von Geschäftsprozessen oder Dienstleistungen an externe Dienstleister entstehen.
OZG (Onlinezugangsgesetz)	Gesetz zur Verbesserung des Onlinezugangs zu Verwaltungsleistungen ( <a href="https://www.gesetze-im-internet.de/ozg/">https://www.gesetze-im-internet.de/ozg/</a> )
OZG-Leistung	Verwaltungsleistungen, die gemäß dem Onlinezugangsgesetz (OZG) digital angeboten werden müssen.
Pilotimplementierung	Kontrollierte Einführung eines neuen Prozesses oder Systems im kleinen Rahmen, um dessen Wirksamkeit und Funktionalität zu testen und zu evaluieren, bevor es vollständig integriert wird.
Pseudonymisierung	Verfahren zur Verarbeitung personenbezogener Daten, bei dem die Identität der betroffenen Person durch ein Pseudonym ersetzt wird, um den Datenschutz zu erhöhen.
Reaktive Designoptionen	Designansatz, bei dem Benutzeroberflächen dynamisch auf Änderungen in den Daten reagieren und sich automatisch aktualisieren.
Report-Builder	Ein Werkzeug zur Erstellung von Berichten, das es ermöglicht, Datenquellen zu verbinden, Abfragen zu erstellen und Berichte zu gestalten.
Review	Eine Überprüfung oder Bewertung von Software, Prozessen oder Dokumenten, um Qualität und Konformität sicherzustellen.
Robotic Process Automation (RPA)	Technologie zur Automatisierung wiederkehrender, standardisierter Aufgaben durch Software-Roboter.

Schutzbedarf	<i>Der Schutzbedarf beschreibt die Notwendigkeit, Informationen und Systeme vor Bedrohungen zu schützen, basierend auf der möglichen Schadenshöhe bei Verlust oder Kompromittierung der Daten.</i>
Schutzbedarfskategorie	Klassifizierung von Daten oder Systemen nach ihrem Schutzbedarf, um geeignete Sicherheitsmaßnahmen zu bestimmen (Kategorien: normaler, hoher und sehr hoher Schutzbedarf).
Scrum	Ein agiles Rahmenwerk zur Softwareentwicklung, das iterative und inkrementelle Prozesse betont.
Server-Performance	Messung der Leistungsfähigkeit eines Servers anhand verschiedener Betriebsprozesse, wie z. B. Datenübertragungsraten, Auslastung der Threads und Speicherplatznutzung.
Service	Eine Dienstleistung oder Funktion, die von einer Software oder einem System bereitgestellt wird.
Skalierbarkeit	Die Fähigkeit eines Systems, seine Leistung und Kapazität bei steigender Last oder zunehmenden Anforderungen zu erhöhen.
Success Stories	Erfolgsberichte über Personen oder Projekte.
Support	Unterstützung und Hilfe, die Nutzern bei der Verwendung von Software oder Systemen angeboten wird.
Tutorial	Anleitung oder Lehrmaterial, das Schritt für Schritt erklärt, wie eine bestimmte Aufgabe oder Funktion ausgeführt wird.
UI-Elemente	Benutzeroberflächenelemente, die zur Interaktivität einer Anwendung beitragen, wie z. B. Schaltflächen, Menüs und Eingabefelder.
Use Case	Ein spezifisches Szenario, in dem ein System eine externe Anfrage erhält und darauf reagiert, um ein Ziel zu erreichen.
Webanwendungen	Softwareanwendungen, die über einen Webbrowser zugänglich sind und auf Webservern gehostet werden.
Wildwuchs	Unkontrollierte und unstrukturierte Entwicklung von Software, die zu Inkonsistenzen und Wartungsproblemen führen kann.
Workflow	Abfolge von Schritten oder Prozessen, die zur Erledigung einer Aufgabe oder zur Fertigstellung eines Projekts führen.
XÖV	XML in der öffentlichen Verwaltung ist ein deutscher Standardisierungsrahmen und eine Sammlung von Standards für den elektronischen Datenaustausch der öffentlichen Verwaltung auf der Basis von Nachrichten in XML-Syntax und zugehörigen Codelisten und Prozessen.

### Auftraggeberin

Vorsitzland UAG Digitalisierung der AG ÖGD der AOLG  
Senatsverwaltung für Wissenschaft, Gesundheit und Pflege  
Oranienstraße 106  
10969 Berlin

E-Mail: [gst-uagdigitalisierung@paktoegd.de](mailto:gst-uagdigitalisierung@paktoegd.de)

### Ansprechpartnerin UAG

**Frau Dr. Sabine Hermann**  
Vorsitzende UAG Digitalisierung der AG ÖGD der AOLG

**Erstellt durch die Geschäftsstelle der UAG Digitalisierung**

### Dienstleister (im Auftrag)

Tim Kaukerit  
Arman Latifi  
Annika Uhlig  
Dr. Susanne Dobratz



**HiSOLUTIONS**

**HiSolutions AG**  
Schloßstraße 1  
12163 Berlin

E-Mail: [info@hisolutions.com](mailto:info@hisolutions.com)

Stand: 02.02.2026