



DB Systel
Digital bewegen.
Gemeinsam.



Digital.Trend.Studie **Do It Yourself Computing**

Die Demokratisierung der IT

DB Systel GmbH | Digital Foresight | Dezember 2021

Einführung

Das Kund:innenerlebnis durch Digitalisierung verbessern	4
Herzlich Willkommen, zu unserer Digital.Trend.Studie „DIY Computing – Die Demokratisierung der IT“!	5

Zusammenfassung

Was verstehen wir unter Do It Yourself (DIY) Computing?	8
Trendbewertung DIY Computing	9
Business Value: Medium	9
Maturity (Reifegrad): Standardizing	10
Kernergebnisse und Empfehlungen	11
Chancen und Risiken	13

Technologische Grundlagen

Agile Softwareentwicklung	15
No-Code- und Low-Code-Entwicklung	17
Intelligente Prozessautomation	18
Selbstkonfigurierende und selbstintegrierende Plattformen	19
Künstliche Intelligenz in der Softwareentwicklung	20

Die Demokratisierung der IT

Der „Citizen Developer“	23
Gesetzliche Rahmenbedingungen und die Rolle der Arbeitnehmervertreter:innen	25
Anwendungsfälle und Herausforderungen für Citizen Development	26

Partnerschaftliche Zusammenarbeit zwischen IT- und Fachseite

Die klassische IT-Organisation	28
Herausforderungen für den Digitalisierungspartner der Zukunft	29
Herausforderungen für die Digitalanwender:innen der Zukunft	30
Neue Formen der Zusammenarbeit und Geschäftsmodelle	32

DIY Computing im Jahr 2031

DIY Computing im mobilen Einsatz	35
DIY Computing im (räumlich nicht festgelegten) Büro	36

Firmenportraits und Impulse

Anbieter von Citizen-Development-Werkzeugen	38
Große ERP Anbieter	40
Trendimpulse	41

Laufende Aktivitäten

Projekte und Maßnahmen bei der Deutschen Bahn	43
---	----

Anhang

Glossar und Abkürzungen	46
Vorgehensweise & Methodik	47
Trendbewertung	48
Ansprechpartner:innen	49



Einführung

Das Kund:innenerlebnis durch Digitalisierung verbessern 4
Herzlich Willkommen, zu unserer Digital.Trend.Studie „DIY Computing – Die Demokratisierung der IT“!.... 5

SHOW BUSINESS

12010111010101001
00101010101000101
0010101010010101

Das Kund:innenerlebnis durch Digitalisierung verbessern

Unsere Kund:innen, ob Reisende oder Geschäftskund:innen in der Logistik, sollen sich auf schnelle und pünktliche Züge verlassen können und einen erstklassigen Service erleben. Das ist unser Anspruch, dem wir uns verpflichtet fühlen. Ohne eine konsequente Digitalisierung werden wir diese in uns gesetzten Erwartungen nicht erfüllen können. Dabei geht es nicht nur um Lösungen, die Kund:innen direkt wahrnehmen, sondern um eine durchgängige Digitalisierung der Wertschöpfungsprozesse der Deutschen Bahn von der Planung bis hin zur Nachbereitung einer Reise oder eines Gütertransports. Betroffen sind dabei selbstverständlich alle Bereiche, von der Schiene über den Bahnhof bis in unsere Werkstätten.

Um diesen Herausforderungen erfolgreich zu begegnen, muss die zugrundeliegende Informationstechnologie nicht nur modern und zuverlässig sein, sie muss auch zum integralen Bestandteil unseres Geschäfts werden. Diese Digitale Transformation, die alle Bereiche durchdringt, erfordert eine enge Zusammenarbeit von IT und Business, eine gelebte „Business IT Fusion“. Auch wenn die „Übersetzung“ des Geschäfts in Technologie durch das Zusammenrücken von Business und IT schneller und besser gelingt, so wird die Entwicklung, das Programmieren von Anwendungen und Apps, eine wesentliche Basis bleiben. Programme erfordern Programmierer:innen - und diese sind leider nicht unbegrenzt verfügbar.

Programmierung darf deshalb nicht den IT-Spezialist:innen vorbehalten bleiben. Technologie muss vielmehr „demokratisiert“ werden. Bereits heute verfügen wir über Werkzeuge, die es dem technikinteressierten Fachexpert:innen ermöglicht, eine auf seine konkrete Business-Anforderung zugeschnittene App selbst zu entwickeln. Mag dies aktuell noch für einfachere Aufgaben möglich sein, so wagt das Team Digital Foresight der DB Systel in dieser Trendstudie auch wieder den Blick in die Kristallkugel: Künstliche Intelligenz wird die Erstellung von Anwendungen und Apps weiter vereinfachen, so dass auch komplexere Aufgabenstellungen durch den Fachexpert:innen ohne Einbindung von IT-Spezialist:innen umgesetzt werden können.

Die Demokratisierung der IT, das „Citizen Development“, erfordert aber nicht nur intelligente Werkzeuge. Vielmehr müssen auch die Hemmschwellen der Nicht-Technolog:innen gesenkt werden, Anwendungen selbst zu entwickeln. Deshalb freut es mich umso mehr, dass in der vorliegenden Trendstudie auch den organisatorischen und kulturellen Aspekten ausreichend Raum gegeben wird. Nur wenn wir alle bei der Digitalisierung unseres Geschäfts mit anpacken und wir den „Do It Yourself“ Gedanken auch in der IT leben, kann die Digitale Transformation im Dienst unserer Kund:innen erfolgreich sein.

Denn wenn alle an einem Strang ziehen, gelingt uns der Weg hin zu einer Wertorientierten IT, die zu einem einzigartig guten Kund:innenerlebnis beiträgt!



Dr. Daniela Gerd tom Markotten
Vorständin Digitalisierung und Technik | Deutsche Bahn AG



Herzlich Willkommen, zu unserer Digital.Trend.Studie „DIY Computing – Die Demokratisierung der IT“!

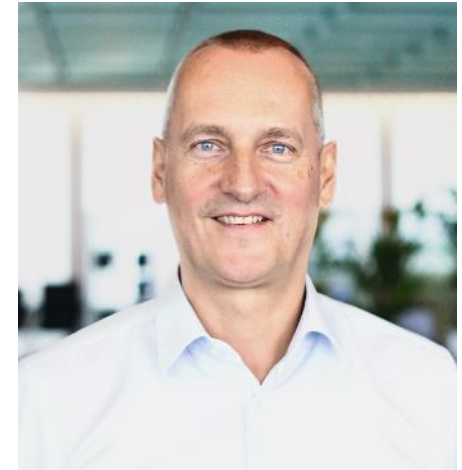
In den vergangenen Monaten wurde das Digitalisierungsdefizit in Deutschland wiederholt thematisiert. In vielen Bereichen, auch bei der Deutschen Bahn, muss Informationstechnologie modernisiert oder, falls noch nicht vorhanden, sogar neu eingeführt werden. Ein Vorhaben, das immense Ressourcen erfordert und nicht nur uns vor große Herausforderungen stellt. Herausforderungen, die noch wachsen, wenn man den zunehmenden Fachkräftemangel im IT-Umfeld einbezieht. Bei diesen Rahmenbedingungen wird schnell klar, dass Digitalisierung mit den bestehenden arbeitsteiligen Strukturen, d.h. anfordernden Fachexpert:innen und spezialisierten IT-Entwickler:innen sehr wahrscheinlich nicht gelingen kann.

Gleichzeitig beobachten wir auf der IT-Produktionsseite Innovation und Fortschritt: sich wiederholende Arbeitsprozesse werden mit Hilfe von Softwarerobotern durch eine sogenannte „[↗ Robotic-Process-Automation](#)“ automatisiert. „[↗ Low-Code-Plattformen](#)“ ermöglichen die Erstellung von Software mittels intuitiver, grafischer Werkzeuge, die durch Künstliche Intelligenz immer besser werden. Fachanwender:innen arbeiten eng mit IT-Spezialist:innen in agilen Teams zusammen. Die Programmierung von Anwendungen und Applikationen wird zukünftig nicht mehr überwiegend in den Händen von Spezialist:innen liegen, sondern sich auf mehrere Schultern verteilen. Mitarbeitende ohne spezielle IT- oder Programmierkenntnisse werden in die Lage versetzt, als so genannte „[↗ Citizen Developer](#)“ Anwendungen für ihr spezifisches Problem selbst zu entwickeln. Einen Trend, den wir als „**Do-It-Yourself (DIY) Computing**“ bezeichnen.

Dieses geänderte Rollenverständnis stellt uns vor herausfordernde Fragestellungen, die wir in der vorliegenden Trendstudie differenziert betrachten wollen: was bedeutet es für einen qualitativ hochwertigen, skalierungsfähigen und sicheren IT-Betrieb, wenn Fachanwender:innen zukünftig eigenständig programmieren? Welche Werkzeuge und Fähigkeiten muss ein Digitalisierungspartner dem Citizen Developer zur Verfügung stellen, damit eben dies gewährleistet werden kann? Welche rechtlichen und mitbestimmungspflichtigen Sachverhalte sind damit verbunden?

In Hintergrundanalysen werden wir Technologien vorstellen, die „Do-It-Yourself“ Computing ermöglichen. Der neuen Rollenteilung und den rechtlichen Rahmenbedingungen wird ausreichend Raum gegeben. Einige Anwendungsbeispiele sind in dieser Trendstudie enthalten. Auch spendieren wir wieder ein paar Seiten „Science Fiction“: wir begleiten zwei Anwender:innen in ihrem Alltag im Jahr 2031 und schauen ihnen über die Schulter, während sie ihre tägliche Arbeit mittels selbsterstellten, kleinen Anwendungen erledigen. Doch wie bei allen Trends ist auch das DIY-Computing bereits in der Gegenwart angekommen.

Wir präsentieren einige interessante Beispiele aus dem Deutsche Bahn Universum, denn **wir sind bereits auf dem Weg in die DIY-Computing Welt, einer komplexen Welt, die wir Euch so verständlich wie möglich erklären wollen. Den großen Herausforderungen der Digitalisierung können wir nur gemeinsam begegnen: IT-Spezialist:innen, die innovative Technologien und Plattformen konsequent an geschäftlichen Mehrwerten ausrichten, arbeiten Hand-in-Hand mit Citizen Developern, die verantwortungsbewusst Applikationen für ihre speziellen Anforderungen selbst entwickeln.**



Wir möchten uns an dieser Stelle ganz besonders bei allen Mitwirkenden bedanken, die uns bei der Erstellung der Studie unterstützt haben. Nicht zu vergessen die zahlreichen Kolleg:innen, die bereits heute in neuen Rollen und innovativen Strukturen mit modernen Werkzeugen bei der Digitalisierung der Deutschen Bahn anpacken. Da es sich um einen sehr facettenreichen Trend mit vielen Perspektiven handelt, freuen wir uns auf Euren Feedback und den Beginn einer gemeinsamen, fruchtbaren Diskussion.

Doch vorher wünschen wir Euch eine spannende und inspirierende Lektüre.



Christian Kolarsch
Team Digital Foresight | DB Systel GmbH

Hinweise in eigener Sache:

Zu den Links auf Quellen und Artikel:

Im Rahmen unserer Trendanalysen greifen wir unter anderem auf die Expertise der Analysten von [Gartner](#) oder [Forrester](#) zurück. Einige Passagen in dieser Digital.Trend.Studie verweisen auf unabhängige Studien dieser führenden Expert:innen. Aus Gründen der Vollständigkeit und Nachvollziehbarkeit haben wir an den entsprechenden Textstellen auf die betreffenden Dokumente verlinkt. Die zugrundeliegenden Quellen sind allerdings nur mit einem aktiven Kundenkonto bei dem entsprechenden Analystenhaus abrufbar. Solltest Du an weiteren Informationen interessiert sein, sprich uns gerne an. Allerdings auch hier bereits der Hinweis: die Dokumente dürfen wir aus rechtlichen und vertraglichen Gründen nicht weitergeben!

Forrester®, Technographics®, Forrester Wave, RoleView, TechRadar, and Total Economic Impact are trademarks of Forrester Research, Inc. All other trademarks are the property of their respective companies. For additional information, go to www.forrester.com.

GARTNER® and HYPE CYCLE are registered trademarks and service marks of Gartner, Inc. and/or its affiliates in the U.S. and internationally and are used herein with permission. All rights reserved.

Fotos: Seite 1/7/14/22/27/34/37/42/45 Sergey Nivens – Adobe Stock | Seite 18 sompong.tom – Adobe Stock | Seite 20 sdecoret – Adobe Stock | Seite 29 metamorworks – Adobe Stock | Seite 30 Max Lautenschläger – DB AG
Seite 35 Edler von Rabenstein – Adobe Stock | Seite 36 WavebreakmediaMicro – Adobe Stock | Seite 41 (Oben) Supphachai – Adobe Stock | Seite 41 (Mitte) panumwat – Adobe Stock | Seite 41 (Unten) Looker_Studio – Adobe Stock



Zusammenfassung

Was verstehen wir unter Do It Yourself (DIY) Computing?	8
Trendbewertung DIY Computing	9
Business Value: Medium	9
Maturity (Reifegrad): Standardizing	10
Kernergebnisse und Empfehlungen.....	11
Chancen und Risiken.....	13

Was verstehen wir unter Do It Yourself (DIY) Computing?

Definition

In der Vergangenheit erfolgte die Entwicklung von neuen IT-Systemen in der Regel arbeitsteilig: Anwender:innen beschrieben ihre Anforderungen an eine neue Applikation. Diese wurde von spezialisierten Entwickler:innenteams umgesetzt und nach einer Abnahme bereitgestellt. Mit zunehmender Digitalisierung wird sich ein grundsätzlich anderes Rollenverständnis ausprägen: fachliche Expert:innen werden befähigt, auf Ihre Anforderungen passende Lösungen selbst zu erstellen. Erst durch die Bereitstellung sowie einen sicheren und zuverlässigen Betrieb von leistungsfähigen IT-Plattformen und Services, wird dies möglich.

Themen wie [↗ No-Code-Entwicklung](#), [Citizen Developer](#) oder [↗ Intelligente Prozessautomatisierung](#) weisen bereits heute den Weg dahin. In unserem Trend DIY Computing betrachten wir sowohl technische Grundlagen als auch mögliche Auswirkungen auf die IT-Organisation und dadurch neu entstehende Geschäftsbeziehungen und -modelle.

Hintergrund

Der Digitalisierungsdruck nimmt in allen Bereichen zu. Eine konsequente Digitalisierung erfordert die „Übersetzung“ realer Sachverhalte und Prozesse in eine virtuelle, digitale Welt. In der Vergangenheit wurden Spezialist:innen benötigt, um fachliche Anforderungen in Ablauf- und Datenstrukturen zu übersetzen und daraus den dafür nötigen Programmcode in Programmiersprachen wie [↗ Python](#) oder [↗ Java](#) zu entwickeln. Der wachsenden Nachfrage steht ein nur sehr [↗ eingeschränktes Angebot an ausgebildeten IT-Spezialist:innen](#) gegenüber. Wie in der aktuellen [↗ Technologiestrategie der DB System](#) aufgezeigt, werden Low-Code- und No-Code-Umgebungen der Fachseite bei der Programmierung Möglichkeiten eröffnen, Softwarelösungen ohne Hilfe von IT-Professionals zu erstellen.

Anders als der Name No- bzw. Low-Code suggeriert, wird mit diesen Werkzeugen immer noch programmiert, wenn auch auf Basis wesentlich intuitiverer und einfacherer grafischer Benutzerschnittstellen. Das Schreiben von kryptischen Befehlen in einer Programmiersprache kann mit diesen Entwicklungsumgebungen in weiten Teilen vermieden werden.

Wenn wir weiter in die Zukunft blicken, wird Künstliche Intelligenz bei der digitalen Lösungsfindung und -entwicklung eine immer größere Rolle spielen und neben professionellen Entwickler:innen immer mehr Fachfremden die Tür zur Entwicklung von Applikationen öffnen. Neben der Programmierung könnte zukünftig eine KI auch die Übersetzung von Problemen in eine strukturierte Logik unterstützen.

Doch nicht nur die Programmierung von Applikationen werden Fachexpert:innen zukünftig bei ihren digitalen Herausforderungen unterstützen. Auch die Automatisierung ganzer Geschäftsprozesse mittels intelligenter Prozessautomatisierung und [↗ Hyperautomation](#) wird dabei helfen. Robotic Process Automation (RPA) stellt eine Vorstufe zu diesen Themen dar. Wiederholbare Arbeitsschritte werden bereits heute mittels RPA automatisch abgearbeitet und schaffen damit mehr kreative Freiräume bei Fachspezialist:innen. Zukünftig werden Fachexpert:innen, gestützt durch Künstliche Intelligenz, Geschäftsprozesse selbstständig digitalisieren und automatisieren.

Trendbewertung DIY Computing

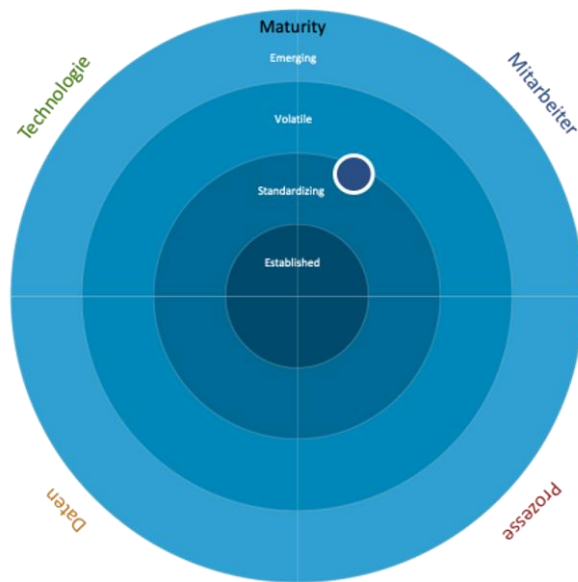


Abbildung 1: Einordnung des Digitaltrends „DIY Computing“ im Digital.Trend.Radar

Im Folgenden geben wir Ihnen unsere Einschätzung hinsichtlich der Relevanz, der Maturity (Reifegrad) und des Business Value (Mehrwert für die DB) zur Hand. Die Trendbewertungen basieren auf systematischen Ableitungen anhand eines objektiven Kriterienkataloges. Im Anhang finden Sie weitere Erläuterungen zu unserer Methodik und Vorgehensweise. Die Abbildung zeigt die Einordnung des Trends „DIY Computing“ auf Grundlage der oben beschriebenen Definition.

Business Value: Medium

Der konsequente Einsatz von DIY Computing Werkzeugen bei Anforderungen mit adäquatem Komplexitätsgrad wirkt sich positiv auf Markteinführungsgeschwindigkeit und Entwicklungskapazität aus. Bei passenden Rahmenbedingungen könnte DIY Computing die IT-Entwicklung in gleichem Maße verändern, wie es die zunehmende Verbreitung von Cloud-Technologien für den IT-Betrieb getan hat.

Unternehmen, denen es gelingt, eine neue digitale Problemlösungskultur zu etablieren, verfügen bereits heute über einen signifikanten Wettbewerbsvorteil. Dieser entsteht, weil Fehler im Transfer zwischen realer und digitaler Welt schneller erkannt und behoben bzw. sogar minimiert werden können.

Außerdem wird die IT-Entwicklungskapazität auf mehrere Schultern verteilt. Fachliche Expert:innen sind als Citizen Developer in der Lage, ihre Ideen und Anforderungen unmittelbar in eigenerstellten Applikationen zu realisieren. Ein Faktor, der sich positiv auf die Umsetzungsgeschwindigkeit neuer Ideen und Geschäftsmodelle und damit die Nutzereffizienz auswirkt.

Allerdings birgt dies auch ein erhöhtes Disruptionspotential, sollten bestehende oder potenzielle Wettbewerber bei der Umsetzung von DIY Computing schneller sein, da sie u.a. Zeitvorteile bei der Einführung neuer Funktionalitäten und Produkte realisieren können.

Die heutige Gesetzeslage, insbesondere die Mitbestimmung bei der Einführung neuer IT-Lösungen, stellt zum jetzigen Zeitpunkt ein regulatorisches Hindernis dar. Wir erwarten jedoch, dass sich dieses aufgrund eines immer breiteren Einsatzes von DIY Computing mittelfristig löst. Direkte größere Effekte bei Umsatzpotential und Kapitaleffizienz ergeben sich nicht. Wenn überhaupt, sind diese nur mittelbar zu erwarten und werden diesem Trend nicht direkt zugerechnet.

Business Value ⓘ	Medium
Umsatzpotential	●●○○○
Nutzereffizienz	●●●●●
Kapitaleffizienz	●○○○○
Regulatorisches	●●●●○
Disruptionspotential	●●●●○

Abbildung 2: Bewertungskriterien des DB Business Value

Maturity (Reifegrad): Standardizing

Viele Technologien und Tools, die den Citizen Developer befähigen, IT zu entwickeln, sind bereits heute vorhanden und können genutzt werden. Welche Aufgaben von Citizen Developern eigenständig entwickelt werden können, hängt von der Komplexität der umzusetzenden Anforderung ab. Diese Komplexitätsgrenze wird sich aufgrund zunehmender Integration und Verbesserung der KI in bestehenden Tools verschieben.

Mit No-Code- und Low-Code-Plattformen sind ebenso wie mit Robotic Process Automation bereits heute Werkzeuge verfügbar und im Einsatz. Eine Ausnahme bilden innovative Themen wie z.B. [selbstintegrierende Applikationen](#). Auch für diese werden nach Prognosen von IT-Analysten zumindest mittelfristig erste marktreife Lösungen erwartet.

DIY Computing wird nicht nur durch Entwicklungswerkzeuge und -plattformen zum Erfolg, sondern erfordert ein verändertes Rollenverständnis und beim Citizen Developer ein Verstehen einfacher technologischer Grundlagen. Entsprechende Trainings- und Beratungsangebote werden damit zum kritischen Erfolgsfaktor. Sollte diese „Befähigung“ nicht erfolgreich sein, so werden innovative Werkzeuge weiterhin weitestgehend nur von IT-Profis eingesetzt und der prognostizierte Business Value wird nicht realisiert.

Am Markt verfügbare Werkzeuge bewegen sich in der proprietären Welt ihrer Hersteller. Eine übergreifende Standardisierung und damit einhergehende Interoperabilität existiert nicht. Auch andere Unternehmen stehen vor diesen die Technik und Qualifikation betreffenden Herausforderungen und sind nicht weiter in der Umsetzung. Die mangelnde Interoperabilität stellt eine zu lösende Aufgabe für den Digitalisierungspartner der Zukunft dar: bereits vorhandene Lösungen für ein fachliches Problem auch anderen relevanten Bereichen in einem Großunternehmen verfügbar zu machen und diese in einem [API-Ökosystem](#) zu verbinden.



Abbildung 3: Bewertungskriterien der Maturity (Reifegrad)

Kernergebnisse und Empfehlungen



Es gibt nie nur den einen richtigen Weg. Es lohnt sich aber darüber nachzudenken, ob man sich treiben lassen will oder proaktiv die Initiative ergreift. Dafür möchten wir Ihnen an dieser Stelle einige Empfehlungen geben.

Kernergebnis 1: Zusammenarbeit zwischen Digitalisierungspartnern und Digitalanwender:innen in einer wertorientierten IT

In der klassischen IT-Welt waren die Rollen zwischen dem fachlichen Auftraggeber und dem IT-Dienstleister klar verteilt. Bei einem DIY Computing in der Fachabteilung wird das nicht mehr funktionieren. Neue partnerschaftliche Zusammenarbeits- und Geschäftsmodelle hin zu einer wertorientierten IT müssen entwickelt werden. In diesem Zusammenhang sind auch Verantwortungen und Besitzfragen einvernehmlich zu klären, wie z.B.: wer stellt die DIY-Plattformen? Wie werden moderne und zukunftsfeste Werkzeuge finanziert? Wem gehört die darauf entwickelte Applikation? ...und viele Fragen mehr.

Empfehlung:

Bereits heute sind Low-Code-Werkzeuge bei der Deutschen Bahn schwerpunktmäßig bei professionellen Entwickler:innen im Einsatz. Mittels einfacher Anwendungsfälle sollte die Nutzung auch durch Nicht-Technologen forciert werden. Digitalisierungspartner und Digitalanwender:innen verständigen sich gemeinsam auf sinnvolle Pilotprojekte mit weniger komplexen fachlichen Herausforderungen wie z.B. der Bearbeitung einfacher Prozesse oder Formular- und Tabellenverarbeitung. Dies bietet auch die Möglichkeit Akzeptanzhürden zu überwinden. Parallel dazu müssen skalierbare Organisationsmodelle erarbeitet werden; kein einfaches Unterfangen bei einer komplexen mehrstufigen IT-Landschaft in einem Großkonzern. Umso wichtiger ist es, früh und partnerschaftlich die Diskussion zu beginnen. Modelle zur Zusammenarbeit, wie im gleichnamigen Kapitel auf Seite 32 vorgestellt, könnten dafür als Blaupause dienen.

Kernergebnis 2: DIY Computing ist mehr kulturelle als technische Herausforderung

Technische Werkzeuge, die DIY Computing ermöglichen, sind heute bereits vorhanden und werden immer intuitiver und einfacher in der Anwendung. Die größere Herausforderung besteht darin, Nicht-Technologen für Technologie zu begeistern. Trainings werden nicht ausreichen. Auch werden „von oben angeordnete“ Strategien und Programme nicht zu einer Demokratisierung der Technologie führen. Vielmehr gilt es Hemmschwellen abzubauen und die Menschen dafür zu begeistern, aktiver Teil der digitalen Transformation zu werden.

Empfehlung:

Mitarbeitende sollten davon überzeugt sein, dass ihnen Technologie in der täglichen Arbeit hilft. Um die Hemmschwelle niedrig zu halten, müssen Werkzeuge einfach verfügbar sein. Auch helfen Kolleg:innen, die im Nachbarbüro sitzen und bereits mit Software-Unterstützung ihren Arbeitsalltag verbessert haben. Neben der Kommunikation, dass uns DIY Computing in der Digitalisierung Vorteile verschafft, und den notwendigen Schulungsmaßnahmen sollten Mentorenprogramme, Coaching und Communities etabliert werden in denen technologisch Interessierte helfen, den weniger Versierten über erste Hürden zu kommen. Mentorenprogramme und Communities bieten sich außerdem an, um verbindliche Leitplanken (siehe auch Kernergebnis 4) und die Gründe dafür einfach zu vermitteln. Damit können weitere Berührungspunkte minimiert werden. Dennoch ganz wichtig: nicht jede Person ist technologieaffin und möchte zum Citizen Developer werden. Auch dies gilt es zu respektieren!

Kernergebnis 3: Bestehende Entwicklungsrollen und -prozesse müssen angepasst bzw. erweitert werden

In unserer arbeitsteiligen IT-Welt sind viele Sachverhalte, die durch Gesetze, Stichwort Datenschutzgrundverordnung (DSGVO) oder Betriebsverfassungsgesetz (BetrVG) vorgegeben werden, in Richtlinien und Prozessen klar geregelt. Die Rollen und die damit einhergehenden Verantwortungen sind verbindlich dokumentiert. Bei Nichteinhaltung unserer Richtlinien kann dies mitunter strafrechtlichen Konsequenzen haben. Im DIY Computing greifen unsere Regelwerke teilweise nicht mehr und auch die Gremienbeteiligung wird aufgrund zunehmender Dynamik erschwert.

Empfehlung:

Es gilt, gemeinsam mit Interessensvertreter:innen wie Datenschutz- und Sicherheitsbeauftragten sowie Betriebsräten zu prüfen, an welchen Stellen unsere bestehenden Regelwerke und Richtlinien wie bspw. die [7 KoRiL 114](#) für ein DIY Computing nicht ausreichen. Bereits frühzeitig müssen die verantwortlichen Datenschutz- und Sicherheitsexpert:innen sowie Gremien einbezogen werden, um bestehende Regelwerke und Prozesse auch für fachseitig entwickelte Applikationen anwendbar zu machen.

Kernergebnis 4: DIY Computing erfordert verbindliche Leitplanken und Regeln

Personen, die Informationssysteme und Applikationen nur von der Anwender:innenseite kennen, müssen sich um Dinge, die „unter der Motorhaube“ passieren, in der Regel nur rudimentär kümmern. Für nichtfunktionale Sachverhalte, wie bspw. Verfügbarkeit, Skalierbarkeit oder Dokumentation, bedarf es verbindlicher und einfach verständlicher Regeln. Auch sollte ein einheitliches Verständnis darüber bestehen, wann DIY Computing ermöglicht und unterstützt wird, und für welche Aufgabenstellungen IT-Expert:innen nach wie vor gefragt sind.

Empfehlung:

Um das Bild der Motorhaube noch einmal zu bemühen: Autofahrer mussten früher wissen, wann der Ölfüllstand oder Reifendruck kritisch wird, um rechtzeitig eingreifen zu können. Gleiches gilt für den Citizen Developer: er benötigt verbindliche „Grenzwerte“, um zu wissen, wann bspw. „seine“ Applikation hohen Leistungs- und Skalierungsanforderungen genügen muss. Es gilt genau diese Leitplanken in einer für Fachfremde verständlichen Sprache zu entwickeln. Doch allein das genügt nicht. Auch wenn wir an das Gute im Menschen glauben, bedarf es unternehmensinterner Instanzen, die die Einhaltung dieser Regeln prüfen. Wo diese Instanzen anzusiedeln sind und mit welchen Kompetenzen sie ausgestattet werden, ist wiederum eine Frage der Organisation (vgl. Kernergebnis 1).

Kernergebnis 5: Einfach nutzbare, intuitive Plattformen und Werkzeuge bilden die Grundlage

Neben der Überwindung kultureller Hürden sind intuitive technische Werkzeuge Grundvoraussetzung für ein erfolgreiches DIY Computing. Technische Plattformen und Werkzeuge müssen nicht nur vorhanden sein, vielmehr ist das einfache Finden von und die Zugreifbarkeit auf die passenden Werkzeuge durch den Citizen Developer ein kritischer Erfolgsfaktor.

Empfehlung:

Mit der [7 Microsoft Power Plattform](#), [7 Pega](#) und [7 UIPath](#) haben wir bereits ausgewählte Entwicklungswerkzeuge verfügbar. Im Rahmen der in Kernergebnis 1 vorgeschlagenen Pilotprojekte können diese hinsichtlich Anwendbarkeit durch Fachexpert:innen genutzt und geprüft werden. Nach einem derartigen „Friendly-User-Test“ und Ausarbeitung der organisatorischen Grundlagen sowie Prozesse und Regelwerke, sollten diese Werkzeuge auch fachseitig weiter verfügbar gemacht werden. Neben Kommunikation werden adäquate Schulungs- und Mentorenmaßnahmen erforderlich.

Chancen und Risiken

Yin oder Yang? Trends und damit einhergehende Entwicklungen sind objektiv betrachtet neutrale Veränderungen. In jedem Trend stecken sowohl Chancen als auch Risiken. Welche Seite der Medaille überwiegt, entscheidet oft das eigene Handeln.

Chancen [wenn wir richtig handeln]:

- Wenn fachliche Kolleg:innen und Expert:innen bis zu einem gewissen Komplexitätsgrad in ihrer Wissensdomäne Anforderungen selbst umsetzen, können nicht wertschöpfende Tätigkeiten, wie z.B. das Schreiben von Anforderungen, eingespart werden.
- Früher haben Anwender:innen Anforderungen beschrieben, die dann über mehrere Schritte in ausführbare Programme „übersetzt“ wurden. Wenn diese ihre Vorstellungen und Anforderungen bis zu einem gewissen Grad nun direkt umsetzen, fallen Fehlerquellen aus einer Übersetzung weg. Neben der Fehlerreduktion werden eigenentwickelte Apps auch im fachlichen Umfeld des Citizen Developers besser akzeptiert.
- Repetitive Tätigkeiten können durch intelligente Prozessautomatisierung reduziert werden. Die Produktivität der fachlichen Mitarbeitenden verbessert sich dadurch. Freiräume für komplexere, wertschöpfende Tätigkeiten werden geschaffen.
- Auf einer unternehmensweiten Plattform werden Services und Applikationen zur Verfügung gestellt. Über die dadurch entstehenden Möglichkeiten einer Wiederverwendung dieser Komponenten wird nicht nur die Verwendung der IT, sondern auch Fachwissen demokratisiert und vergemeinschaftet.
- No-Code- und Low-Code-Plattformen bieten sich für weniger komplexe Aufgabenstellungen an, die bisher am Arbeitsplatz über Tabellenkalkulationen oder Textverarbeitung zur Formularverarbeitung gelöst wurden. Richtig eingesetzt können diese Anwendungen professionalisiert, d.h. robuster und sicherer werden.
- Der Mangel an IT-Fachkräften wirkt sich nicht nur auf die Realisierungszeiten von Applikationen negativ aus. Mit zunehmender Digitalisierungsgeschwindigkeit wird sich dieser Druck noch erhöhen. Durch DIY Computing können Entwicklungsleistungen auf mehr Schultern verteilt werden.

Risiken [wenn wir nicht die richtigen Dinge tun]:

- DIY Computing und die damit einhergehende Demokratisierung von Technologie erfordert verbindliche „Rahmenbedingungen“. Ansonsten drohen die Gefahren einer [↗ nicht professionell entwickelten und betriebenen IT](#) mit entsprechenden Qualitäts- und Sicherheitsrisiken. Diese können bei geschäftskritischen und sicherheitsrelevanten Abläufen massive Folgen haben.
- Auch bei kleineren Applikationen spielen Themen wie Compliance, Datenschutz oder Revisionssicherheit eine wesentliche Rolle. Verbindliche Regeln sind gefordert, um nicht aufgrund von Gesetzes- oder Regelverstößen belangt werden zu können.
- Dem Citizen Developer sind mitunter die Bedeutung nicht-fachlicher Aspekte wie Verfügbarkeit, Lastverhalten oder Skalierbarkeit nicht bewusst. Wenn eine kleine eigenentwickelte App eine kritische Bedeutung für einen Geschäftsprozess hat, kann dies zu Fehlern bis hin zu Ausfällen führen. Weitere Aufwände für Wartung und Dokumentation werden vom Fachunkundigen häufig unterschätzt.
- Wenn in einem Großunternehmen die Applikationsentwicklung ohne einen übergreifenden Ansatz demokratisiert wird, könnte die Komplexität unbeherrschbar werden: für gleiche und ähnliche Anforderungen werden redundante, teilweise sogar sich widersprechende Lösungswege gewählt. Eine professionelle, störungsfreie Betriebsführung wird unmöglich.
- In vorhandenen Richtlinien zur Informationssicherheit sind Rollenverantwortungen, z.B. die Zuständigkeit für den Geschäftsprozess oder der Applikation klar geregelt. Sollte es zu einem Sicherheitsvorfall mit einer selbsterstellten Lösung kommen, ist dem Citizen Developer mitunter nicht klar, welche Verantwortung er trägt.



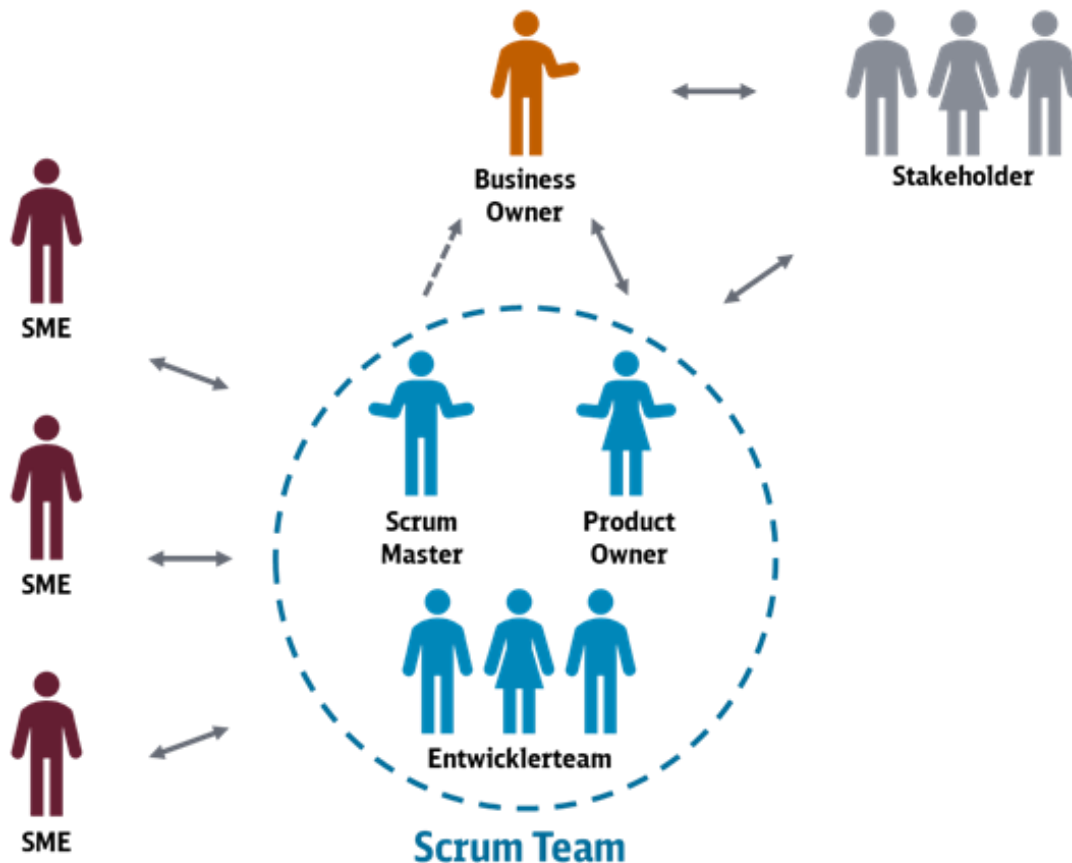
Technologische Grundlagen

Agile Softwareentwicklung	15
No-Code- und Low-Code-Entwicklung	17
Intelligente Prozessautomation	18
Selbstkonfigurierende und selbstintegrierende Plattformen	19
Künstliche Intelligenz in der Softwareentwicklung	20

Technologische Grundlagen

Um DIY Computing möglich zu machen, spielen viele unterschiedliche technologische Konzepte und Softwarewerkzeuge zusammen. Davon wurden viele Themen bereits auf den vorherigen Seiten genannt und teilweise auch kurz erklärt. Im vorliegenden Kapitel beschreiben wir diese Technologien detaillierter und wollen auch Interessierten ohne IT-Hintergrund die Sachverhalte möglichst verständlich und transparent erläutern.

Agile Softwareentwicklung



Zunächst eine Vorbemerkung zu [Agiler Softwareentwicklung](#): DIY Computing wird durch [Agilität](#) nicht ermöglicht und ist auch keine Grundvoraussetzung dafür. Eine No- bzw. Low-Code-Entwicklung könnte auch im Rahmen eines klassischen Vorgehensmodells arbeitsteilig erfolgen. Allerdings arbeiten bei agiler Vorgehensweise IT- und Fachspezialistinnen enger Hand in Hand als bei klassischer Softwareentwicklung. Dies fördert das gegenseitige Verstehen. Ein wichtiger Punkt, der sich auch in der [Technologiestrategie der DB System](#) findet: „Die Förderung von Entwicklungskompetenz profitiert auch enorm dadurch, dass Teams ihre Entscheidungen und Erfahrungen transparent machen und so Eigenschaften offenlegen und den **Wissensaustausch fördern**.“¹ Da sich DIY Computing bei Nicht-IT-Expert:innen, die ein gewisses technologisches Grundverständnis haben, besser entwickeln wird, widmen wir dem Thema einen eigenen Abschnitt.

Mit einer agilen Vorgehensweise soll die Transparenz und Geschwindigkeit bei der Softwareentwicklung erhöht werden. Selbstorganisierte, gemischte Teams und kleine iterative Entwicklungsschritte fokussieren auf die schnelle Entwicklung und den Roll-out von neuen Funktionalitäten. Durch die **Verlagerung des Schwerpunkts** von der Entwurfs- auf die Entwicklungsphase und die inkrementelle Vorgehenseise verspricht man sich erhöhte Flexibilität, Qualität und Schnelligkeit in der Softwareentwicklung.

¹ Technologiestrategie der DB System, S. 58

Historisch wurde in der IT-Produktion, wie in anderen Bereichen auch, stark arbeitsteilig Software entwickelt. Durch eine hohe Standardisierung und Verteilung der Arbeit [↗ auch ins Ausland](#), versprach man sich in Zeiten der Globalisierung Kostenvorteile und Skaleneffekte. So wurden große IT-Verfahren in stark standardisierten Vorgehensmodellen (u.a. [↗ Wasserfall-Modell](#), [↗ V-Modell](#)), umgesetzt. Durch die Fokussierung auf Entwurf und Beschreibung des zu liefernden Gewerks versprach man sich zumindest mittelfristige Planungssicherheit. Schnelligkeit und Flexibilität standen nicht im Vordergrund.

Mit fortschreitender Digitalisierung haben sich jedoch die Marktregeln geändert. Skalierungseffekte allein entscheiden nicht mehr über einen großen Markterfolg. Die Fähigkeit, **schnell neue (digitale) Produkte mit großem Nutzen für die Kund:innen** zu generieren, haben die Marktregeln grundlegend geändert: neue Geschäftsmodelle und Märkte entstehen, bestehende Markteintrittsbarrieren sinken oder entfallen ganz durch digitale Lösungen. Eine an den Kund:innen ausgerichtete wertorientierte IT ist in dieser neuen Welt ein Schlüsselement zum Wettbewerbserfolg. Der Einsatz von Methoden wie dem Wasserfall- oder V-Modell werden aus diesem Grund zunehmend von agilen Vorgehensweisen ersetzt. Methoden wie [↗ Scrum](#), [↗ Design Thinking](#), [↗ Kanban](#) oder [↗ Prototyping](#) werden heute in unterschiedlichsten Kombinationen eingesetzt, um eine höhere Flexibilität und einen schnelleren Kundennutzen in der Softwareentwicklung zu erzielen.

Auch wenn in gemischten agilen Teams fachliche und technologische Expert:innen eng zusammenarbeiten, so basiert der Kern der Softwareentwicklung weiterhin auf klassischen Programmiersprachen. Für diese werden immer noch gut ausgebildete Softwareentwickler:innen und IT-Architekt:innen benötigt. Der **Bedarf an digitalen Lösungen** steigt jedoch weiter so stark, dass verfügbare Programmierkapazitäten zum Flaschenhals werden. Einen Ausweg aus dieser Situation bieten Werkzeuge, die aufgrund intuitiver Bedienung, die Entwicklung von Software auch Fachfremden und damit einer breiteren Basis zugänglich macht.

No-Code- und Low-Code-Entwicklung

Der **zunehmende Mangel an IT-Expert:innen**, die klassisch programmieren, steht einem wachsenden Bedarf an Applikationen aufgrund der Digitalisierung entgegen. Durch sogenannten [↗ No-Code-](#) und [↗ Low-Code-Plattformen](#) erhofft man sich einen Weg aus diesem Dilemma: eine **intuitive und grafische Benutzeroberfläche** befähigt auch IT-Laien, ausführbare Applikationen zu generieren, um ihre fachlichen Probleme zu lösen und Prozesse zu digitalisieren.

Eine Hoffnung, die sich in großen Wachstumszahlen und Gründen für den Einsatz von Low-Code-Plattformen ausdrückt: so prognostiziert Gartner für dieses Jahr ein Marktwachstum für Low-Code-Entwicklungsumgebungen von [↗ mehr als 20%](#). Als Grund für den Einsatz von Low-Code-Werkzeugen geben laut Forrester [↗ annähernd 40 % befragter IT Manager](#) an, „Mitarbeitende außerhalb der IT zu befähigen, Anwendungen bereitzustellen“.

No-Code und Low-Code-Entwicklungswerkzeuge versprechen Softwareentwicklung auch ohne große Programmierkenntnisse. Nach dem Versprechen der Hersteller wird dies vor allem durch intuitive grafische Benutzerschnittstellen möglich. Anwender:innen wählen aus einem Baukasten grafische Elemente, die sie über Möglichkeiten wie [↗ Drag and Drop per Maus](#) oder direkte Berührung auf Touchscreens miteinander in Bezug setzen. **Grafische Elemente ersetzen dabei die [↗ Syntax einer Programmiersprache](#).**

Da diese Werkzeuge aufgrund der Komplexität eines Problems an Ihre Grenzen stoßen können, bieten Low-Code Werkzeuge Möglichkeiten, die Vorteile der grafischen mit der klassischen Programmierung zu verbinden. [↗ Im Unterschied dazu](#) verzichten No-Code-Plattformen auf diese Option. Neben den Einsatzmöglichkeiten unterschiedlicher Programmierarten unterscheiden sich die Tools auch hinsichtlich des zu betrachtenden Problemraumes: Werkzeuge können eine bestimmte Branche oder Domäne adressieren oder einen eher generalistischen Ansatz verfolgen.

Wenn wir uns vorstellen, dass in einem großen Unternehmen eine Vielzahl von Mitarbeitenden ohne IT- oder Programmierkenntnisse über No-Code und Low-Code-Plattformen Lösungen entwickeln, liegen **mögliche Probleme** auf der Hand:

- viele unterschiedliche Lösungen für ähnlich gelagerte Probleme werden erstellt,
- Sicherheitsaspekte werden nicht ausreichend berücksichtigt und ermöglichen bösartige Hackerangriffe,
- Anforderungen des Datenschutzes werden nicht eingehalten, was Schadensersatzforderungen nach sich ziehen kann,
- fehlendes Know-how im Umgang mit Daten kann zu Datenverlusten oder falschen Daten führen,
- Skalierungserfordernisse werden nicht einbezogen, d.h. eine Lösung kann für wenige Berechnungen gut, bei einer Masse an Daten aber nicht mehr funktionieren.

Wie aus diesen wenigen Beispielen bereits deutlich wird, bedarf es eines gewissen **Ordnungsrahmens**, um derartige Risiken zu vermeiden. Wir werden diesen Punkt später bei den organisatorischen Gesichtspunkten näher beleuchten. Außerdem sollten Nicht-IT-Expert:innen, die No-Code- oder Low-Code-Plattformen nutzen, ein technologisches Grundverständnis mitbringen und in Teilen **wie ein IT-Profi denken**. Auch wenn bei der Programmierung eine sprachliche Syntax durch grafische Elemente ersetzt wird, ist ein Grundwissen zur Verwendung der grafischen Elemente erforderlich.



Intelligente Prozessautomation

Spricht man heute von Prozessautomation im Büro, denkt man als erstes an Werkzeuge, mit denen sich **wiederkehrende Aufgaben** automatisch von IT-Systemen erledigen lassen. Um dies zu erreichen werden „**Robotic Process Automation (RPA)**“ Lösungen eingesetzt. Der Name, sinngemäß ins Deutsche übersetzt „Roboter gestützte Prozessautomatisierung“ kann dabei auch falsch verstanden werden. Wir sprechen hier nicht von Robotern, die z.B. in einer Fertigungsstraße Schweißarbeiten übernehmen, sondern über Software, die eine wiederkehrende **Büroaufgabe**, wie bspw. eine Rechnungsprüfung, automatisch erledigt.

Bei der Aufzeichnung von Prozessen können dabei No-Code-Ansätze zum Tragen kommen. Am Bildschirm werden Mausklicks und Eingaben aufgezeichnet, um sie dann wiederholt ablaufen lassen zu können. Eine gute RPA Software sollte in der Lage sein, den Code auszuführen, wenn sich der Bildschirm oder seine Auflösung ändert.

Wenn wir weiter in die Zukunft blicken, werden Fachanwender:innen noch besser unterstützt. Verschiedene Technologien werden weiter zusammenwachsen. Insbesondere **KI wird dabei unterstützen**, Änderungen zu erkennen und die Automatisierung entsprechend anzupassen. Bereits heute werben RPA-Anbieter wie [↗ Kofax](#) oder [↗ Workfusion](#) damit, anhand ihrer intelligenten Automatisierungslösungen signifikante Einsparpotentiale zu realisieren.

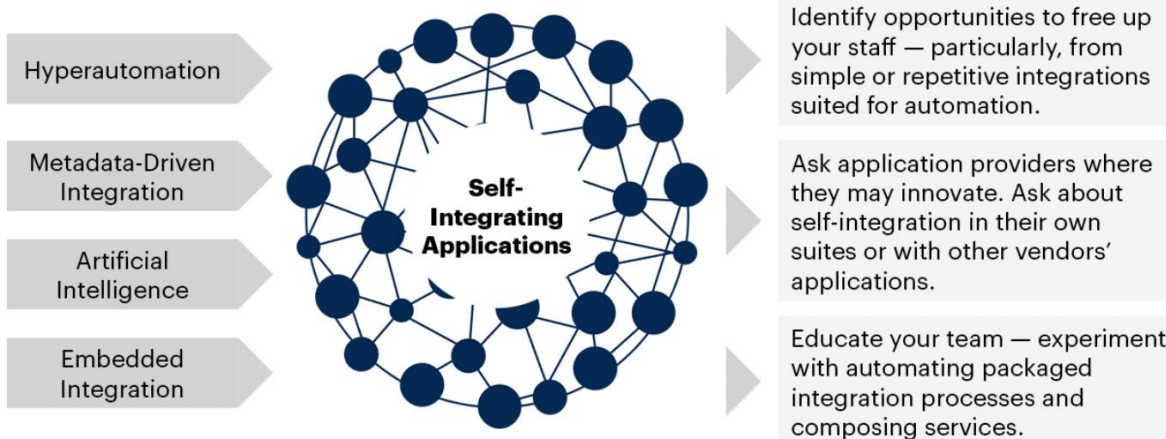
Durch den Einsatz von [↗ Machine Learning](#) werden auch weniger versierte Anwender:innen die Software nutzen können und bei reduziertem manuellen Anpassungsbedarf Stabilität und Qualität der RPA Lösungen verbessert werden. Das Analysehaus Gartner geht in seinen Prognosen noch einen Schritt weiter: durch den koordinierten Einsatz mehrerer Technologien und Tools wie RPA, KI, Low-Code bzw. No-Code etc. wird intelligente Prozessautomatisierung, d.h. die Identifizierung, Automatisierung und Optimierung von Geschäftsprozessen, möglich.



Selbstkonfigurierende und selbstintegrierende Plattformen

In gewachsenen Anwendungslandschaften finden sich eine Vielzahl von Insellösungen und Plattformen, die oft nicht miteinander verknüpft sind. So müssen bei der Umsetzung neuer Anforderungen oder der Neugestaltung bzw. Optimierung bestehender, übergreifender Geschäftsprozesse mitunter erhebliche Anstrengungen unternommen werden, um **Brücken zwischen Anwendungsinselfen** zu bauen und diese zu integrieren. Ein nicht unerheblicher Teil der Kapazitäten von IT-Professionals fließt in die Verknüpfung bestehender Services, Anwendungen und Plattformen, um die fachlichen Anforderungen durch neue, optimierte oder geänderte Geschäftsprozesse abzubilden.

Enablers for Self-Integrating Applications



Source: Gartner
736682_C

Gartner

Quelle: Gartner, Innovation Insight for Self-Integrating Applications, Eric Thoo, Keith Guttridge, March 10, 2021

Wäre es denkbar, dass ein fachlicher Prozessverantwortlicher seine Erfordernisse an die IT selbst umsetzt? Und dies auch vor dem Hintergrund einer komplexen [Systemintegration](#)? In einem kürzlich erschienenen [Bericht sagte Gartner voraus](#), dass "**selbstintegrierende Applikationen**" innerhalb von 5 bis 10 Jahren zur Selbstverständlichkeit werden.² Diese setzen eine Kombination aus automatischer Service-Erkennung, einem intelligenten Metadatenmanagement sowie das Erkennen von Abhängigkeiten zwischen Anwendungen und Services voraus. Auf diese Weise könnten in der Zukunft Applikationen in die Lage versetzt werden, sich mit minimaler menschlicher Interaktion in ein bestehendes Anwendungssportfolio zu integrieren.

Auf dem Weg zu diesem Zielbild werden gekapselte Softwarekomponenten unterstützen, die klar definierte **Geschäftsfähigkeiten** ([Packaged Business Capabilities](#)) abbilden und in neue Lösungen integriert werden können. Der „[Citizen Integrator](#)“ der Zukunft wird darauf ebenso zugreifen können, wie auf andere in der Cloud verfügbare Dienste, die intuitive und codefreie Werkzeuge zur Entwicklung von Integrationsprozessen anbieten. Auch wenn wir dabei von einer „Selbstintegration“ entfernt sind, so wird der Geschäftsanwender:innen schnell und ohne Hilfe von IT-Spezialist:innen eine Vielzahl von Integrationsaufgaben selbst ausführen können.

² Gartner, "Hype Cycle™ for the Future of Applications, 2021", Micky Keck, Yefim Natis, Stefan Van Der Zijden, August 10, 2021

Künstliche Intelligenz in der Softwareentwicklung

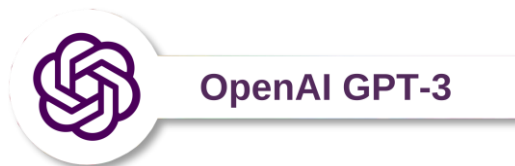
Im Mai 2017 veröffentlichten die Forscher von Google Brain „[↗ AutoML](#)“, eine Künstliche Intelligenz, die in der Lage ist, selbstständig weitere KIs zu generieren. Mittels [↗ Reinforcement Learning](#) generiert das System Machine-Learning-Modelle für eine Echtzeit-Objekterkennung in Videos. AutoML erschafft gewissermaßen sein eigenes „künstliches Kind“, das früheren von Menschen programmierten KIs überlegen ist.

„Künstliche Intelligenz ist das Ergebnis von Programmierung. Jetzt ist Programmierung das Ergebnis von künstlicher Intelligenz“³ heißt es in einem [↗ Forbes Artikel](#) vom Februar 2020. Sind die Maschinen der Zukunft also die besseren Programmierer? In Anbetracht des massiven Fachkräftemangels an IT-Kräften und vor allem KI-Entwickler:innen, eine verführerische Vorstellung. Laut einem [↗ Deloitte Report](#) zu „[AI-assisted software development](#)“ setzen immer mehr Unternehmen auf unterschiedliche Werkzeuge zur **KI-gestützten Softwareentwicklung**, darunter auch [↗ Tech-Giganten wie IBM oder Microsoft](#).



Im Zuge der Erwartungshaltung wöchentlicher Releases und Updates wächst der Druck bei Entwicklungs- und Testteams. „Kolleg:in Roboter“ könnte hier als Unterstützung für den menschlichen Entwickler:innen im „[↗ Pair Programming](#)“ oder als „Co-Pilot“ eingesetzt werden. Auf diese Weise können KI-gestützte Systeme bereits heute Code schreiben und überprüfen, Fehler identifizieren, Software testen und Entwicklungsprojekte optimieren. Viele der heute verfügbaren Lösungen funktionieren ähnlich wie Rechtschreib-, Grammatikprüfung oder „Autovervollständigung“. Durch das **Vorschlagen geeigneter Codezeilen oder Funktionen** können Entwickler:innen ihre Produktivität erhöhen. Gleichzeitig werden Fehler reduziert sowie Tests automatisiert, die zur Überprüfung der Qualität erforderlich sind. Durch die Analyse großer Datenmengen mit Hilfe von Machine Learning werden darüber hinaus Erkenntnisse gewonnen, was „guten“ bzw. „schlechten“ Code ausmacht. Dadurch können professionelle Softwareentwickler:innen in ihrer täglichen Arbeit produktiver werden, aber auch eine immer größere Personengruppe ohne spezifische Vorkenntnisse Lösungen entwickeln.

Der [↗ GitHub Copilot](#) bspw. ist ein **KI-Pair-Programmierer**, der Entwickler:innen dabei helfen soll, besseren Code zu schreiben. Dabei interpretiert das Programm den Kontext des Codes, an dem gerade gearbeitet wird und schlägt neue Programmzeilen und Funktionen vor. Auf diese Weise unterstützt der GitHub Copilot dabei, Alternativen zur Problemlösung zu finden, Tests zu generieren oder neue APIs ohne langwierige Suche zu finden.



Die Forschungsinitiative [↗ OpenAI](#), die u.a. von Elon Musk und Microsoft unterstützt wird, hat [↗ GPT-3](#) entwickelt, das **aktuell fortschrittlichste Sprachverarbeitungsmodell**. Es ist in der Lage, menschenähnliche Texte und Zeitungsartikel zu verfassen. Auch soll es Aufgaben ausführen können, für die es nie geschult wurde, wie z.B. [↗ das Schreiben von SQL-Abfragen oder Javascript-Programmen](#) auf Basis von Spracheingaben.

Den vielversprechenden Anwendungsgebieten stehen gleichzeitig **ethische und rechtliche Fragen** gegenüber: Wer haftet im Falle sich selbst entwickelnder Systeme für mögliche Fehler oder Verstöße im

³ Im Original: Artificial Intelligence is the result of coding. Now coding is the result of artificial intelligence

Bereich Security oder Datenschutz? Bei maschinell generierten Codes ist mitunter für den Menschen nicht mehr transparent, auf welchen Entscheidungen oder konkreten Zielsetzungen Programmzeilen und Algorithmen basieren. „[↗ Explainable AI](#)“ wird zu einer zwingenden Voraussetzung.

Das Ableiten von Code auf Basis bereits bestehender Modelle kann außerdem Innovationskraft vermindern, die erst durch **menschliche Kreativität** entsteht. Zweitklassige Lösungen und sogar Fehler können sich bei mangelnder Kontrolle im Zeitalter von [↗ Open Source](#) in immenser Geschwindigkeit replizieren und verbreiten. Auch der übermäßige **Energieverbrauch**, der durch Methoden wie Reinforcement Learning entsteht, sollte nicht ignoriert werden. Ohne begleitende Kommunikation wird die zunehmende Automatisierung zu Widerständen aufgrund der Angst vor Arbeitsplatzverlust führen.

Auch wenn die vorgestellten Beispiele beeindruckend wirken, sollten sie nicht überschätzt werden. Die Mehrzahl der aktuellen Anwendungen sind **bessere „Autovervollständiger“**, bei denen das System neue Programmzeilen auf Grundlage seiner bestehenden „Erfahrungsbasis“ vorschlägt. Zeitliche Ersparnisse bei der Programmierung oder Erleichterungen beim Testen sind willkommene positive Effekte. Insbesondere bei hochkomplexen IT-Projekten mit ständig wechselnden Anforderungen können sie die menschliche Kreativität nicht ersetzen. Dieser Komplexität ergänzt um weitere Herausforderungen wie Performanz, Sicherheit, Datenschutz oder Benutzerfreundlichkeit wird sich keine Maschine innerhalb des nächsten Jahrzehnts stellen können. Ob sich derartige Systeme dennoch langfristig zu echten nutzenstiftenden „Partnern“ entwickeln, ist heute noch nicht abzusehen. Dementsprechend ist es eher unwahrscheinlich, dass zukünftige KI-Anwendungen Programmierer mit ihrer Kreativität ersetzen werden. Wir sind aber überzeugt davon, dass KI-gestützte Softwareentwicklung positiv zur Demokratisierung der Technologie beitragen wird.



Die Demokratisierung der IT

Der „Citizen Developer“	23
Gesetzliche Rahmenbedingungen und die Rolle der Arbeitnehmervertreter:innen.....	25
Anwendungsfälle und Herausforderungen für Citizen Development	26

Die Demokratisierung der IT

Auf den vorhergehenden Seiten haben wir Hintergründe rund um Technologie und Organisation vorgestellt. Doch wie wirkt sich DIY Computing auf uns und unser Arbeitsleben aus? Um diese Frage zu beantworten, gehen wir im Detail auf den Begriff des Citizen Developer ein. Auch beleuchten wir Aspekte rund um Daten- und Arbeitnehmerschutz, die in einer Welt des DIY Computing überprüft und gegebenenfalls angepasst werden müssen.

Der „Citizen Developer“

Die COVID-19 Pandemie und die damit einhergehende Remote Arbeit haben den Digitalisierungsdruck auf Unternehmen erhöht. Um in einem flexiblen und dynamischen Umfeld zu bestehen, bedarf es einer besseren **Befähigung und intuitiver Werkzeuge** für die Mitarbeitenden. Die Digitalisierung und damit Wettbewerbsfähigkeit eines Unternehmens bedeutet, dass Prozesse, Verfahren, Richtlinien, Daten und Informationen schnell in Software übersetzt und kontinuierlich verbessert werden können.

Wie schon zuvor festgestellt, werden diese Aufgaben aus Kapazitätsgründen nicht ausschließlich durch IT-Spezialist:innen gelöst werden können. Soweit passend muss **IT demokratisiert** werden. Mitarbeitende beteiligen sich im Rahmen ihrer Interessen und Fähigkeiten als Citizen Developer an der Entwicklung und gestalten dadurch selbstbestimmt die durch sie angestrebte digitale Lösung. Demokratisierte IT sollte es diesen Mitarbeitenden ermöglichen, ihre Arbeit zu optimieren und Schwachstellen selbst zu beheben, ohne IT-Projekte beauftragen zu müssen. Um dies zu erreichen, bedarf es intuitiver Werkzeuge für No-Code- bzw. Low-Code-Entwicklung oder RPA.

Zunächst drängt sich der Eindruck auf, dass demokratisierte IT-Entwicklung nicht besonders neu ist. Mit dem Siegeszug von **Office-Anwendungen** in den vergangenen Jahrzehnten war es zunehmend möglich, Aufgaben in Tabellenkalkulations- oder Textverarbeitungsprogrammen zu **automatisieren** und die Anwendungen hinsichtlich der eigenen Bedarfe anzupassen. So hat Microsoft bspw. mit [Visual Basic for Applications](#) eine eigene Programmiersprache in das Office Programmpaket integriert. Dabei handelt es sich um eine aus Basic abgeleitete Syntax, die sich an technisch interessierte Office-Anwender:innen richtet. Citizen Development greift wesentlich weiter: Interessierte, die nicht syntaktisch programmieren können und wollen, sollen durch visuelle Werkzeuge angesprochen werden.

Entsprechend der [Technologiestrategie der DB System](#) werden Citizen Developern Werkzeuge und Methoden zur Verfügung gestellt, die „Nicht-Fachleuten eine Beteiligung an der Entwicklung digitaler Lösungen ermöglichen“. D.h. um Citizen Development erfolgreich zu machen, bedarf es nicht nur intuitiver Werkzeuge, sondern auch **Methoden und Prozesse**, die neben fachlichen Anforderungen auch **nicht-funktionale Sachverhalte** rund um Verantwortlichkeiten, Prozesse und Fähigkeiten einbeziehen.

Jede Person, die über Interesse und eine gewisse technologische Wissensbasis verfügt, könnte in der Welt des DIY Computing ein:e Entwickler:in sein. Je nach Befähigung und Komplexität differenzieren wir unterschiedliche Arten von Entwickler:innen. Es kann bspw. zwischen professionellen (Pro-) Entwickler:innen und Citizen Developern unterschieden werden, die ihrerseits wieder in „Power User“ und „End User“ aufgeteilt sind:

Anyone Can Be a Developer

	Citizen Developer (End User)	Citizen Developer (Power User)	Business Unit Pro Developer	Enterprise IT Pro Developer
Full-Time Developer	No	No	Yes	Yes
Preferred Tools	"No-code"	Low-code	Low-code & Pro-code	Low-code & Pro-code
Typical Apps	Individual & Workgroup	Workgroup	Departmental	Enterprise

Source: Gartner (April 2021)

749246_C

Gartner

Quelle: Gartner, How to Define and Guide Citizen Development Practices, Jason Wong, Saikat Ray, Wan Fui Chan, Adrian Leow, April 20, 2021

Mit fortschreitender Intuitivität und Intelligenz der Tools werden sich Komplexitätsgrenzen zukünftig verschieben. Citizen Developer werden auch schwierigere Probleme mit geeigneten KI-gestützten Werkzeugen als heute bearbeiten. Unabhängig davon muss ein Citizen Developer bereits heute **in Grundlagen technologisch denken** können und ein Gefühl für so unterschiedliche technische Sachverhalte wie IT-Sicherheit, Benutzeroberflächen oder Dokumentation entwickeln. In diesem Zusammenhang sollte das Risiko von „**Wissenssilos**“ nicht außer Acht gelassen werden: wenn ein Citizen Developer das Unternehmen verlässt, stellt sich die Frage, ob seine Applikationen auch von Dritten verwendbar sind. Wie man aus diesen Herausforderungen lesen kann, werden Initiativen scheitern, die versuchen, jeden zum Citizen Developer zu machen.

Gesetzliche Rahmenbedingungen und die Rolle der Arbeitnehmervertreter:innen

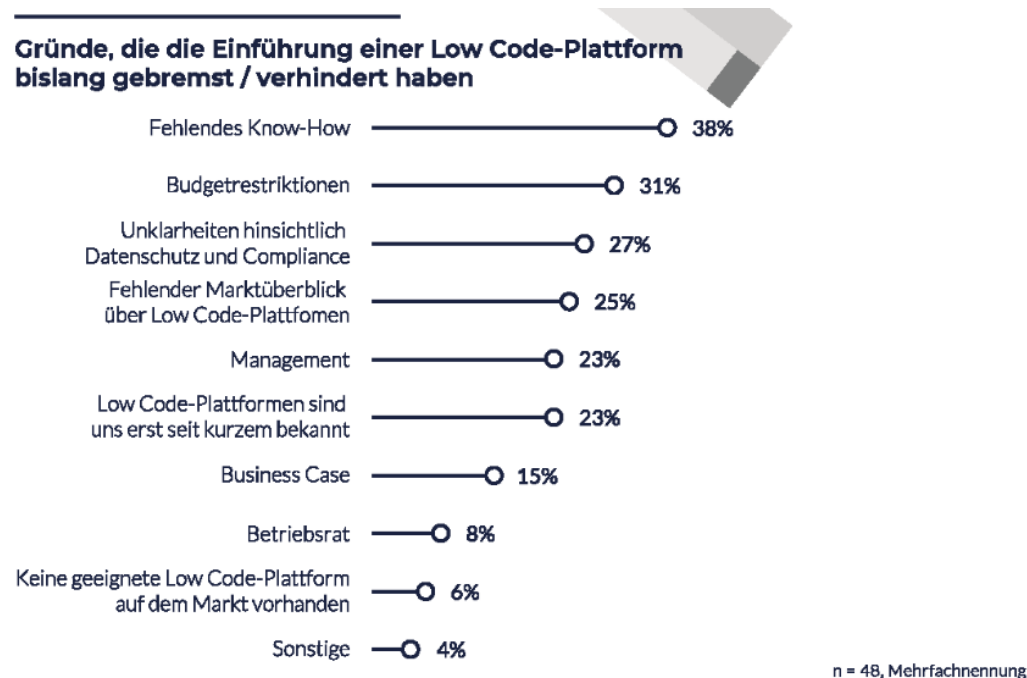
Bei der Inbetriebnahme von IT-Systemen sind neben technischen auch **formale Aspekte** zu betrachten. Gesetzliche Regelungen müssen beachtet und eingehalten werden. So unterliegen Anwendungen, über die „Verhalten oder Leistungen von Arbeitnehmern“ nachverfolgt werden können, einer Mitbestimmungspflicht nach [§ 87 Abs. 1 Nr. 6 BetrVG](#). Mit anderen Worten: der zuständige Betriebsrat muss seine Zustimmung geben, bevor entsprechende Systeme in den Betrieb gehen dürfen.

In der heutigen IT-Welt mit einer klaren Rollenverteilung und bewährten Prozessen sind entsprechende **Kontrollmechanismen** fest verankert. Wie bspw. mit [§ Schutzbedarfen](#) und Risikobewertungen umzugehen ist, ist für die Einführung neuer Systeme klar festgelegt. Das stellt DIY Computing vor Herausforderungen, die durch die folgenden drei Sachverhalte verstärkt werden:

- wird eine Low-Code-Plattform als Entwicklungstool für IT-Spezialist:innen freigegeben oder als Citizen-Developer-Plattform, auf der auch Anwender:innen Lösungen erstellen können? Letzteres muss der Fall sein, wenn Fachexpert:innen selbst entwickeln.
- Darüber hinaus zu betrachten sind nicht nur die Plattformen, mit denen der Citizen Developer arbeitet, sondern auch die Lösungen bzw. Applikationen, die darauf erstellt werden.
- Nicht nur bei einer Einführung von neuen Applikationen sind gesetzliche Vorgaben einzuhalten, sondern auch bei einer Veränderung an bestehenden Anwendungen,

Da verwundert es nicht, dass befragte Führungskräfte in der Studie „[§ Low Code Development](#)“ von CRISP Research als dritthäufigsten Hinderungsgrund für die Einführung derartiger Systeme „**Unklarheiten hinsichtlich Datenschutz und Compliance**“ angeben.

Wenn wir in die Zukunft blicken, werden Anwender:innen, sei es bei der Modellierung und Automatisierung von Prozessen, wie auch bei der Entwicklung von Applikationen zunehmend von KI unterstützt. Bzgl. KI werden aktuell unter dem Begriff [§ Digitale Ethik](#) Sachverhalte aus verschiedenen Perspektiven, wie z.B. Datenschutz, Verbraucherschutz oder Diskriminierungsfreiheit diskutiert, die in absehbarer Zeit in Regelwerken und Gesetzen festgeschrieben werden. Nach derzeitiger Lage wäre es mindestens fraglich, wenn nicht sogar rechtlich bedenklich, Anwendungen einzusetzen, die durch selbstlernende Systeme entwickelt wurden. Auch wenn der Citizen Developer nur durch die KI unterstützt wird, müsste von Gremien geprüft werden können, wie ein Algorithmus zu einer Entscheidung kommt.



Quelle: CRISP Research Studie „Low Code Development“, 2019

Anwendungsfälle und Herausforderungen für Citizen Development

Wie wir gesehen haben, lauern auf dem Weg zu einem erfolgreichen Citizen Development im Unternehmen verschiedene Herausforderungen. Nicht zu unterschätzen ist der menschliche Faktor. **Nicht jeder Mitarbeitende** eignet sich dafür oder will technologisch arbeiten. Neben den Herausforderungen rund um Qualifizierung, Befähigung und der Begeisterung für Technologie, müssen rechtliche und organisatorische Rahmenbedingungen geklärt werden. Unabhängig davon lohnt es sich, einen Blick auf präferierte Einsatzfelder und auch einige Beispiele zu werfen.

Citizen Development ist bereits heute bei **weniger komplexen und nicht-unternehmensweiten Anwendungsfeldern** erfolgreich im Einsatz. In einer [Umfrage haben die Analysten von Gartner](#) folgende besonders geeignete Anwendungsfelder herausgearbeitet:

- Apps für Formulare und Datenerfassung (57%)
- Apps zur Orchestrierung von Geschäftsprozessen und Workflows (49 %)
- Apps zum Ersetzen von Papier, E-Mail und Spreadsheets (42 %)

Auch wenn derartige Einsatzmöglichkeiten begrenzt erscheinen, bieten sie erste Ansatzpunkte, um Entwicklungen auf der Fachseite zu erproben. Mit zunehmender Erfahrung, besseren Tools und Klärung organisatorischer Rahmenbedingungen könnten auch andere Anwendungsfelder in Betracht kommen. Doch auch bei einfacheren Anwendungen gilt es, Hürden zu überwinden und Vorbehalte abzubauen.

Kulturelle Schwierigkeiten können erheblich bremsen. Teams auf der Fachseite sind es nicht gewohnt, Technologie selbst zu entwickeln. Wenn eine Einführung z.B. von No-Code-Tools unbegleitet und unmoderiert abläuft, könnten Widerstände und damit Scheitern die Folge sein. Außerdem hatten wir bereits festgestellt, dass nicht alle Mitarbeitenden Technologie selbst entwickeln wollen. Unternehmen sollten daher nicht versuchen, für Einzelpersonen eine Rolle als Citizen Developer vorzusehen. Vielmehr können durch gute Kommunikation einer Citizen Development Strategie und die Sichtbarmachung von ersten Erfolgen eine kulturelle Basis geschaffen werden.

Die Berater von [Accenture stellen in ihrem Technologie Trend „Ich, der Technologe“](#) ein Beispiel vor: bei G&J Pepsi wird zur Low-Code-Entwicklung Microsoft Power-Apps im Bereich Inventarisierung eingesetzt. Ein:e Mitarbeiter:in ohne Softwareentwicklungserfahrung erstellt bspw. eine App, die ein Bild eines Regals untersucht, um die Anzahl und Art der darauf befindlichen Flaschen zu identifizieren und dann automatisch die richtigen Artikel für die Wiederauffüllung auf der Grundlage historischer Trends zu bestellen. Laut Accenture erstellten Citizen Developer in der Arbeitsgruppe acht Anwendungen ohne einen IT-Profi und ermöglichten dadurch signifikante Einsparungen.



Partnerschaftliche Zusammenarbeit zwischen IT- und Fachseite

Die klassische IT-Organisation	28
Herausforderungen für den Digitalisierungspartner der Zukunft	29
Herausforderungen für die Digitalanwender:innen der Zukunft.....	30
Neue Formen der Zusammenarbeit und Geschäftsmodelle	32

Partnerschaftliche Zusammenarbeit zwischen IT- und Fachseite

Im Bereich der Informationstechnologie waren Märkte und Organisation bisher von einer starken Arbeitsteilung geprägt. Die digitale Umsetzung von fachlichen Anforderungen liegt bei ausgebildeten Spezialist:innen und Programmierern. Entsprechend ausgerichtet waren die klassischen Geschäftsmodelle zwischen Digitalisierungs- und Fachpartnern. Im Folgenden werden wir zunächst noch einmal die uns bekannte Welt vorstellen, bevor wir auf die durch DIY Computing entstehenden organisatorischen Herausforderungen und Lösungen eingehen.

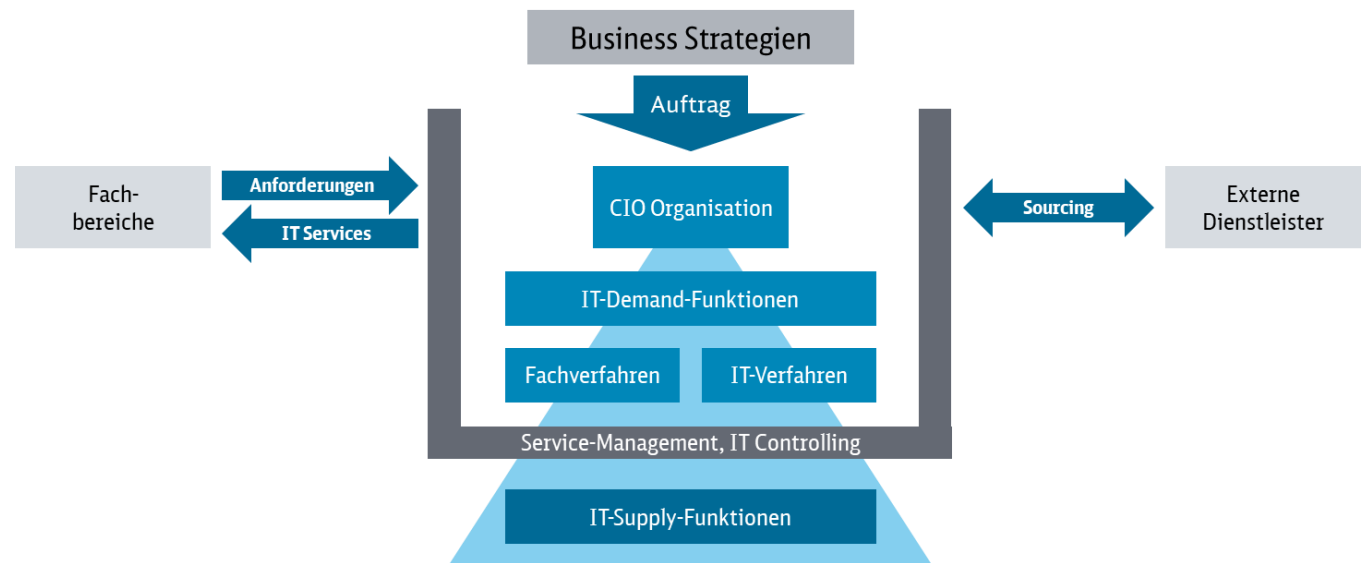
Die klassische IT-Organisation

In der Vergangenheit war eine IT-Organisation häufig rein technisch ausgerichtet. Mit zunehmendem Schwerpunkt auf Fachlichkeit hat sich die IT zunehmend zu einer **kritischen Businessfunktion** entwickelt. Wenn IT zu stark von der Fachseite entkoppelt ist, bestehen Risiken, die Anforderungen der Geschäftsseite nicht wie erforderlich zu erfüllen.

Seit Anfang der Zweitausender Jahre hat sich mit dem **IT-Demand und Supply-Modell** eine Organisationsform etabliert, die den wachsenden Business-Herausforderungen Rechnung tragen soll und heute in vielen Unternehmen etabliert ist. In diesem Modell wird zwischen nachfragenden und anbietenden Einheiten ein Demand Management angesiedelt, in dem die für die IT-Steuerung notwendige Fach- und IT-Kompetenzen gebündelt werden. Das Sourcing und die Beauftragung von IT-Leistungen werden im Idealfall aus dieser Funktion gesteuert.

Durch die fachliche und technische Kompetenzbündelung in einem **Demand Management**, erhofft man sich Spannungen zu minimieren, die zwischen einer aus dem Geschäft getriebenen Nachfrageseite und einer auf Effizienz ausgerichteten Anbieterseite entstehen. Diese Reibungsflächen können sich bei föderalen Organisationen und Konzernen durch eine Mehrstufigkeit der Verantwortungen noch vergrößern.

Auch in diesem Modell geht man jedoch von einer **Trennung** zwischen fachlicher Nachfrage und IT-seitiger Umsetzung aus. Die Herausforderungen, die durch ein DIY Computing entstehen, d.h. durch die direkte Umsetzung von geschäftlichen Anforderungen auf der Fachseite, können durch dieses Organisationsmodell nicht gelöst werden.



Quelle: [Das Referenzmodell für IT Management](#)

Herausforderungen für den Digitalisierungspartner der Zukunft

Auch in der Zukunft wird die klassische IT-Entwicklung bei der Lösung komplexerer Anforderungen gefragt sein. Doch die **Komplexitätsgrenze**, ab der Citizen Development sinnvoll und möglich ist, wird sich perspektivisch aufgrund immer intelligenterer Technik verschieben. Dies führt einerseits zu einer Entlastung der IT-Spezialist:innen, da nicht jede Anforderung erfasst und durch die IT umgesetzt werden muss. Die zunehmende Anzahl von Citizen Developern in Fachabteilungen führt jedoch auch zu neuen Herausforderungen: IT-Anbieter müssen sich zu **Digitalisierungspartnern** weiterentwickeln.

So stellen beispielsweise die Analysten von [Forrester in dem Report „Low-Code Citizen Developer Programs“](#) fest, dass ein Umdenken und ein **Paradigmenwechsel in Bezug auf die Rolle der IT** notwendig werden. In einer [Studie des MIT Center for Information Systems Research \(CISR\)](#) wird die Abschaffung der traditionellen IT-Organisation prognostiziert. Vielmehr werden sich funktionsübergreifende Produktteams etablieren. Dies impliziert jedoch nicht die Abschaffung der IT, sondern vielmehr, dass jede Person „in der IT“ arbeitet.

Nach wie vor gefragt ist die Expertise von IT-Spezialist:innen bei der **Lösung komplexer Anforderungen und Probleme**. Auch wenn [Accenture in den aktuellen Top-Technologietrends](#) die Demokratisierung der Technologie vorhersagen, stellen sie fest, dass ein leistungsfähiger Digitalisierungspartner nach wie vor große Implementierungen, die Skalierung erfolgreicher Programme sowie die Identifikation und Einführung moderner Technologien verantworten wird.

Da sich Entwicklungsleistungen auf mehr, insbesondere auch fachliche Schultern perspektivisch verteilen, steigen Anforderungen an die Befähigung von Nicht-Technologen, sowie an Koordination und Orchestrierung von technologischen Lösungen. Folgende Aufgaben muss der Digitalisierungspartner deshalb im Fokus haben:

- Die **Verfügbarkeit moderner und zukunftsfester Technologien** bilden eine wesentliche Grundlage für DIY Computing. Der Digitalisierungspartner muss moderne Plattformen und Werkzeuge, die Low-Code- und No-Code-Entwicklungen ermöglichen, sowie Bibliotheken mit vorgefertigten Services, die einfach gefunden werden können, bereitstellen.
- Im vorhergehenden Kapitel hatten wir bereits mögliche Herausforderungen von Low- und No-Code-Development dargestellt: In großen Organisationen und Konzernen können redundante, sich teils sogar widersprechende Lösungen für ähnlich gelagerte Probleme entstehen. Es bedarf daher eines **Ordnungsrahmens** bzw. einer **Governancefunktion**, die Leitplanken für eine föderale Entwicklung bilden. Dies kann bis zu einer zentralen Instanz führen, die Lösungen freigibt, wie [Accenture am Beispiel des Unternehmens Schneider Electric in ihrer Technology Vision](#) darstellen.



- Damit diese Funktion nicht überstrapaziert wird, bedarf es eines grundlegenden technologischen Verständnisses bei dem Citizen Developer. Um dieses zu erreichen, wird mindestens ein entsprechendes **Training** erforderlich. Dies wird in der Regel nicht ausreichen und sollte durch vom Digitalisierungspartner angebotene **Mentorenprogramme** für technisch interessierte Mitarbeitende unterstützt werden.
- Darüber hinaus sind weitere Fragen zu lösen, die in der heutigen, noch stark arbeitsteiligen IT-Welt geregelt sind: wer übernimmt **klassische Projektmanagement Aufgaben**? Wie werden die Budgetierung und finanzielle Steuerung von IT-Vorhaben geregelt? Wer verantwortet das Qualitätsmanagement? Auch entstehen Herausforderungen rund um Datensicherheit, Datenschutz, das Risikomanagement oder die Verantwortungsübernahme für IT-Prozesse.

Grundsätzlich können diese Aufgaben bei einem Digitalisierungspartner verortet werden. Das wird aber in der sich stärker demokratisierenden Technologie-Welt nicht immer der Fall sein. Wir werden eine evolutionäre Entwicklung erleben, bei der zumindest am Beginn der Reise, ein starker Digitalisierungspartner gefragt ist.

Herausforderungen für die Digitalanwender:innen der Zukunft

Wir sprechen heute vom Zeitalter der Digitalisierung und den damit einhergehenden großen Umbrüchen. Doch was bedeutet das im Konkreten für ein Unternehmen? Das Geschäft der Zukunft wird immer mehr von Technologie durchdrungen. Es ist heute unumstritten, dass durch digitale Lösungen bestehende Kund:innenbeziehungen um digitale Komponenten bereichert werden, bis hin zum Entstehen völlig neuer Geschäftsmodelle. Diese können ganze Märkte verändern und die Existenz von großen Unternehmen gefährden. Diese Entwicklung und die damit einhergehenden **Disruptionen** werden sich zukünftig weiter beschleunigen.

Unternehmen, die digitale Lösungen schnell umsetzen und anwenden können, werden erfolgreicher sein als andere. Die damit einhergehende IT-Entwicklungskompetenz wird damit zum **strategischen Wettbewerbsvorteil**, zusätzlich verstärkt durch den wachsenden Fachkräftemangel insbesondere in den [MINT-Fächern](#).

Im erfolgreichen Unternehmen der Zukunft werden deshalb Fachanwender:innen auf sie zugeschnittene Anwendungen für weniger komplexe Anforderungen **selbst entwickeln** und nicht auf Technologieexpert:innen angewiesen sein. Die Vorteile liegen auf der Hand: kund:innenorientierte Lösungen entstehen unmittelbar am Ort des festgestellten Bedarfs. Zeitintensive Umwege für das Übersetzen von Markterfordernissen in Anforderungen und die technologische Realisierung durch fachfremde Expert:innen entfallen. Außerdem wird durch diese Demokratisierung der IT jeder Mitarbeitende selbst zu einem aktiven Teil der digitalen Transformation.



Diese Vorteile müssen jedoch hart erarbeitet werden. Es genügt nicht, ein No-Code-Werkzeug bereitzustellen, damit Anwender:innen flott loslegen, um kleine Apps selbst zu erstellen. Es sind vielmehr einige **Grundvoraussetzungen** zu erfüllen und Rahmenbedingungen zu berücksichtigen:

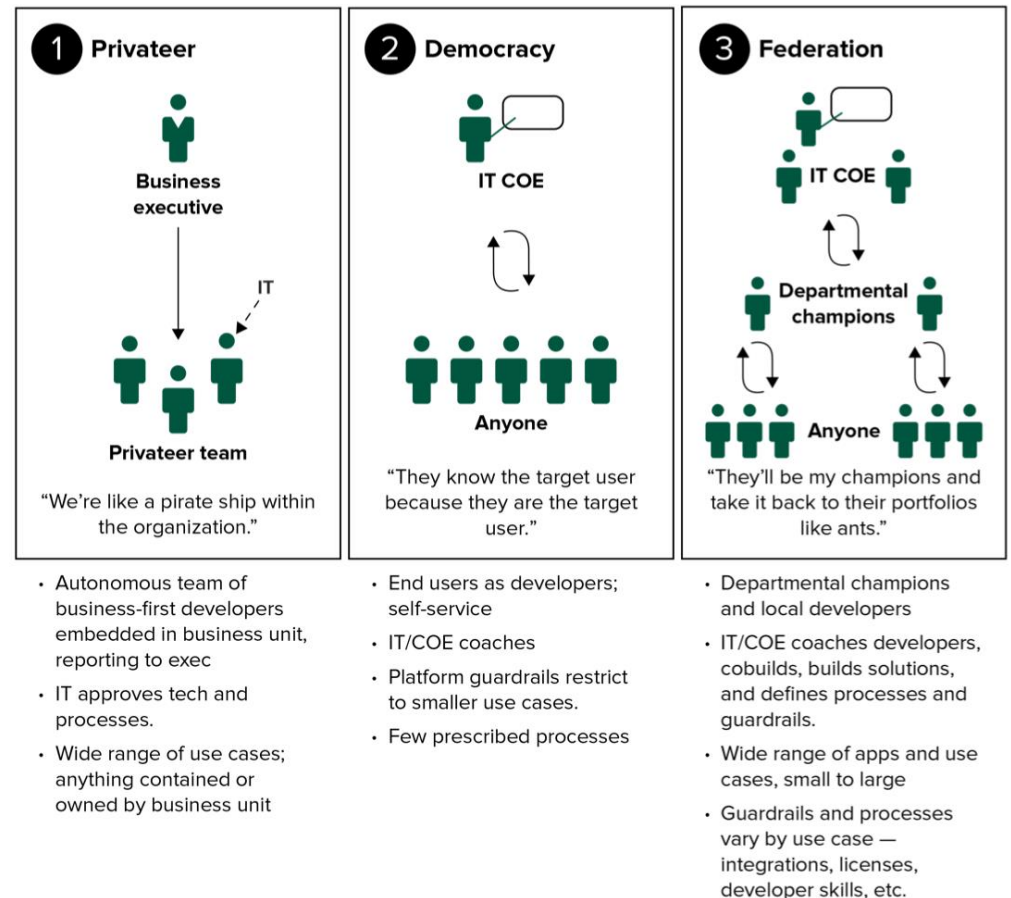
- Eine wesentliche Basis bilden die Mitarbeitenden. Die **menschliche Komponente** ist ein wesentlicher Schlüssel zum Erfolg. Wenn fachseitig entwickelt werden soll, muss ein grundlegendes technologisches Verständnis vorhanden sein. Unternehmen sollten daher **in die Technologiekompetenz ihrer Mitarbeitenden investieren**. Technologie sollte nicht nur als Werkzeug, sondern vielmehr als Lösung gesehen werden. Wie bereits im Vorkapitel beschrieben empfiehlt es sich neben passenden Technologietrainings auch Mentorenprogramme und Communities aufzubauen.
Bei allen Vorteilen eines DIY Computing muss man akzeptieren, dass nicht alle Mitarbeitenden sich dafür interessieren oder geeignet sind. Menschen sind unterschiedlich: manche lassen sich für Technologie begeistern, andere nicht. Sie haben dafür andere Qualifikationen und Interessen, die es ebenso zu fördern gilt. Nicht jede Person will Citizen Developer werden und verfügt über die erforderlichen Interessen und Grundlagen!
- Wenn viele Citizen Developer entwickeln, können für ähnlich gelagerte Problemstellungen unterschiedliche sich teils widersprechende Lösungen entstehen. Um **Redundanzen zu vermeiden** und **Wiederverwendung zu fördern**, sind die Digitalanwender:innen der Zukunft gefordert. Es genügt nicht mehr, die kleine Applikation in einem No-Code-Tool aufzunehmen und „unter Verschluss in der stillen Kammer“ zu nutzen. Verantwortungsvolle Citizen Developer sind vielmehr gefragt, ihre Lösung für andere transparent und verfügbar zu machen. Dieses Bewusstsein muss geschaffen werden.
- Die **technischen Werkzeuge** bilden die notwendige Basis. Die Bereitstellung der passenden Plattformen liegt, wie bereits oben dargestellt, beim Digitalisierungspartner. Auf der Fachseite muss jedoch auch bekannt sein, welche Werkzeuge für unterschiedliche Zwecke zur Verfügung stehen, wo sie gefunden und wie sie verwendet werden können. Einerseits ist dies Teil der Qualifizierung und von Trainings. Es bedarf allerdings darüber hinaus auch einer entsprechenden **Wissenskultur auf der Anwender:innenseite**: gibt es bereits vergleichbare Lösungen für mein Problem? An wen kann ich mich wenden, wenn ich nicht weiterkomme? Bewege ich mich mit meiner Lösung im zulässigen Rahmen, z.B. in Fragen des Datenschutzes?
- Letztere Frage deutet in Richtung einer weiteren großen Herausforderung. Auf der Seite der Digitalanwender:innen bedarf es **eines organisatorischen Rahmens** und Regelwerks. In der klassischen IT selbstverständlich ist die Berücksichtigung von nicht fachlichen technischen Anforderungen wie z.B. Betriebsführbarkeit oder Lastverhalten, sowie von gesetzlichen und organisatorischen Erfordernissen wie z.B. Datenschutz und Gremienbeteiligung.
Insbesondere eine konsequente Datennutzung in neuen [datenbasierten Geschäftsmodellen](#) setzt **Cybersicherheit und die Einhaltung des Datenschutzes** voraus. Wird dem nicht entsprochen, drohen durch wenig reglementiertes DIY Computing [erhebliche Risiken](#). Ein schmaler Grat zwischen kreativen Freiräumen und verbindlichen Regeln, der gefunden werden muss.

Neue Formen der Zusammenarbeit und Geschäftsmodelle

Die Rollen von IT- und Fachabteilungen werden sich grundlegend verändern. **Grenzen werden verschwimmen.** Mit der zunehmenden Demokratisierung der IT kommen vermehrt Fragen auf, wem die IT und die darauf geschaffenen Applikationen gehören, wer Technologiestrategie und -leitplanken vorgibt. Es empfiehlt sich, dass Unternehmen sich proaktiv mit diesen Fragestellungen auseinandersetzen, um gestalten zu können, bevor absehbare Konflikte entstehen.

Sehr wahrscheinlich werden wir bei dem Zusammenspiel von Digitalisierungspartner und -anwender:innen eine evolutionäre Entwicklung erleben. Die Analysten von Forrester stellen in Ihrer Studie „Low-Code Citizen Developer Programs“ drei mögliche Zusammenarbeitsformen vor:

1. Der „**Privateer**“ ist der „Freibeuter“ in der Organisation. Dabei handelt es sich um kleine gemischte Teams, die auf der Geschäftsseite angesiedelt sind. Sie benutzen Low-Code-Umgebungen, um kleine bis komplexe Anwendungen für ihren fachlichen Bereich zu entwickeln. In diesem Modell gibt die IT-Abteilung Leitplanken für Technologie und Entwicklungsprozesse vor, nach denen sich die „Freibeuter-Teams“ richten. Bei diesem Modell können Kapazität und Skalierung aufgrund Kapselung und Größe der Teams schnell an Grenzen stoßen.
2. In der Demokratie („**Democracy**“) darf jeder Mitarbeitende in der Organisation als Citizen Developer aktiv werden. Dabei setzt er kleine Anwendungsfälle aus seinem geschäftlichen Umfeld entsprechend vorgegebener Richtlinien um. Unternehmen verfolgen laut Forrester damit das Ziel Tabellenkalkulationen oder Formularbearbeitungen zu ersetzen. Der Citizen Developer wird in diesem Modell von einem IT Kompetenzzentrum unterstützt, das Training und Support anbietet. Große und komplexe Anforderungen werden von diesem Modell nicht unterstützt.
3. In der Föderation („**Federation**“) unterstützt ein zentrales IT Kompetenzzentrum Citizen Developer Teams rund um sogenannte Abteilungschampions, die für bestimmte Produkte oder Kernprozesse Lösungen entwickeln. Das IT Kompetenzzentrum stellt dafür Plattformen und Richtlinien zur Verfügung, die je nach Anwendungsfall differenziert sein können.



Quelle: Forrester, Low-Code Citizen Developer Programs, John Bratincevic, August 3, 2021

Laut Forrester sollten Unternehmen mit dem „Privateer“-Modell starten, allerdings bereits die anderen beiden Modelle in Betracht ziehen, um mögliche Skalierungsgrenzen schnell überwinden zu können.

An diesen drei oder auch anderen denkbaren Modellen wird deutlich, dass sich die Aufgabenverteilung zwischen dem Digitalisierungspartner und -anwender:innen zukünftig anders gestaltet. Dies wirft **Fragen hinsichtlich der Geschäftsmodelle** auf. In der klassischen Welt gab es klar verteilte Auftraggeber- und Auftragnehmerrollen. Preise wurden auf Basis der zu erstellenden Anwendung ausgehandelt. Klassische Werk- oder Dienstverträge werden bei zunehmender IT-Demokratisierung nicht mehr ausreichen. Doch wie gehen wir in einer Welt, in der die Anwender:innen selbst entwickeln, damit um? Auf welcher Basis entwickeln sich Geschäftsmodelle und Preise?

In neuen Geschäftsmodellen ist zu klären, wie IT-Kompetenzzentren, moderne Citizen-Developer-Plattformen, neue organisatorische Erfordernisse wie Communities, Trainingsprogramme und vieles mehr finanziert werden können. Naheliegend als **Kalkulationsgrößen** sind Plattformnutzung und Ressourcenverbrauch. Doch dies allein wird nicht ausreichen. Hier bieten sich neue geschäftsorientierte Abrechnungsmodelle an. Diese erfordern messbare Geschäftsgrößen, die als Berechnungsgrößen zugrunde gelegt werden können.



employee per hour by department

PEOPLE
FORUMS
BUY
SALE

Distribution of market share among
the major industry players.

SEARCH
CONTACTS
MESSAGES

DIY Computing im Jahr 2031

DIY Computing im mobilen Einsatz	35
DIY Computing im (räumlich nicht festgelegten) Büro	36

SHOW BUSINESS

12010111010101001
091010 010100101
10101010010101

DIY Computing im Jahr 2031

Wir befinden uns im Jahr 2031. Die Weiterentwicklung und Nutzung von Applikationen haben sich weitgehend verändert. Dank der Unterstützung durch Künstlicher Intelligenz und intuitiver Werkzeuge nehmen wir die Gestaltung und Anpassung von Applikationen gar nicht mehr aktiv wahr. Vieles geschieht im Hintergrund. DIY Computing hat weniger mit aktiver visueller Programmierung zu tun. Vielmehr nutzen wir unsere natürlichen Kommunikationsmöglichkeiten, um digitale Lösungen zu gestalten. Im Folgenden wagen wir in zwei Zukunftsschichten, im Außendienst und am Büroarbeitsplatz, den Blick in die Kristallkugel.

DIY Computing im mobilen Einsatz

Auch wenn sich Visionen wie Flugtaxis bisher nur in Nischen durchsetzen konnten, hat sich die Mobilität in urbanen Gebieten stark weiterentwickelt. On Demand Mobilität hat mittlerweile zumindest in der Stadt das persönliche Auto verdrängt. Hoverboards konnten sich zwar nur eingeschränkt durchsetzen und der Elektro-Scooter hat aufgrund vermehrter Unfälle an Akzeptanz verloren. Nach wie vor ein beliebtes Fortbewegungsmittel ist das Fahrrad. Die Deutsche Bahn ist für sogenannte Leihräder mit Call A Bike immer noch der präferierte Anbieter. Im Unterschied zum Jahr 2021 sind die Fahrräder heute, im Jahr 2031, motorisiert und technisch komplexer.

Was sich nicht geändert hat: auch wenn die Robustheit aufgrund neuer Materialien zugenommen hat, gibt es immer noch Defekte. In unserer Geschichte meldet das Fahrrad kontinuierlich und in Echtzeit seinen Zustand. Anhand seines digitalen Zwillings ist sofort klar, dass eine Fernreparatur in diesem Fall auch im Jahr 2031 den Schaden nicht beheben kann. Da die Zweiradmechatronikerin Henriette sich gerade in der Nähe des defekten Fahrrads befindet, wird sie mit ihrem autonomen Reparaturmobil direkt zum schadhafte Fahrrad gebracht. Auch hat sie bereits die Meldung in ihre smarte Brille bekommen, dass es sich bei dem betroffenen Fahrzeug, um das neueste Modell des Herstellers „E-Esel“ handelt, von dem erst einige wenige Exemplare in der Flotte im Einsatz sind.

Henriette hat über den Digitalen Zwilling des Zweirads übermittelt bekommen, dass es sich um einen Defekt am Antrieb handelt. Sie beschreibt den Schaden und die Spracherkennung gibt den Schadensfall an die zentrale Auftrags- und Reparaturdatenbank. Leider gibt es für diesen Schaden an dem neuen Modell noch keine Anleitung oder ein vorausgefülltes digitales Auftragsdatenblatt. Das System sucht nach ähnlich gelagerten Schäden bei älteren Modellen des Herstellers und wird auch hier leider nicht fündig. Während der Einführungsphase für das neue Modell kann über eine Schnittstelle direkt auf das Supportsystem von „E-Esel“ zugegriffen werden. Dort findet sich auch tatsächlich eine passende Reparaturanleitung, die an das Headset von Henriette übertragen wird.

Da die Suche und das Auspielen Dank hoch performanter Systeme und Netze nur Bruchteile von Sekunden benötigte, hat Henriette die komplexen Bearbeitungsschritte und das Einbinden der neuen Dokumente in die Systeme von Call A Bike gar nicht mitbekommen. Gleichzeitig wird eine neue Auftragsart ohne weitere Aktivitäten unserer Zweiradmechatronikerin automatisch angelegt. Mittels Spracheingabe möchte Henriette den Auftrag im System quittieren. Dabei fällt ihr auf, dass sie über mehrere komplexe Dialoge geführt wird, um den Auftrag abzuschließen. Das müsste doch einfacher gehen! Jetzt muss Henriette doch noch ihr Smartboard in die Hand nehmen. Auf dem Gerät wird ihr der aktuelle Dialogfluss angezeigt. Über den Touchscreen fasst sie die sich wiederholenden Abfragen zusammen. Das System prüft im Hintergrund die Änderungen auf Plausibilität und ob keine wichtigen Informationen vergessen werden. Dem ist nicht so und Henriette kann den Auftrag einfach abschließen. Die von Henriette „programmierte“ Verbesserung steht sofort allen anderen im Unternehmen zur Verfügung.



Einige Wochen später zeigt sich, anhand des Schadenscodes der automatisch für dieses Fahrradmodell angelegt wurde, dass sich derartige Ausfälle bei den neuen „E-Esel“-Rädern häufen. Der Hersteller wird automatisch über die Schwachstelle informiert, die er bei zukünftigen Lieferungen beheben kann.

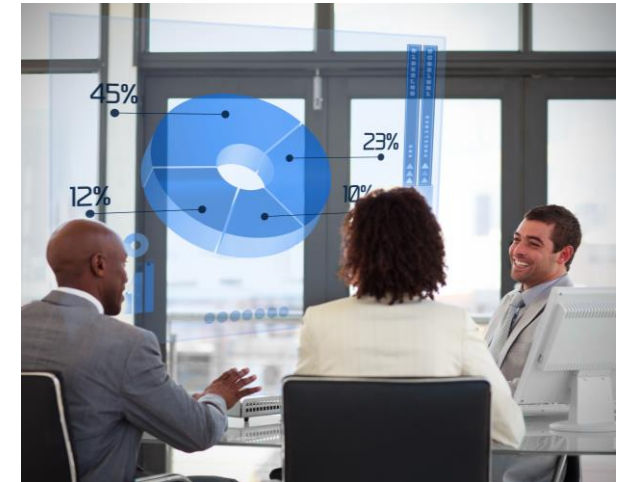
DIY Computing im (räumlich nicht festgelegten) Büro

Das morgendliche Briefing in der Personalabteilung findet wie üblich hybrid statt. Noch vor einigen Jahren war es üblich dazu ein Headset zu tragen und vor einem Bildschirm zu sitzen. In seinem Homeoffice hat sich Sebastian eine kleine Besprechungsecke eingerichtet. Für die tägliche Arbeit im Jahr 2031 ist sie mittlerweile wichtiger als der Schreibtisch mit dem Monitor, der mittlerweile etwas eingestaubt ist. Doch zurück zur Besprechungsecke: dort hat sich mittlerweile virtuell Sebastians Team eingefunden. Virtuell bedeutet, dass alle, repräsentiert durch dreidimensionale Hologramme, am Besprechungstisch sitzen, als ob sie physisch anwesend wären.

Im heutigen Meeting geht es darum, wie die neue Mitarbeitendenumfrage gestaltet werden kann. Da das Marktforschungsunternehmen „AnalyseDüse“ die operative Durchführung begleiten soll, sitzt das Hologramm der Geschäftsführerin mit am Tisch. In dem Meeting bekommt Sebastian den Auftrag die Softwareunterstützung von der Konzeption bis zur Kommunikation zu gestalten. Die Aufgaben werden schnell verteilt, so dass Sebastian direkt zur Tat schreiten kann. Wenn er derartige konzeptionelle Tätigkeiten hat, arbeitet er doch lieber an seinem Schreibtisch als in der Besprechungsecke.

Zunächst wirft Sebastian einen Blick auf die Software, die bei der letzten Umfrage eingesetzt wurde. Teile kann er sicherlich wiederverwenden. Er bekommt auf seinem Monitor den Prozessfluss mit allen Dialogfenstern angezeigt. Der multimediale Fragebogen von „AnalyseDüse“ wird direkt in den Workflow eingebunden. Die einzige Aktion die Sebastian am Computer durchführen muss, ist eine Bestätigung, dass er die Umfragedialoge einbinden will. Da bei einer Mitarbeitendenumfrage mit sehr sensiblen Daten gearbeitet wird, wirft Sebastian noch einmal einen kritischen Blick auf die grafische Darstellung der Arbeitsschritte und den Datenfluss im Hintergrund. Sebastian kann keine kritischen Sicherheitslücken entdecken. Ein Check durch einen künstlichen Softwareagenten bestätigt das nochmal. Die Ergebnisse der Prüfung werden auch direkt an die Arbeitnehmervertreter:innen übermittelt. Bis auf eine kleine Rückfrage, die sich schnell erledigen lässt, haben auch sie keine Bedenken.

Der Aufbau der Umfrage hat sich damit für Sebastian verhältnismäßig einfach gestaltet. Etwas mehr Zeit wird die Anpassung der Auswertungen benötigen. Bei der letzten Umfrage lag der Schwerpunkt auf Fragen rund um die Einführung der neuen Unternehmensstrategie. Jetzt wird es wieder eher um klassischen Themen wie Karriereentwicklung und Work-Life-Balance gehen. Da sich mit Hologramm-Meetings der Arbeitsalltag noch weiter als bisher ins Home-Office verschoben hat, ist auch diesem Umstand ein Teil der Abfrage gewidmet. Früher hatte sich Sebastian mithilfe von Datenexporten im CSV-Format und Auswertungen in Excel-Tabellen ausgeholfen. Doch seit die neuen Reporting-Werkzeuge im Einsatz sind, genügt es, dass er die gewünschten Visualisierungen am Bildschirm einfach zusammenstellt. Auch wenn sich das Reporting gegenüber dem letzten Mal stark verändert hat, ist Sebastian mit der Neugestaltung der holografischen dreidimensionalen Auswertungsgrafiken in nicht mal einer Stunde fertig. Für diese Tätigkeiten hätte Sebastian früher ein Vielfaches der Zeit benötigt. Schnelle „persönliche“ Meetings im virtuellen Raum und Software, die sich selbst konfiguriert und Änderungen intuitiv ermöglicht spart viel Zeit, die Sebastian für die konzeptionelle Planung einer neuen Recruiting-Kampagne gut gebrauchen kann.





Firmenportraits und Impulse

Anbieter von Citizen-Development-Werkzeugen	38
Große ERP Anbieter	40
Trendimpulse.....	41

SHOW BUSINESS

Anbieter von Citizen-Development-Werkzeugen

Aufgrund der unterschiedlichen Schwerpunktsetzung finden sich eine Vielzahl von Unternehmen in einem komplexen Marktumfeld. In diesem Abschnitt stellen wir Euch exemplarisch Anbieter von No-Code-, Low-Code und RPA-Plattformen vor. Die Auswahl und Reihenfolge beinhalten keine Wertung und keinen Anspruch auf Vollständigkeit. Hier soll lediglich ein erster Impuls gegeben werden, was heute bereits möglich ist und wohin die Reise gehen könnte.



Microsoft kann man als ein **Urgestein des Citizen Developments** bezeichnen. Fast täglich nutzen wir eines der Microsoft 365 Programme Excel, Word oder Powerpoint. Bereits sehr früh hat Microsoft in diesen Anwendungen einfache Programmiermöglichkeiten über die Programmiersprache „[↗ Visual Basic for Applications](#)“ geschaffen. Neben den Office Anwendungen ist Microsoft heute mit Teams einer der Marktführer für Kollaborations- und Kommunikationssoftware und damit **auf vielen PCs** vertreten. Bei diesen Entwicklungen ist es naheliegend, auch Nicht-Technologen zu ermöglichen, ihren digitalen Arbeitsplatz zu individualisieren und zu automatisieren.

Zu diesem Zweck stellt [↗ Microsoft die Power Plattform](#) zur Verfügung, mit der Workflows gestaltet und Informationen visualisiert werden können. Die Power Plattform setzt sich aus folgenden Komponenten zusammen:

- Power BI für die Visualisierung von Daten und Informationen,
- Power Apps zur Entwicklung eigener Anwendungen,
- Power Automate zur Automatisierung von Prozessen sowie
- Power Virtual Agents zur einfachen Erstellung von eigenen Chatbots.

Aufgrund der weiten Verbreitung von Microsoft auf Büro-PCs ist davon auszugehen, dass die Power Plattform eine wichtige Rolle bei der Demokratisierung der IT spielen wird. Auch bei der [↗ Deutschen Bahn kann die Power Plattform](#) genutzt werden.



Während der Schwerpunkt von Pega bei Low-Code-Entwicklung liegt, fokussiert das amerikanische Unternehmen [↗ UiPath](#) auf die **Automatisierung von Prozessen**, die durch **ERP- oder Office-Systeme** unterstützt werden. UiPath wurde 2005 in Rumänien gegründet und hat danach die typische Karriere eines „Digitalen [↗ Einhorns](#)“ vorzuweisen. Mit dem Wachstum hat das Unternehmen seinen Firmensitz nach New York verlegt und ist seit Anfang 2021 börsennotiert.

Auch durch Zukäufe stellt das Unternehmen eine breite Palette an Anwendungen zur Prozessautomatisierung zur Verfügung. Kern ist der UiPath Automation Hub über den der **gesamte Lebenszyklus** vom Erkennen automatisierbarer Prozesse bis hin zur Ausführung verwaltet wird. Wie in den anderen vorgestellten Produkten auch, ermöglicht UiPath über das Modul „StudioX“ Low-Code-Entwicklung.



Das börsengehandelte Unternehmen [↗ Pega](#) wurde bereits vor mehr als 30 Jahren in Cambridge, Massachusetts gegründet. Der Schwerpunkt des Unternehmens liegt in den Bereichen [↗ Business Process Management](#) und [↗ Customer Relation Management](#).

Mit der Pega Plattform wurden in der Vergangenheit bereits sehr **komplexe Geschäftsprozesse** realisiert. So setzen im deutschsprachigen Raum neben vielen Banken und Versicherungen auch große Konzerne wie die Deutsche Telekom und Siemens die Lösung ein. Ein wichtiges Kernelement ist ein Schichtensystem (situational Layer-Cake), das einen **hohen Grad an Wiederverwendbarkeit** inklusive einer Vererbungslogik bietet. Dadurch wird Entwicklungszeit eingespart und es werden Wartungsaufwände reduziert. Zudem hat Pega bereits früh auf eine Cloudstrategie gesetzt und den Einsatz von Künstlicher Intelligenz in seiner Software forciert.

Auf Basis dieser ausgereiften Funktionalität hat Pega die Plattform in den letzten Jahren sukzessive **um Low-Code-Ansätze erweitert**. Dadurch wird eine teamübergreifende Zusammenarbeit ermöglicht. Auch die Verwaltung und Skalierung von Low-Code-Apps wird unterstützt. Pega bewirbt seine Low-Code-Plattform unter anderem damit, dass fachliche Anwender:innen aktiv am Entwicklungsprozess teilnehmen können und damit in die Lage versetzt werden, in Echtzeit auf neue Geschäftsanforderungen zu reagieren. Das [↗ DB Systel Team flow.works](#) steht bei Fragen zur Nutzung der Pega-Plattform der Deutschen Bahn zur Verfügung.



Gegründet 2005, lange bevor der Begriff „Low Code“ etabliert war, kann [↗ Mendix](#) durchaus als „Veteran“ der Branche bezeichnet werden. Das amerikanische Unternehmen wurde 2018 von Siemens übernommen. Mendix hat bereits früh die einer **Low-Code-Entwicklung zugrunde liegenden Konzepte** verwendet. Das Unternehmen stellt Anwender:innen eine Entwicklungsumgebung zur Verfügung, die Anbindungen an **offene Standards** nutzt: [↗ Unified Modeling Language \(UML\)](#), [↗ Business Process Model and Notation \(BPMN\)](#) oder [↗ Open Data](#). Mendix bietet zwei Versionen an: „Mendix Studio“ als auch „Mendix Studio pro“. Richtet sich ersteres an fachfremde Personen, ist der Fokus von Mendix Studio Pro auf professionellen Entwickler:innen. In der Pro-Version wird der Programmierer durch den Mendix Assistent unterstützt. Dieser überprüft Logiken, antizipiert nächste Schritte und schlägt entsprechende Auswahloptionen vor oder deckt Fehler auf. [↗ Mendix arbeitet eng mit SAP](#) zusammen. Die Nutzung großer Datenmengen für IoT- und KI Anwendungen auf der [↗ SAP HANA](#) Plattform steht dabei im Fokus.



➤ **OutSystems** stellt seine **Low-Code Plattform** seit 2014 sowohl in einer kostenfreien als auch kostenpflichtigen Version bereit. Fokus der Software ist die Verkürzung der Entwicklungszeiten durch **visuelle Elemente** und die **Unterstützung durch KI**, die auf 33 Millionen anonymisierte Vorlagen und Abläufe zurückgreifen kann. OutSystems integriert mit visuellen Elementen die Benutzerschnittstelle, Logik- und Datenmodelle sowie Geschäftsprozess-Modellierungen. Bei der Veröffentlichung von Applikationen, die mit OutSystems entwickelt wurden, setzt die Plattform konsequent auf eine „ein-Klick“ Strategie. Fehleranalyse und Anwendungsoptimierung sind automatisiert. Abhängigkeiten werden automatisch mit der Funktion „True Change“ überprüft. OutSystems setzt strategisch auf die visuelle, KI-gestützte Erstellung von Applikationen durch Citizen Developer. Auf eine **klassische Programmiersprache wird verzichtet**. Daraus folgt, dass auch Pro-Developer keine Möglichkeit haben, auf den Code zuzugreifen, um diesen für komplexere Applikationen anzupassen.

Große ERP Anbieter



ORACLE

Unternehmen wie Pega, Mendix oder OutSystems erheben den Anspruch plattformübergreifend No-Code- und Low-Code Werkzeuge bereitzustellen. Wie groß der Trend hin zu der Befähigung von Citizen Developern ist, wird auch deutlich, wenn man einen Blick auf die **Marktführer von ERP Systemen** wirft. Auch wenn SAP bereits eine enge Partnerschaft mit Mendix verbindet, hat das [Walldorfer Softwarehaus Anfang 2021](#) zusätzlich den finnischen No-Code-Spezialisten [AppGyver](#) übernommen. Auch Salesforce bietet mit [Salesforce Flow](#) Möglichkeiten an, Low-Code-Applikationen zu entwickeln. Oracle spricht mit seinem Tool [APEX](#) zwar primär den Technologen an, ist damit jedoch auch im Low-Code-Umfeld tätig. Außerdem finden sich sowohl bei den großen ERP Anbietern als auch bei den spezialisierten Anbietern Werbeaussagen bzgl. der Unterstützung von Anwender:innen durch Künstliche Intelligenz. Neben dem Engagement der großen Softwarehäuser ist auch dies ein **Indiz zu einem starken Trend hin zu DIY Computing**.

Trendimpulse

Neben den Herstellern, die wir im vorherigen Kapitel kurz vorgestellt haben, gibt es eine Vielzahl kleinerer Startups und interessanter Anwendungsbeispiele. Auf dieser Seite stellen wir Ihnen exemplarisch einige Microtrends aus dem Trenduniversum von TRENDONE vor, die zeigen, wie Citizen-Development-Tools eingesetzt werden können.



Visuelle Darstellung von Geschäftsprozessen

Das Berliner Start-up [n8n](#) hat mit „n8n.cloud“ ein visuelles Workflow-Automatisierungstool entwickelt. Es basiert auf visueller Low-, bis No-Code-Bearbeitung und bietet damit ein leicht bedienbares Tool zur Modellierung anspruchsvoller Online-Geschäftsprozesse. Einzelne Arbeits- oder Synchronisierungsschritte können mithilfe grafischer Knoten miteinander verbunden und dadurch automatisiert werden. Daten zwischen verschiedenen Apps können hiermit synchronisiert, Produkte oder Dienstleistungen auf eine einfache und zeitsparende Art erstellt und Geschäftsprozesse automatisiert werden.



Low Code Plattform für Serviceroboter

Das Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA hat im Rahmen des Projektes [SeRo-Net](#) mittels xito.one eine offene IT-Plattform für Anwender:innen, Lösungsdienstleister, Robotik- und Komponentenhersteller für die Servicerobotik gelauncht. Mit der Plattform soll dabei ein baukastenähnliches Ökosystem für die Entwicklung von Robotersystemen geschaffen werden, in dem wiederverwendbare Komponenten für professionelle Servicerobotik-Anwendungen einfach zugänglich gemacht und darüber hinaus Orientierung gegeben werden, welche Anbieter und Komponenten am Markt verfügbar sind.



Interaktive No-Code-Plattform für Autoren

Das Start-up [Dorian](#) hat eine interaktive No-Code-Plattform entwickelt, die Autoren als Tool nutzen können, um sowohl ihre Geschichten in Games zu verwandeln als auch für deren Vertrieb. Autoren können ihre Charaktere vollständig anpassen und ohne Programmier- oder Kunstkenntnisse die Ergebnisse ihrer Arbeit sofort sehen. Auch können sie in Echtzeit in der Engine zusammenarbeiten und Feedback austauschen, bevor sie ihren Followern Inhalte zur Verfügung stellen. In einem Test mit Spielern konnten Autoren, die ihre datengesteuerten Vorschläge aufnahmen, die Monetarisierung um 70 Prozent verbessern.



Laufende Aktivitäten

Projekte und Maßnahmen bei der Deutschen Bahn..... 43

Projekte und Maßnahmen bei der Deutschen Bahn

Die systematische Auseinandersetzung mit Digitaltrends und deren Überführung in ein Prototyping sind entscheidend, um den wachsenden Herausforderungen der Bahn von morgen zu begegnen. Wie gewohnt stellen wir im Folgenden eine Auswahl der zahlreichen Aktivitäten aus unserem Konzern vor, die bereits heute den Weg in die DIY Computing Welt aufzeigen.

DIY Computing im DB Konzern befindet sich in der Anfangsphase. Ausgewählte Werkzeuge stehen dem Citizen Developer bereits heute zur Verfügung. Aktuell werden Citizen Development Tools hauptsächlich von professionellen Entwickler:innen genutzt. Zu verschiedenen Werkzeugen und Themen rund um DIY Computing haben sich erste Communities etabliert. Dies ist ein erster Schritt, um die Demokratisierung der Softwareentwicklung voranzutreiben. Einige dieser Initiativen und Aktivitäten stellen wir im Folgenden vor:

Die Low-Code-Community der Deutschen Bahn

Die Low Code Community bietet eine Austauschplattform für alle Themen rund um No-Code- und Low-Code-Entwicklung. Angesprochen werden Citizen Developer und Low-Code-Interessierte im DB Konzern. Die DB Planet Seite ist in erster Linie für Veröffentlichungen und organisatorische Informationen der Betreiber-Gruppe gedacht. Das Erstellen von neuen Beiträgen auf dieser Intranetseite ist nicht möglich. Wer sich aktiv in die Organisation einbringen möchte oder Ideen für Verbesserungen hat, kann gerne die [↗ Community Manager](#) ansprechen. Ergänzend für den aktiven Austausch, Nachrichten und Fragen bietet die [↗ gleichnamige MS Teams Gruppe](#) einen ersten Anlaufpunkt.

Die DB Power Plattform

Auf der DB Power Plattform können Ideen für Geschäftsanwendungen, die die Arbeit erleichtern, ohne ausgeprägte Programmierkenntnisse selbst entwickelt werden. Die [↗ DB Planet Seite](#) bietet dafür einen ersten Anlaufpunkt. Neben der Bereitstellung der Werkzeuge sind für Entwicklung und Nutzung von Apps auf der DB Power Plattform [↗ Genehmigungs- und Freigabeprozesse](#) verbindlich geregelt. Auch hinsichtlich der Nutzung können sich Interessierte in einem [↗ MS Teams Kanal](#) austauschen.

Die Pega-Plattform der Deutschen Bahn

Das Team [↗ flow.works](#) bietet mit „[↗ Pega](#)“ eine Workflow-Management Lösung an, die es über eine intuitive Benutzeroberfläche erlaubt, Geschäftsprozesse zu digitalisieren. Citizen Developer, die diese Plattform nutzen, haben die Möglichkeit sich in der [↗ Pega Citizen Developer Community](#) auszutauschen.

Automatisierung von Geschäftsprozessen

Robotic Process Automation mit „[↗ UI Path](#)“ wird durch [↗ RPA Solutions](#) angeboten. In dieser Plattform integriert ist eine Low-Code-Umgebung, die IT-affine Mitarbeitende der Deutschen Bahn befähigt, selbst die Digitalisierung von Prozessen umzusetzen. Neuigkeiten zu Robotic Process Automation finden sich in der gleichnamigen [↗ Community](#).

Weitere Aktivitäten in Geschäftsfeldern und Tochtergesellschaften

Bei der DB Kommunikationstechnik wird Low Code Entwicklung unterstützt. Fachmitarbeitenden wird im Rahmen eines [Citizen Developer Programms](#) die Möglichkeit eröffnet, auf einer Microsoft Power Plattform eigene Applikationen zu entwickeln.

Die DB System hat gemeinsam mit dem CIO Bereich des Konzerns ein [Whitepaper „Citizen Developer“](#) herausgegeben. In der Publikation werden grundlegende Begrifflichkeiten geklärt und es wird aufgezeigt, wie sich die IT-Landschaft durch Fachbereichsentwickler:innen in Zukunft verändert.

Das Team der regionalen Instandsetzung Südwest der DB Netz lässt vom Team [Low-Code Development and Consulting der DB System](#) in Kooperation mit dem CIO Bereich der Konzernleitung eine App entwickeln, um einen komplexen, manuellen Beauftragungsprozess auf Basis der Power-Plattform in der RIS-App zu automatisieren.

Das Team [Low-Code Development and Consulting der DB System](#) hat mit der KVP-App (Kontinuierlicher Verbesserungsprozess) einen PoC in Kooperation mit FINANCE 4 DB und dem SSC Buchhaltung erfolgreich entwickelt, um eine bestehende Excel-Lösung perspektivisch abzulösen. Die Umsetzung als Power-App befähigt Mitarbeitende, Verbesserungsvorschläge besser zu verwalten.

Etwa 20.000 DB Kolleg:innen nutzen Tableau, Qlik, Power BI und SAP BusinessObjects als Self-Service Business Intelligence Lösungen, die das [Team Command Your Business Data as a Service](#) bereitstellt.

Im Rahmen des Projekts [Production IT as a Service](#) wird den Service Designern der DB System die Möglichkeit geboten, selbst ihre Workflows zur Leistungserbringung auf der Pega-Plattform zu modellieren.



Anhang

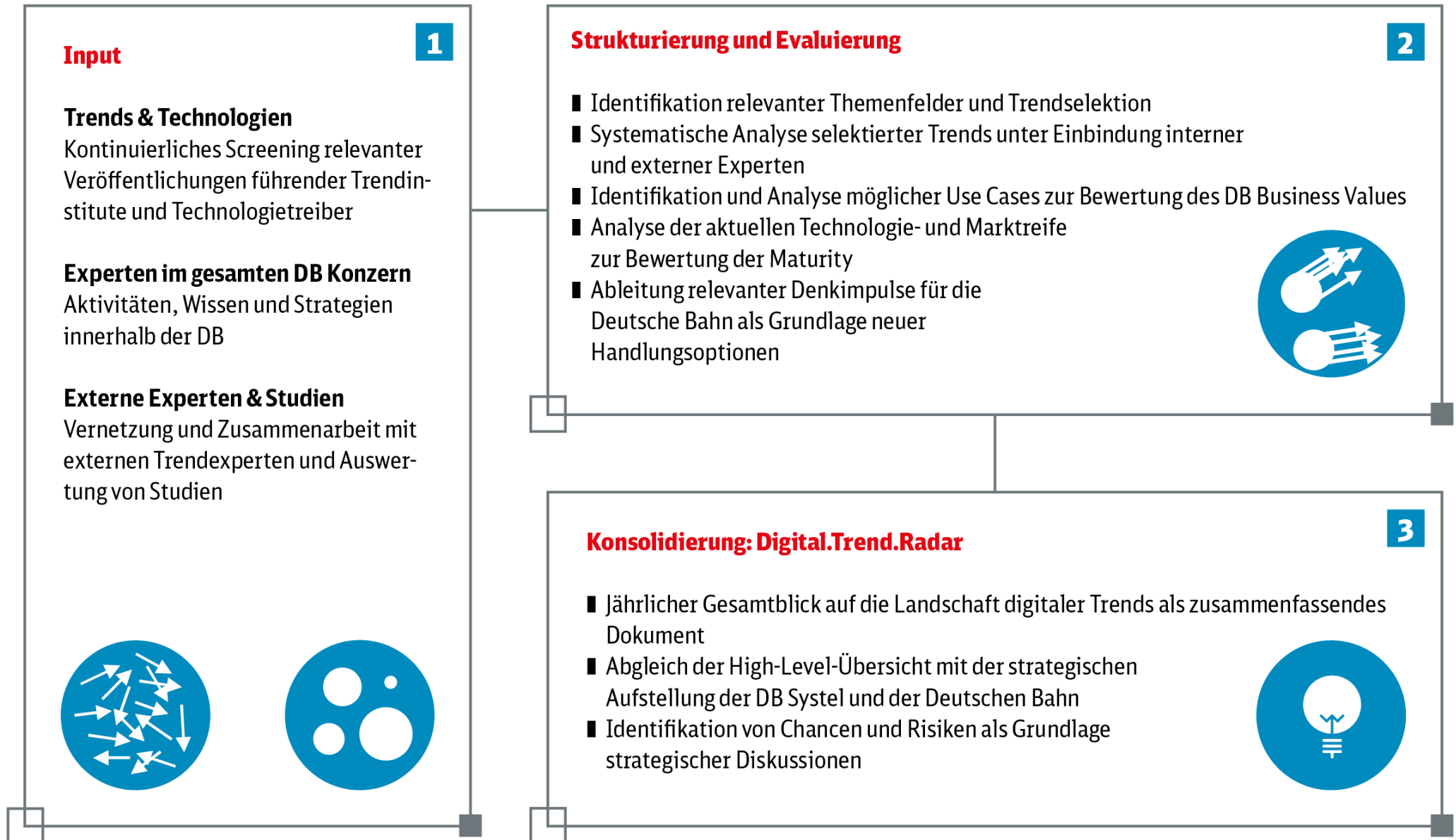
Glossar und Abkürzungen.....	46
Vorgehensweise & Methodik	47
Trendbewertung	48
Ansprechpartner:innen	49

Glossar und Abkürzungen

- **Agile Softwareentwicklung**
- **BPMN** Business Process Model and Notation
- **Business Process Management**
- **Citizen Developer**
- **Citizen Integrator**
- **Customer Relationship Management**
- **DIY** Do It Yourself
- **ERP** Enterprise Resource Planning
- **IT-Demand-Supply**
- **Hyperautomation**
- **Low-Code-Plattform**
- **Machine Learning bzw. Maschinelles Lernen**
- **Open Data**
- **Open Source**
- **Pair Programming**
- **Reinforcement Learning**
- **RPA** Robotic Process Automation
- **Schatten-IT**
- **Selbstintegrierende Applikationen**
- **UML** Unified Modeling Language

Vorgehensweise & Methodik

Durch unsere strukturierte Vorgehensweise bleibt die DB im Fokus



Trendbewertung

Die Trendbewertungen basieren auf systematischen Ableitungen anhand eines objektiven Kriterienkatalogs. Mittels einer transparenten und nachvollziehbaren Systematik wird ein Konsens in der Einschätzung der betrachteten Trends erzeugt. Das einheitliche Zukunftsbild ermöglicht eine konstruktive Diskussion zukünftiger Chancen und Risiken. Neue strategische Optionen werden identifiziert und finden Eingang in die Strategieentwicklung.

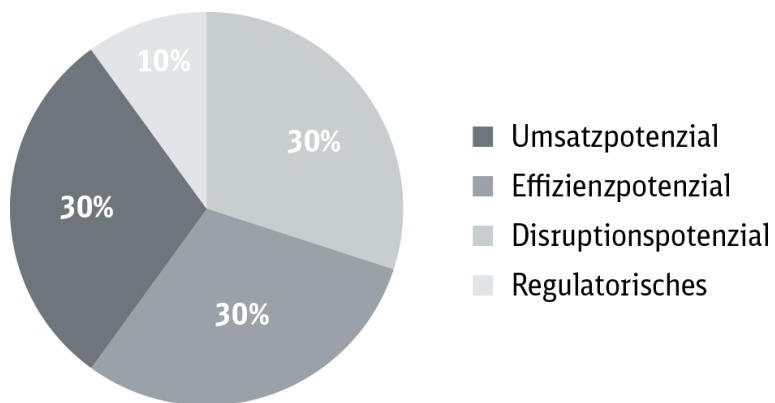
Bewertungskriterien

DB Business Value

Der DB Business Value ist ein Maß für die Relevanz eines Trends für den DB Konzern auf Sicht von zehn Jahren. Dazu werden Chancen und Risiken zukünftiger Anwendungsszenarien bewertet.

Der DB Business Value beantwortet die Frage:

Wie wichtig ist es für die Deutsche Bahn, sich mit dem Trend auseinanderzusetzen?

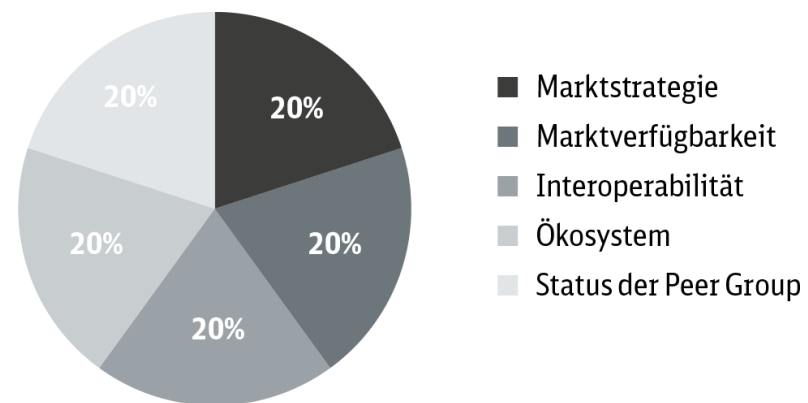


Maturity

Die Maturity beschreibt die heutige Reife eines Trends. Neben dem technischen Entwicklungsstand spielt die dazugehörige Markt- und Leistungsinfrastruktur eine wesentliche Rolle.

Die Maturity beantwortet die Frage:

Wie leicht ist es heute bereits, die aus dem Trend resultierenden Möglichkeiten zu nutzen?



Ansprechpartner:innen

Neben dem hier aufgeführten „Kernteam“ haben außerdem zahlreiche weitere Kolleg:innen sowie externe Partnerorganisationen an dieser Studie mitgewirkt, ihre Expertise eingebracht, Impulse gegeben und unterstützt. An dieser Stelle bedanken wir uns nochmal ganz herzlich für den wertvollen Input und das konstruktive Feedback. Selbst ein kurzes Gespräch kann neue Perspektiven aufzeigen und inspirieren.

Autor:innen der Trendstudie

Team Digital Foresight

Christian Kolarsch | [↗ E-Mail](#) | [↗ Chat](#)

Christine Mohn | [↗ E-Mail](#) | [↗ Chat](#)

Ádám Lászlop | [↗ E-Mail](#) | [↗ Chat](#)

Weitere Kollegen aus der DB System

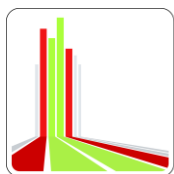
Dirk Böning-Cortierer | [↗ E-Mail](#) | [↗ Chat](#)

Johannes Gerner | [↗ E-Mail](#) | [↗ Chat](#)

Robert Kacer | [↗ E-Mail](#) | [↗ Chat](#)

Stefan Gerberding | [↗ E-Mail](#) | [↗ Chat](#)

Sie haben Fragen oder möchten mit uns zum Thema „DIY Computing“ diskutieren? Dann besuchen Sie unsere [↗ MS Teams Gruppe](#) „Digital Foresight Trend Community“ oder schreiben uns eine E-Mail an digital.foresight@deutschebahn.com – Wir freuen uns auf Sie!



**DIGITALE TRENDS
& INNOVATIONEN**

Für eine starke Schiene.