DB Systel
Digital bewegen.
Gemeinsam.

Digital.Trend.Impuls

Softwareentwicklung 2030

Trends und Auswirkungen für die DB

DB Systel GmbH | Digital Foresight | November 2022

Softwareentwicklung 2030

Anhang

Softwareentwicklung 2030: Eine Grundlage der Digitalisierung

Die Digitalisierung ist ein entscheidender Hebel, die Deutsche Bahn für die Anforderungen der Zukunft fit zu machen und den Erwartungen von Politik und Gesellschaft gerecht zu werden. Digitalisierung funktioniert nur mit massivem Einsatz von IT-Systemen und führt zu einer zunehmenden Komplexität der IT-Architekturen. Dabei wird uns der akute Mangel an IT-Fachkräften noch viele Jahre begleiten.

Ohne Software gibt es keine Digitalisierung. Zunehmend rückt daher die Softwareentwicklung selbst in den Fokus. Neue Technologien und Vorgehensweisen werden zukünftigen Entwickler:innen zur Seite stehen, sie bei ihren Aufgaben unterstützen und Teilumfänge automatisieren. Die Integration vorhandener Software wird immer mehr Bedeutung erlangen. Es werden aber auch neue Anforderungen aufkommen. Um ein klareres Bild der Veränderungen zu gewinnen, haben wir fünf Veränderungsbereiche identifiziert, die wir Ihnen auf den nächsten Seiten vorstellen wollen.

Nicht alle Chancen liegen dabei allein in der IT. Durch ein immer engeres Zusammenrücken von IT und Business wird immer mehr IT-Kompetenz in die Business-Bereiche gelangen, mit positiven Auswirkungen auf Qualität und Geschwindigkeit.

All dies bedeutet in den kommenden Jahren große Veränderungen für die Arbeit der Softwareentwickler:innen. Wie werden sich die Arbeitsumgebungen weiterentwickeln? Wie werden dabei Plattformen das Ökosystem erweitern? Zu welchem Anteil wird uns KI die Arbeit abnehmen können? Vieles erscheint aus unserer heutigen Sicht kaum vorstellbar. Dennoch müssen wir uns frühzeitig mit den Veränderungen der Softwareentwicklung beschäftigen, auch um unsere jetzigen und zukünftigen Kolleg:innen in der IT gezielt darauf vorzubereiten. Nur so werden wir die anstehenden Aufgaben der Digitalisierung in der Geschwindigkeit und Qualität erfüllen können, die unsere Kunden, die Reisenden und Transporteure, zu Recht von uns erwarten.

Der vorliegende Digital.Trend.Impuls „Softwareentwicklung 2030“ wirft ein Schlaglicht auf das Thema und zeigt Chancen auf, die sich uns eröffnen. Gleichzeitig möchte er Zuversicht verbreiten, dass wir die anstehenden Herausforderungen gemeinsam stemmen können.

Gerald Hofer | Geschäftsführer Operations DB Systel GmbH

Softwareentwicklung – Kontinuität oder Disruption?

Konrad Zuse hat mit seiner Entwicklung Z3 im Jahre 1941 die erste vollautomatisierte, frei programmierbare Rechenmaschine vorgeführt. Dem waren bereits Konzepte mechanischer Rechenmaschinen und über Lochkarten programmierbare Webstühle vorausgegangen. Dies war der technologische Startpunkt der Softwareentwicklung, wie wir sie prinzipiell bis heute kennen, und die uns auch in der weiteren Zukunft intensiv beschäftigen wird. Denn insbesondere die immer schneller voranschreitende Digitalisierung ist ohne die Entwicklung von Software undenkbar – und erfordert auch die Weiterentwicklung des Erstellens von Software.

Eine Idee von Software ist, die zugrundeliegenden Hardwaretechnologien, die sich ebenfalls kontinuierlich weiterentwickeln, flexibel, transparent und optimiert zu nutzen. Diese Anforderung besteht bis heute unverändert, und erst recht in der Zukunft, angesichts immer komplexer werdender Strukturen verteilter Systeme. Hier kommen die Softwareentwickler:innen ins Spiel, die einerseits fachliche Anforderungen umsetzen sollen, und sich andererseits mit zugrundeliegenden IT-Architekturen auseinandersetzen müssen. Dies alles erfolgt vor dem Hintergrund immer kürzer werdenden Entwicklungszyklen.

> **„In der schnelllebigen IT-Branche ist der Fachkräftemangel das Einzige, was man für die nächsten 10 Jahre ziemlich sicher voraussagen kann“**
>
> Stefan Schmeißer | Implementation Lead | DB Vertrieb GmbH

Vor dem Hintergrund des Fachkräftemangels, der gerade auch die IT betrifft, muss Softwareentwicklung von allen Ineffizienzen entlastet werden, die Softwareentwickler:innen vom Kern ihrer Aufgabe, der IT-Umsetzung von „Fachlichkeit“, abhalten. Zudem müssen die Umgebungen dafür sorgen, dass der Programmcode eine immer höhere Qualität besitzt, um Fehler zu vermeiden. Gleichzeitig wird von den Entwickler:innen erwartet, dass sie kontinuierlich lernen und sich den Veränderungen stellen.

Dazu öffnet sich ein weites Feld von Tools und Hilfsmitteln über ↗ Containerisierung bis hin zu der Vision, dass künstliche Intelligenz auch in diesem Bereich massiv unterstützen kann. Doch wie genau wird Softwareentwicklung in fünf bis zehn Jahren aussehen, woraus wird die hauptsächliche Arbeit von Entwickler:innen bestehen? Werden KI-Systeme die Software der Zukunft eigenständig entwickeln? Oder bleibt Software ein Bereich, in dem der Mensch auch auf absehbare Zeit unverzichtbar sein wird?

Zu diesen Fragen will dieser Digital.Trend.Impuls Ausblicke, Bewertungen und Anregungen geben.

Uwe Friedrichsen, was bringt die Zukunft der Softwareentwicklung?

Uwe Friedrichsen ist CTO der codecentric AG, Herausgeber der IT-Fachzeitschriften Softwerker und OBJEKTspektrum sowie Autor, Berater und Speaker auf internationalen Konferenzen. Als Reisender in der komplexen Welt der IT ist er stets am Puls der Zeit und weiß, welche Themen die IT von (über-) morgen bewegen.

Uwe, in deinen Key Notes rufst Du dein Publikum dazu auf, IT „anders zu denken". Was meinst Du damit?

Viele Unternehmen denken IT immer noch als Cost Center, das schlicht auf Produktionseffizienz zu optimieren ist und bauen darauf ihre Prozesse und Organisation auf. Die fortschreitende digitale Transformation bedeutet aber im Kern, dass Business und IT untrennbar geworden sind: Business ist IT und IT ist Business. Das bedeutet: Steuere ich IT als Cost Center, steuere ich mein Business als Cost Center – was in einer immer dynamischeren und unberechenbareren Welt (Stichwort: ↗ VUCA) keine Option mehr ist.

Mal losgelöst vom üblichen „Buzzword Bingo" – welche Themen bewegen die Softwareentwicklung 2030?

Ich denke, Resilienz und (nicht nur ökologische) Nachhaltigkeit werden in den kommenden Jahren sehr wichtige Themen in der IT werden. IT ist ähnlich wie Strom unverzichtbar geworden, aber unsere Systemlandschaften taumeln aufgrund jahrzehntelangen Cost-Center-Denkens am Rande der Unwartbarkeit und Unbetreibbarkeit. Da müssen wir dringend ran. Zusätzlich wird uns der demographische Wandel und der damit verbundene Fachkräftemangel bewegen. Auf Seiten der Technologie denke ich, dass sowohl die Public Cloud als auch das Zusammenwachsen von IT und OT die Softwareentwicklung stark beeinflussen werden.

Eines deiner Herzensthemen ist „Resilienz", wie designen wir robuste und zukunftsfähige Systeme?

Wenn man diese Frage nur in zwei oder drei Sätzen beantworten könnte! Das ist ein sehr großes Thema. Vielleicht nur kurz zwei der Kernprobleme, denen ich häufig begegne: Wenn ich ↗ Dev und Ops strikt voneinander trenne, dann funktioniert es nicht, weil wir enge Feedbackschleifen zwischen Dev und Ops benötigen. Und wenn ich immer nur auf maximale Effizienz schiele, dann wird es auch nichts, denn für gute Resilienz muss ich ein ganz klein wenig Effizienz aufgeben.

„You built it, you run it" – Wird mit DevOps alles gut?

Das kommt darauf an, was man erreichen will. Im Kern ging es zumindest ursprünglich bei DevOps darum, die Durchlaufzeiten durch die IT-Wertschöpfungskette systematisch zu verkürzen: Wie schnell bekomme ich eine neue Business-Idee hinaus zu den Kunden und kann von deren Feedback lernen. Das hilft mir, mich erfolgreicher in dynamischen Märkten bewegen zu können. Das bedeutet aber auch, dass man IT „anders denken" muss (siehe erste Frage). Und damit tun sich viele Unternehmen noch schwer.

So alt wie Machine Learning und künstliche Intelligenz selbst, ist die Angst vor der Herrschaft der Maschinen. Machen hochentwickelte Sprachverarbeitungsmodelle wie GPT-3 Programmierer bald arbeitslos?

Dazu gibt es viele komplementäre Meinungen. Meine Meinung dazu ist: Wir müssen in der Softwareentwicklung zwischen Routinetätigkeiten und den kreativen Tätigkeiten unterscheiden. Nicht alle Entwicklungsaufgaben sind kreative Akte. Viele Teile sind einfach Fleißarbeit. An den Stellen kann KI sicherlich Aufgaben übernehmen, was den Fachkräftemangel vielleicht etwas entschärfen könnte. Bei den kreativen Anteilen der Softwareentwicklung sehe ich das nicht.

Die Zukunft der Softwareentwicklung spielt sich auf verschiedenen Ebenen ab

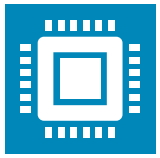
Richtet man den Blick auf die Zukunft der Softwareentwicklung, entdeckt man mehrere Ebenen, auf denen sich diese abspielen wird.

Naheliegend ist es, zunächst in der Kategorie der **Werkzeuge** zu denken, die Softwareentwickler:innen tagtäglich nutzen. Es gibt Tools, die bei der Produktion des Codes unterstützen, der die fachlichen Prozesse abbildet. Andere Hilfsmittel dienen dazu, das Produkt, letztlich ausführbare Programmteile, in der geforderten Qualität dorthin zu bringen, wo sie betrieben werden. Hier ergeben sich Möglichkeiten der Effizienzsteigerung, die angesichts der auf Jahre hinaus angespannten Personalsituation unverzichtbar sind.

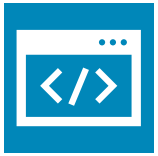
Zudem unterstützen verstärkt Instrumente der **künstlichen Intelligenz** die tägliche Arbeit. Im einfachsten Fall werden Entwickler:innen situativ Codefragmente zur Verwendung vorgeschlagen. Zukünftig ist vorstellbar, dass gerade standardisierbare Bestandteile einer Anwendung automatisch ergänzt werden. Dies setzt aber voraus, dass die künstliche Intelligenz zielführend angelernt wurde.

Der Erfolg von Softwareentwicklung hängt maßgeblich von der **Organisation** ab, in der sie erfolgt.
Wie sind die Teams aufgeteilt, welche Aufgaben haben sie, wie sehen die Schnittstellen untereinander aus?
DevOps und agiles Arbeiten sind richtungsweisend, aber wie entwickelt sich die Landschaft weiter?

Der Fachkräftemangel in der IT ruft nach effizienterer Softwareentwicklung. Ein wichtiges Konzept hierbei ist die **Wiederverwendung** von Programmteilen, sei es innerhalb eines Unternehmens oder -verbundes, oder ganz frei, nach dem Open Source-Gedanken.



Und nicht zuletzt sind es sogenannte „weiche“ Faktoren, die für die Akzeptanz entwickelter Software bei den Kunden und Nutzern immer wichtiger werden. Software muss den Kriterien ↗ **Digitaler Ethik** genügen, **barrierefrei** sein und sich zudem dem Prinzip der ↗ **Green IT** verpflichten. Dies sollte bereits beim Entstehen von Software weitestgehend automatisiert unterstützt werden, um den Entwickler:innen mehr Zeit für die Umsetzung der fachlichen Anforderungen zu lassen.



Impuls 1: Werkzeuge der Softwareentwicklung

Beschreibung

Mitte des vorigen Jahrhunderts entstanden die ersten maschinell nutzbaren Programmiersprachen, Sprachen sind seitdem ein Inbegriff der Programmierung. Zu den Werkzeugen zur Softwareentwicklung gehören aber u.a. auch Systeme zur Verwaltung von Quellcode, zum automatisierten Testen, genauso wie Code-Bibliotheken zur einfachen Nutzung von Komponenten unterschiedlicher Anwendungsarchitekturen.

Zudem wandelt sich Softwareentwicklung dank Entwicklungsplattformen und unter Betriebsaspekten immer mehr in Richtung Softwareintegration und IT-Engineering. Hierbei spielen wiederverwendbare und kombinierbare Services eine große Rolle.

Im Laufe der Zeit wächst die Menge bestehender Software ständig weiter. Parallel gibt es stetig neue Programmiersprachen und Entwicklungswerkzeuge. Daher werden die Softwaremodule langfristig nachhaltig wartbar entwickelt werden müssen.

Ausblick

Die Technologien rund um die Softwareentwicklung und -integration verändern sich stetig, eine exakte Voraussage zur „state of the art“ 2030 ist entsprechend schwierig. Zudem muss das Bestehende kontinuierlich erneuert werden, auch um etablierte Softwareentwicklungs-Mechanismen zumindest teilweise in technologisch neue Umgebungen zu überführen. Damit wird den Entwickler:innen kein vollständiges Umlernen zugemutet, was sie verstärkt von ihrer produktiven Tätigkeit abhalten würde.

Zunehmend verteilte Systeme und darüber ausgebreitete Prozesse verkomplizieren die Softwareentwicklung immer weiter. All dies kann von einem Einzelnen nicht mehr überblickt werden. Entwickler:innen sollten ihre Expertise zuerst auf das Verstehen und Umsetzen der fachlichen Anforderungen verwenden können, insbesondere angesichts des weiterhin bestehenden Mangels an IT-Fachkräften. Daher werden Systeme wichtiger, die die unterliegende Komplexität „verstecken“ und über einfache Schnittstellen plattformübergreifende Entwicklung fachlicher Software ermöglichen.

> **„Erfolgreiche Projekte sind die, bei denen die Entwickler verstanden haben, was die Anwender benötigen“**
>
> **Thorsten Gau** | CTO IBM Consulting

All dies gelingt nur mit adäquater Toolunterstützung. Im Folgenden sind beispielhaft einige Aspekte der Weiterentwicklung von Entwicklungswerkzeugen aufgeführt, die auch den Betrieb von Anwendungen betreffen können:

↗ **Performance Engineering** setzt mit einem umfassenden systematischen Ansatz in jeder Phase der Softwareentwicklung an, um die Anforderungen u.a. an Stabilität, Verfügbarkeit, und Skalierbarkeit effizient zu erfüllen. Die Integration weiterer Anforderungen wie z.B. aus Green IT oder Digitaler Ethik sind vorstellbar.

↗ **Observability** beschreibt die Fähigkeit, die internen Zustände eines Systems zu messen, indem man seine Ausgabewerte untersucht. Dieses Vorgehen wird zunehmend im Zusammenhang mit der Performance-Verbesserung ↗ verteilter IT-Systeme verwendet. Es dient zudem dazu, Fehlerzustände zu erkennen und diesen abzuhelfen. Dadurch können die Ursachen einer Vielzahl von Problemen ermittelt sowie die Leistung eines IT-Systems verbessert werden.

Das Konzept des ↗ **Chaos Engineering** ist ein „Stresstest für Anwendungen“, bei dem diese mit scheinbar beliebigen Maßnahmen auf Fehler und Schwächen analysiert werden. Anders als der Begriff suggeriert, wird dies jedoch geplant und strukturiert durchgeführt. Chaos Engineering wurde auch schon bei DB Systel verprobt. Das im nachfolgenden Abschnitt dargestellte „Best Practice“-Beispiel zeigt den erfolgreichen Einsatz von Chaos Engineering bei DB Vertrieb.

Einschätzung und Chancen für die DB

- Es werden kontinuierlich neue Methoden und Verfahren entwickelt, um Softwareentwicklung oder Teile daraus sowie den Betrieb von Anwendungen zu optimieren. Dies betrifft die gesamte Entwicklungs-Pipeline vom Design über Programmierung und Test bis zum Deployment.
- Neue Methoden und Verfahren sollten jeweils in Piloten evaluiert werden, um mögliche Effizienzgewinne zu verifizieren.
- Effizienzen können u.a. bei Performance, Zeitdauer und Entwicklereinsatz entstehen. All dies hilft, dem Fachkräftemangel in der IT zu begegnen.

Best Practice: Chaos Engineering bei DB Vertrieb

DB Vertrieb

Immer komplexere Geschäftsprozesse, gestiegene Benutzeranforderungen und sich immer schneller ändernde Umgebungen erhöhen die Komplexität unserer IT-Systeme massiv. Um auch in hochkomplexen, verteilten Systemen den Überblick zu behalten und Risiken zu minimieren, hat DB Vertrieb vor 3 Jahren erfolgreich „Chaos Engineering“ in den Entwickleralltag eingeführt. Was dahinter steckt und welche Vorteile Chaos Engineering zusätzlich zu herkömmlichen Qualitätstests bietet, lest ihr in unserer „Best Practice“.

Der Schmetterlingseffekt. Komplexe, dynamische Systeme verhalten sich per se nicht linear. Hochverfügbare, verteilte IT-Systeme auf Basis von Microservices und die agile Arbeitsorganisation in interdisziplinären, selbstorganisierten Teams schaffen eine soziotechnisch hochkomplexe Umgebung. Kommunikation und Abstimmungsprozesse werden immer herausfordernder und der Blick für das Gesamtsystem verschwimmt. Die Folge kennen wir aus der ↗ Chaostheorie: Kleine, vermeintlich unbedeutende Fehler und Störungen schaukeln sich zu unvorhersehbaren, systemweiten Problemen hoch oder führen gar zu Totalausfällen. Die Zusammenhänge zwischen Ursache und Wirkung können oft erst im Nachhinein entschlüsselt und verstanden werden oder bleiben gar völlig in Verborgenheit. Dies geht nicht nur auf Kosten der Performance, sondern letztendlich auch der Sicherheit und Robustheit der IT-Systeme.

Was kann schon schief gehen? Beim Release des ersten großen Funktionalitäten der neuen Vertriebsplattform kam es „pünktlich zum Wochenende“ zu einem Totalausfall. Eine unglückliche Kombination von Konfigurationsfehlern und Softwarebugs hatte sich zu einer systemweiten Störung hochgeschaukelt, deren Ursachen so komplex waren, dass erst Tage später alle Details identifiziert und analysiert werden konnten. Die riesige Testabdeckung aus automatisierten Unit-Tests, Integrationstest, Ende-zu-Ende-Test, manuellen explorativen Test, technischen Abnahmetests und mehr hatten dieses Szenario nicht vorhersagen und den dadurch ausgelösten Totalausfall nicht verhindern können. Denn die Ursachen lagen nicht in den exakt vorhersehbaren, reproduzierbaren Abläufen der einzelnen Komponenten, sondern folgten sprichwörtlich dem „Chaos-Prinzip“ eines komplexen verteilten Systems. In einem solchen Umfeld stoßen etablierte Qualitätssicherungsmechanismen an ihre Grenzen und es entsteht eine Lücke, die verheerende Folgen mit sich bringen kann.

Warum Chaos Engineering anders ist. Um dieser Unbeherrschbarkeit entgegenzuwirken, entschlossen sich die Kolleg:innen von DB Vertrieb, Chaos Engineering in ihren Entwicklungsprozessen zu verproben. Im Gegensatz zu klassischen Tests, bei denen Umfang und Ergebnis im Vorhinein feststehen, besitzt Chaos Engineering einen experimentellen Charakter. Grundsätzlich steht am Anfang eine Hypothese oder Vermutung über die Auswirkungen des Tests. Wie sich das System oder der Prozess tatsächlich während des Experiments verhalten wird, gilt es erst herauszufinden. Indem gezielt Fehler in die Systeme „injiziert“ werden, können Schwachstellen der IT- und Organisationssysteme aufgedeckt werden. Das Systemverhalten wird währenddessen genau beobachtet und die Erkenntnisse dokumentiert. Wie bei Brandschutzübungen der Feuerwehr, werden beim Chaos Engineering in so

„Letztendlich ist es viel kostengünstiger, durch den Einsatz von Chaos Engineering die Robustheit und Stabilität unserer IT-Systeme zu stärken, als Fehler erst im ‚Post Mortem‘ zu identifizieren, also dann, wenn der Worst Case bereits eingetroffen ist“

Jonas vor dem Berge
Implementation Lead Vertriebsplattform | DB Vertrieb

genannten „Game Days“ gezielt „Brände gelegt“ und beobachtet, wie sich die Systeme verhalten und welche Maßnahmen ergriffen werden müssen, um diese wieder zu löschen.

Fazit und Ausblick. Durch die Einführung von Chaos Engineering konnten die Kolleg:innen von DB Vertrieb ihren Entwicklern eine effiziente Methode an die Hand geben, den gestiegenen Anforderungen komplexer Systemlandschaften zu begegnen, Risiken zu minieren und sich auf den Ernstfall vorzubereiten. Viele Teams führen mittlerweile selbstorganisiert die Game Days durch. Dennoch bleibt es eine Herausforderung, Chaos Engineering in den Entwickleralltag zu integrieren und auch die ↗ Product Owner und Fachseiten zu überzeugen. Denn bislang über 170 durchgeführte Game Days bedeuten auch eine nicht unwesentliche Investition. Doch dem gegenüber stehen 260 entdeckte Erkenntnisse, darunter viele ernstzunehmende Lücken, die über die herkömmlichen Testverfahren bislang nicht identifiziert wurden.

Zudem lassen sich auch Chaosexperimente automatisieren. Hierzu hat sich innerhalb der letzten Jahre bereits eine kleine Landschaft an Chaos Engineering Tools gebildet. Mittlerweile gibt es für jede Technologie die passenden Tools, sei es auf der Ebene Virtual Machine, Docker Container/Kubernetes, JVM oder NodeJS. ↗ Chaos Monkey ist ein von Netflix entwickeltes Tool und beendet virtuelle Maschineninstanzen und Container, die in der Produktionsumgebung ausgeführt werden. Wie ein kleines Äffchen, dass man an den Rechner setzt, um Chaos zu stiften. Neben dem Chaos-Äffchen bietet Netflix mittlerweile weitere Tools wie die ↗ Simian Army oder die ↗ Chaos Automation Platform an. ↗ Pumba ist ein Chaos-Testing-Tool für Docker Container. Wie Pumba und Chaos Monkey bei DB Vertrieb zum Einsatz kommen, präsentieren Oliver Kracht und Jonas vor dem Berge von DB Vertrieb in einem ↗ Vortrag auf der SoftwerkerKonf 2020 von codecentric.

Einschätzung und Chancen für die DB

Chaos Engineering reduziert die Risiken und deckt Schwächen sowohl im technischen Bereich als auch in der sozialen Organisation und im Miteinander von fachlicher und technischer Seite in interdisziplinären und übergreifenden Teams. Aufgrund dieser Mehrwerte empfiehlt sich ein breiterer Einsatz bei der DB.

Weitere Informationen zum Thema „Chaos Engineering“

- ↗ **Principles of Chaos Engineering**
- ↗ **„Awesome Chaos Engineering“** auf GitHub
- ↗ Informationen auf dem **Netflix TechBlog**
- ↗ DB Systel: **Interview mit Russ Miles**

Ihr möchtet auch organisiertes Chaos stiften? Dann wendet Euch gerne an die Kollegen von DB Vertrieb:

Jonas vor dem Berge
Implementation Lead Vertriebsplattform
DB Vertrieb GmbH

Stefan Schmeißer
Implementation Lead Vertriebsplattform
DB Vertrieb GmbH

Impuls 2:
Künstliche Intelligenz in der Softwareentwicklung – Effizienz durch Automatisierung

Beschreibung

↗ Künstliche Intelligenz (KI) ist der Oberbegriff für die maschinelle Nachbildung menschenähnlicher Intelligenz, mit dem Ziel der eigenständigen Bearbeitung komplexer Probleme. Hierzu gehört die Fähigkeit eines Systems, Ereignisse und Daten zu analysieren und aus diesen Informationen zu lernen, um eigenständig bestimmte Ziele und Aufgaben zu erreichen.

In der Softwareentwicklungen gibt es sehr hohe Erwartungen an KI. In der Tat scheinen strukturierte Vorgänge wie das Programmieren einer Geschäftsanwendung prädestiniert für die einer KI zugrunde liegenden Mechanismen, wie z.B. ↗ Machine Learning.

> **„KI wird sich bei der Code-Erzeugung um die alltäglichen Aufgaben kümmern: Einfache Teile des Codes werden von KI bereitgestellt“**
>
> **Dave Wright** | Chief Innovation Officer, ServiceNow

Ausblick

Trotz aller Fortschritte im Bereich der KI werden auch in zehn Jahren Entwickler von Software nicht überflüssig. KI wird bis dahin nicht in der Lage sein, aus einer komplexen fachlichen Spezifikation einen fertigen Programmcode erzeugen. Dennoch werden KI-basierte Werkzeuge und Verfahren großen Einfluss auf die Softwareentwicklung nehmen, um diese zu Beschleunigen und die Qualität der Ergebnisse zu steigern. Dabei werden die Entwickler:innen situativ unterstützt: **AI-Augmented Software Engineering** (AIASE) greift in die bestehende Werkzeugarchitektur und „berät“ die Entwickler:innen, insbesondere bei komplexen, plattformbezogenen Problemstellungen. **Predictive Code Completion**, das Vorschlagen von Codefragmenten und Subroutinen, ist ein Bestandteil davon. Hierbei können sich aber Lizenzfragen stellen, zudem sollte die Qualität des vorgeschlagenen Codes gesichert sein.

> **„Machine Learning kann zu ‚predictive costs‘ und ‚predictive sustainability’ gegenüber den Hyperscalern führen“**
>
> **Markus Eisele** | Developer Strategy, RedHat

Testautomatisierung ist bereits heute Standard, ↗ **Autonomous Testing** wird aber in der Zukunft weit darüber hinaus gehen, indem es – angelernt von vielen realen Testszenarien und sich ständig optimierend – vollständig ohne menschliche Eingriffe auskommt. Dabei werden Software- und Systemtests nicht nur automatisch ausgeführt, sondern auch konzipiert, geplant und die Ergebnisse analysiert.

Ein weiterer Bereich der KI in der Softwareentwicklung betrifft **Konzeption und Steuerung des Betriebs der unterliegenden Infrastruktur**. Hierbei wird die optimierte Umgebung über die Zeit erlernt. Dies führt zu besserer Kostenplanung sowie -einsparungen bei der Nutzung dynamischer Cloudressourcen, zudem ist der ökologische Fußabdruck der Plattformen optimierbar.

Machine Learning wirkt aber auch indirekt auf die Softwareentwicklung: Diese wird in geeigneten Szenarien vereinfacht, da nicht alle Fachlichkeit von den Entwickler:innen programmiert werden muss, sondern die Anwendung sich selbstlernend optimiert.

Dabei muss beachtet werden, dass ein erfolgreicher Einsatz von KI und Machine Learning ausreichende Trainingsdaten zum Anlernen voraussetzt. Dies könnte sich als verzögernd beim produktiven Einsatz erweisen. Die Erzeugung synthetischer Daten könnte ein Lösungsweg sein. Andere Techniken wie das **Visual Language Model (VLM)** erlauben das parallele Erlernen über Bilder, Videos und Text. Ein Beispiel ist ↗ Deepmind Flamingo.

In der ↗ Digital.Trend.Studie „Do It Yourself Computing" werden weitere Beispiele für die Verwendung von KI in der Softwareentwicklung aufgezeigt.

Weitere große Entwicklungsschritte werden sich mittelfristig durch hybride Berechnungen auf ↗ **High Performance Computern** und ↗ **Quantencomputern** ergeben. Dabei wird zunächst entschieden werden müssen, welcher Anteil der Problemstellung je nach Problemklasse auf klassischer oder auf Quantenarchitektur umgesetzt wird. Bei Entwicklungen für Quantencomputer werden neben klassischer Programmierung neue Fähigkeiten benötigt, wie z.B. Grundkenntnisse der Quantenphysik und der Linearen Algebra. Es entsteht das Berufsbild des Quanteninformatikers als Symbiose aus Informatik, Physik und Mathematik, erste Studiengänge werden dahingehend bereits entwickelt.

Einschätzung und Chancen für die DB

- Künstliche Intelligenz und Machine Learning bieten zunehmend Werkzeuge, die als Assistenzen dabei helfen können, bestimmte Tätigkeiten der Softwareentwicklung durchzuführen.
- Einsatzfelder sind das monotone Erstellen von sogenanntem ↗ Glue Code oder das Kodieren von Entwurfsmustern auf der Basis von Beispielen. (Glue Code ist Programmcode, der keinerlei Funktionalität zum Erreichen der Programmziele beiträgt, sondern ausschließlich dem „Zusammenkleben" verschiedener Teile des Programmcodes dient, die sonst nicht kompatibel wären.)
- Das vollautomatische Kodieren einer Anwendung mit gegebener Semantik wird in absehbarer Zeit sehr wahrscheinlich nicht realisiert. Es wird verstärkt Unterstützung bei der Softwareentwicklung geben, aber keine Revolution.

Impuls 3: Organisation der Softwareentwicklung

Beschreibung

Die Organisation der Softwareentwicklung hat sich stets auf unterschiedlichen Ebenen weiterentwickelt. Die klassische Entwicklung nach Wasserfallmodell geht zu agilen Formen über. Projekte, in denen große Anwendungen vollständig entwickelt wurden, werden von der Integration flexibler Services abgelöst. Geschäft und IT wachsen bei der Entwicklung zunehmend zusammen, und die Rolle des ↗ **Citizen Developers** tritt immer stärker auf den Plan.

Das Design der Organisation der Softwareentwicklung ist neben den eingesetzten technischen Tools der entscheidende Faktor für Effektivität und Effizienz. Dies betrifft u.a. die Aufteilung der Aufgaben auf Teams und die organisatorischen Schnittstellen untereinander. DevOps erscheint uns heute als das für die Digitalisierung favorisierte Modell, da es die Geschwindigkeit der Softwareentwicklung erhöht und somit die „Time to Market" verbessert. Es überall einsetzbar, wo die Rahmenbedingungen vorhanden sind oder entsprechend gestaltet werden können. Dies wird sich in den nächsten Jahren weiterentwickeln, auch um mit der wachsenden Komplexität standzuhalten.

Parallel wird durch die Verlagerung der Entwicklung digitaler Lösungen in Fachteams unter Nutzung von ↗ **No Code-** und ↗ **Low Code-Plattformen** die Möglichkeit eröffnet, einfache Problemstellungen ohne fundierte Softwareentwicklungskenntnisse umzusetzen.

↗ **Business-IT-Fusion** ist ein Organisationsmodell, in dem Geschäft und IT gemeinsam und kundenzentriert digitale Produkte entwickeln. Dies ist ein ganzheitlicher Ansatz bei der DB, um Geschäft und IT zusammenwachsen zu lassen. Mit ↗ **BizDevOps** wird eine Auflösung getrennter Organisationstrukturen verfolgt und die Verantwortung für Fachlichkeit und technische Umsetzung in einem Team gemeinschaftlich wahrgenommen.

Ausblick

Anwendungsentwicklung erfordert immer mehr Fähigkeiten, über die Umsetzung der reinen fachlichen Anforderungen hinaus. Gleichzeitig sollen sich die Entwickler:innen aber genau darauf konzentrieren, business- und kundenorientiert zu arbeiten. Wie beschrieben wird dies einerseits durch Tools unterstützt, zudem wird aber auch mehr Fachwissen durch Experten in die Teams gelangen müssen. Für DevOps wird diese crossfunktionale Weiterentwicklung durch „DevXxOps" beschrieben. So bedeutet beispielweise ↗ **DevSecOps** die Integration und Erfüllung von Security- und Compliance-Anforderungen in die Teams und Umgebungen, möglichst automatisiert und transparent. Dies kann geeignete Rollentrennungen erfordern, innerhalb von Teams wie über Teams hinweg.

Beispiele prominenter Unternehmen wie Netflix oder Spotify, die versuchen, die gesamte Spanne von „Dev" bis „Ops" in einem Team abzubilden, werden wir an unsere Bedarfe beispielsweise hinsichtlich der Skalierung in Breite und Tiefe anpassen müssen.

Konkret kann dies den Wechsel von horizontalen zu mehr vertikalen, spezialisierteren Teams zur Folge haben. Dies kann aber auch einen Neuschnitt von Themen innerhalb eines Teams bedeuten, um das erforderliche Know-How zu begrenzen, indem z.B. große Systeme in Services zerlegt werden. Technische Services werden dann durch Plattformen bereitgestellt und im Self-Service genutzt.

„Low Code ist ein weiteres Werkzeug in unserem Werkzeugkoffer“

Bernd Drothen | Verantwortlicher Solution Engineering Deutschland, Salesforce

Der Einsatz von **Entwicklungsplattformen**, die den Anwendungen bestimmte Services verfügbar machen, bedeutet einen erheblichen Gewinn an Effektivität und Effizienz. Plattformen werden kontinuierlich weiterentwickelt, wovon alle darauf basierenden Entwicklungen profitieren. Aufgrund ihrer offenen Architektur skalieren Plattformen besser als individuelle, anwendungsspezifische Entwicklungen. Zusammen mit weiteren Vorteilen sind Plattformen zudem auch deutlich kostengünstiger als Einzelentwicklungen.

Einen erweiterten organisatorischen Ansatz für Entwicklung und Betrieb von Softwaresystemen bietet ↗ **„Team Topologies“**, ein schrittweises und anpassungsfähiges Modell für Organisationsdesign und Teaminteraktion. Grundlage sind vier Teamtypen, die über drei sogenannte Transaktionsmodi verbunden werden.

Viele der Digitalberufe 2032 werden aus Tätigkeiten bestehen, die vergleichbar mit den heutigen Aufgaben und Bedürfnissen sind. Doch durch die neuen gesellschaftlichen und technologischen Rahmenbedingungen werden bestehende Tätigkeitsprofile aufgebrochen oder in neue übergreifende Tätigkeitsfelder gebündelt. Die notwendigen Fachkompetenzen verändern oder erweitern sich. Eine übergreifende vertiefende **Studie „Digitalkompetenzen 2032“** ist dazu in Vorbereitung.

Do It Yourself Computing wird einen Teil der unteren Seite des Anwendungsspektrums abdecken. Dabei werden fachliche Expert:innen befähigt, auf ihre Anforderungen passende Lösungen selbst zu erstellen. Dies wird allerdings erst umfassend durch Bereitstellung sowie einen sicheren und zuverlässigen Betrieb von leistungsfähigen IT-Plattformen und Services möglich. ↗ **No-Code-** und ↗ **Low-Code-Plattformen** stellen dazu intuitive und grafische Benutzeroberflächen zur Verfügung. Zum DIY Computing hat Digital Foresight von DB Systel 2021 ↗ eine detaillierte Studie herausgegeben.

Einschätzung und Chancen für die DB

- Softwareentwicklungs-Kompetenz ist nötiger denn je, von anspruchsvoller professioneller Entwicklung komplexer Systeme bis zur Umsetzung von einfachen Lösungen durch interessierte Laien. Nur so kann die nötige Entwicklungs-Kapazität erreicht werden.
- Deshalb sollten entsprechende Kompetenzen gefördert sowie Training und Ausbildung in diesem Bereich systematisch eingesetzt werden, und die Möglichkeiten zur autonomen und durch Eigenmotivation getriebenen Weiterbildung unterstützt werden.
- Das kann die Vielfalt von leicht oder frei zugänglichem Material und Lernmöglichkeiten umfassen, sowie auch das Engagement mit eigener Arbeit in offenen Projekten.
- Zur schnelleren Umsetzung der Anforderungen sollte in agilen Teams gearbeitet und bevorzugt DevOps als Modell eingesetzt werden. Dies verkürzt die Entwicklungszyklen und ermöglicht, „Wert“ schneller zu realisieren.
- DevOps und BizDevOps muss geschäftsfeldübergreifend funktionieren, da die Digitalisierung nicht an den Grenzen der Geschäftsfelder Halt macht.
- Agilität ist ohne Transparenz nicht denkbar. Die Förderung von Entwicklungskompetenz profitiert auch enorm dadurch, dass Teams ihr Entscheidungen und Erfahrungen transparent machen und so Eigenschaften offenlegen und den Wissensaustausch fördern.
- Ganzheitliche Ansätze zum noch engeren Zusammenwachsen von Business und IT ermöglichen es, dass IT sich nicht mehr nur als Dienstleister der Fachbereiche versteht, sondern als Wegbereiter des digitalen Business.

Impuls 4: Wiederverwendbarkeit und Open Source

Beschreibung

↗ Open Source ist allgegenwärtig in der IT (↗ 78% des Codes ist Open Source). Auch der Anteil industriespezifischer Open Source-Software nimmt zu. Vorteile von Open Source sind einerseits erhöhte Unabhängigkeit und Interoperabilität, weil alle Unternehmen und Nutzer offenen Zugang haben. Zudem sorgt Open Source für hohe Effizienz, weil die Kosten für gemeinsame Teile des Software-Stacks geteilt werden. Hinzu tritt der Aspekt der beschleunigten Innovation durch das Mitwirken vieler Entwickler am Code.

Für große Software-Hersteller wie z.B. ↗ Microsoft oder ↗ SAP ist das Entwickeln von Open Source-Code selbstverständlich, zehntausende Entwickler schreiben in den großen Unternehmen Open Source.

Ausblick

Open Source ist ein wichtiger Ansatz, um dem Fachkräftemangel zu begegnen. Dadurch, dass vorhandener Code, der nach dem Community-Prinzip zumeist in hoher Qualität vorliegt, genutzt wird, muss dieser nicht mehr von eigenen Entwickler:innen programmiert werden.

Bei Vorbehalten gegen die Nutzung „öffentlicher" Software in bestimmten unternehmenskritischen Anwendungen kann das ↗ **Inner Source-Prinzip** Abhilfe schaffen. Dabei wird Programmcode frei innerhalb eines Unternehmens, eines Konzernverbunds oder einer anders definierten Nutzergruppe geteilt. Dies erfordert aber die Schaffung der kulturellen Voraussetzungen, eine entsprechende „Sharing-Kultur" muss im Unternehmen begründet werden. Dies kann beispielsweise durch eine Community vorangetrieben werden, in der die Befürworter einen Rahmen und Regeln festlegen, der die Entwickler:innen motiviert, daran teilzunehmen. Ziel ist auch, Entwicklern Vertrauen zu „fremden" Code zu geben, anstatt sich nur mit Eigenentwicklung auf der sicheren Seite zu fühlen.

„2030 wird jeder Entwickler Open Source-Software schreiben"
Cornelius Schumacher, Open Source Steward im Chief Technology Office DB Systel

Allerdings bedeutet dies, dass Entwickler mehr die Verantwortung zur Auswahl des richtigen Codes tragen. Es wird wichtiger werden, vorhandenen Code zu analysieren anstatt neu zu schreiben. Entwickler benötigen daher die Fähigkeit, durch den Open Source-Dschungel zu navigieren.

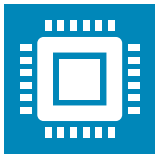
Der Open Source-Ansatz wird sich zukünftig auch auf das weitere Umfeld der Softwareentwicklung ausdehnen, wie z.B. Design oder Dokumentation.

Im Jahr 2021 wurde eine Initiative gegründet, um eine Foundation für Open Source-Zusammenarbeit im Bahn-Sektor zu gründen, die ↗ **OpenRail Foundation**. Neben der dem internationalen Eisenbahn-Verbund ↗ UIC sind hier vor allem die Eisenbahngesellschaften aus Frankreich (↗ SNCF), der Schweiz (↗ SBB) und die DB beteiligt. Weitere Bahnen und Forschungseinrichtungen haben schon ihr Interesse bekundet teilzunehmen. Auf der Mitgliederversammlung der UIC Ende letzten Jahres wurde die Initiative vorgestellt. Mehr Informationen finden sich auf der ↗ OpenRail Homepage.

Eine weitere Open Source-Initiative im Logistikbereich ist die ↗ **Open Logistics Foundation**, die von DB Schenker mit gegründet wurde. Als Herzstück wird dort die Plattform Open Logistics Repository geplant, auf der sämtliche Open Source-Software kostenlos und uneingeschränkt zur Verfügung stehen soll, um eine breite Akzeptanz zu fördern.

Einschätzung und Chancen für die DB

- Open Source-Einsatz ist in der Entwicklung allgegenwärtig. Ein souveräner und kompetenter Umgang mit Open Source-Software erhöht die Effizienz der Softwareentwicklung.
- Beteiligungen an Open Source-Projekten finden in einigen Fällen statt, oft wird aber das Potenzial des Engagements in Open Source-Communities noch nicht genutzt.
- Inner Source wird als Konzept anerkannt, auch vom CIO Board der DB. Für die Lizenzen existieren bereits Vertragsvorlagen. In einzelnen Projekten wird dies bereits genutzt, die Werkzeuge wurden erfolgreich eingesetzt. Aber hier besteht ein großes Potenzial, mehr Zusammenarbeit zu verwirklichen und Silos aufzulösen. Neues sollte als Inner Source realisiert werden.
- Die DB hat beim Open Source-Engagement Nachholbedarf. Dies wird oftmals als „zusätzliche Arbeit“ betrachtet und ist heute noch kein integraler Bestandteil der Entwicklungsarbeit.
- Zur Förderung der Softwareentwicklungs-Kompetenz gehören insbesondere auch Möglichkeiten und Angebote aus den internen und externen Communities.



Impuls 5: Corporate Digital Responsibility

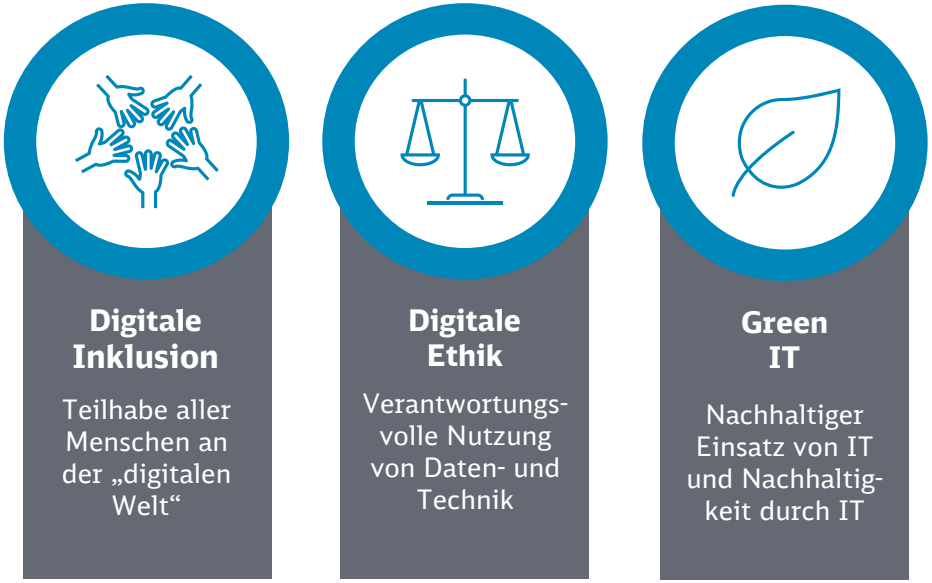
Beschreibung

Mit dem fortschreitenden Einzug von Technologien wie KI, dem Internet der Dinge oder Mixed Reality in unserem persönlichen und beruflichen Alltag rücken auch die gesellschaftlichen und sozialen Perspektiven zunehmend in den Vordergrund. Diskriminierende Algorithmen und Deep Fakes fordern unser Werte- und Rechtssystem heraus, rechenintensive Datenverarbeitungen in riesigen Server-Farmen sorgen für keinen unerheblichen CO2-Ausstoß und Teile der Gesellschaft werden durch die rasanten Entwicklungen möglicherweise abgehängt.

Um langfristig erfolgreich zu sein, darf ein Unternehmen also nicht nur die technologischen oder wirtschaftlichen Aspekte im Blick behalten, sondern muss Transparenz und Vertrauen bei Mitarbeitenden, Kund:innen, öffentlichem Sektor und Stakeholdern schaffen. Dabei gilt der Grundsatz, dass Technologie dem Menschen dienen muss und nicht umgekehrt. Innerhalb der letzten Jahre verstärkt sich zunehmend der Handlungsdruck durch Gesetze und Richtlinien. Corporate Digital Responsibility lässt sich in drei Handlungsfelder unterteilen:

- **Digitale Inklusion:** Mit der Allgegenwärtigkeit digitaler Medien erstreckt sich auch die Anforderung nach Inklusion auf die digitale Welt. Für die knapp 8 Millionen Menschen in Deutschland mit Schwerbehinderung ist Barrierefreiheit nicht nur in physischer Form (wie bspw. Rampen für Rollstuhlfahrer:innen) eine Grundvoraussetzung für ein selbstbestimmtes Leben. Barrierefreiheit bedeutet auch, dass alle Menschen digitale Medien und Services nutzen können. Das bedeutet, dass sich digitale Dienste in ihrer Form und Funktionsweise den individuellen Bedürfnissen aller Menschen anpassen. Die ↗ EU-Richtlinien 2019/882 oder auch „European Accessibility Act“ fordert die Umsetzung von digitaler Barrierefreiheit für Unternehmen, ansonsten drohen erhebliche Sanktionen. Für die Softwareentwicklung bedeutet dies, Standards für Schnittstellen einzuhalten, damit Hilfsmittel wie z.B. Screen Reader für seheingeschränkte Personen und Blinde einwandfrei ausgeführt werden können.

- ↗ **Digitale Ethik:** Die Grenzen zwischen digitaler und physischer Welt verschwimmen zunehmend. Insbesondere die rasante Entwicklung von künstlicher Intelligenz und zunehmender Automatisierung führt zu ethischen Anforderungen, wie z.B. dem Schutz der Privatsphäre, diskriminierungsfreie Interpretation von Daten oder die Nachvollziehbarkeit von (KI-)Entscheidungen. Digitale Ethik beschäftigt sich ↗ „mit moralischen Fragen des digitalen Wandels“. Sie stellt die Frage nach dem „guten und richtigen“ (Zusammen-) Leben in einer Welt, die von digitalen Technologien geprägt ist und formuliert entsprechende Grundregeln. Der Grundgedanke lautet: Das Verhältnis zwischen Mensch und Maschine sollte in einer freiheitlichen Gesellschaft nicht nur durch das technologisch Machbare, sondern auch ↗ durch das moralisch Wünschenswerte bestimmt werden.
- ↗ **Green IT:** Eine beschleunigte Digitalisierung kann ↗ bis zu 41% der erforderlichen CO2e-Einsparungen realisieren, um die deutschen Klimaziele 2030 zu erreichen. Doch mehr Digitalisierung führt nicht automatisch zu mehr Nachhaltigkeit. Die IT hat schon jetzt einen Anteil von bis zu 4% bei den weltweiten Treibhausgas-Emissionen. Digitalisierung und Nachhaltigkeit müssen also zusammen gedacht werden.



Einschätzung und Chancen für die DB

- Digitale Inklusion, Digitale Ethik und Green IT sind grundlegende Aspekte, deren Berücksichtigung bereits heute bei Anwendungen erwartet wird, sowohl von der Gesellschaft als auch konkret vom Gesetzgeber. Die Softwareentwicklung hat daher eine wichtige und strategische Rolle bei der Entwicklung und Umsetzung ethischer Richtlinien im Rahmen der digitalen Transformation.
- Um zukunftssicher zu sein, müssen Software und Architektur so entwickelt werden, dass sie flexibel auf sich ändernde Gesetzesanforderungen, veränderte Kundenanforderungen oder neue Erkenntnisse aus Unternehmenssicht reagieren können. Eine nachträgliche Implementierung wird stets deutlich aufwändiger.
- Entwickler:innen müssen für die ethischen und moralischen Aspekte ihrer Arbeit sensibilisiert werden, z.B. durch geeignete Schulungsmaßnahmen.
- Anforderungen an Software, die sich aus der Corporate Digital Responsibility ergeben, könnten zukünftig vermehrt durch Softwareentwicklungswerkzeuge automatisiert in neue Software eingebaut werden.

Anhang

Interessante Quellen

Artikel, Reports und Videos

- ↗ Autonomous Testing
- ↗ BizDevOps
- ↗ Business-IT-Fusion
- ↗ Citizen Developer
- ↗ Containerisierung
- ↗ Deepmind Flamingo
- ↗ Dev und Ops
- ↗ DevSecOps
- ↗ Digitale Ethik
- ↗ Digital.Trend.Studie „Digitale Ethik in einer vernetzten Welt“
- ↗ Digital.Trend.Studie „Do It Yourself Computing”
- ↗ European Accessibility Act
- ↗ Chaostheorie
- ↗ Chaos Engineering
- ↗ Principles of Chaos Engineering

Chaos Engineering bei Netflix:

- ↗ Chaos Engineering
- ↗ Chaos Monkey
- ↗ Simian Army
- ↗ Chaos Automation Platform
- ↗ Chaos-Testing-Tool Pumba
- ↗ Chaos-Testing-Tools Pumba und Chaos Monkey bei DB Vertrieb
- ↗ GitHub: Awesome Chaos Engineering
- ↗ Interview mit Russ Miles, CEO von ChaosIQ
- ↗ Green IT
- ↗ High Performance Computing
- ↗ Inner Source
- ↗ Klimaeffekte der Digitalisierung (bitkom)
- ↗ Künstliche Intelligenz (KI)
- ↗ Low Code-Plattformen
- ↗ Machine Learning
- ↗ No Code-Plattformen
- ↗ Observability
- ↗ Open Source
- ↗ Open Source Security and Risk (OSSRA) report 2022
- ↗ Microsoft and open source
- ↗ SAP und Open Source: Passt das zusammen? (E-3 Magazin)
- ↗ Open Logistics Foundation
- ↗ OpenRail Foundation
- ↗ Performance Engineering
- ↗ Quantencomputing
- ↗ Team Topologies
- ↗ Verteilte IT-Systeme

Vorgehensweise & Methodik

Durch unsere strukturierte Vorgehensweise bleibt die DB im Fokus

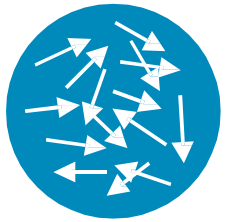
Input

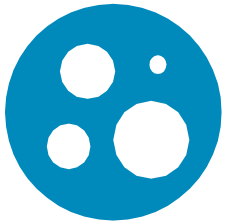
1

Trends & Technologien
Kontinuierliches Screening relevanter Veröffentlichungen führender Trendinstitute und Technologietreiber:innen

Expert:innen im gesamten DB Konzern
Aktivitäten, Wissen und Strategien innerhalb der DB

Externe Expert:innen & Studien
Vernetzung und Zusammenarbeit mit externen Trendexpert:innen und Auswertung von Studien

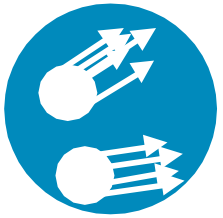




Strukturierung und Evaluierung

2

- Identifikation relevanter Themenfelder und Trendselektion
- Systematische Analyse selektierter Trends unter Einbindung interner und externer Expert:innen
- Identifikation und Analyse möglicher Use Cases zur Bewertung des DB Business Values
- Analyse der aktuellen Technologie- und Marktreife zur Bewertung der Maturity
- Ableitung relevanter Denkimpulse für die Deutsche Bahn als Grundlage neuer Handlungsoptionen



Konsolidierung: Digital.Trend.Radar

3

- Jährlicher Gesamtblick auf die Landschaft digitaler Trends als zusammenfassendes Dokument
- Abgleich der High-Level-Übersicht mit der strategischen Aufstellung der DB Systel und der Deutschen Bahn
- Identifikation von Chancen und Risiken als Grundlage strategischer Diskussionen

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 herzlich für den wertvollen Input und das konstruktive Feedback. Selbst ein kurzes Gespräch kann neue Perspektiven aufzeigen und inspirieren.

Autoren des Trendimpulses

Christian von Knobloch | ↗ E-Mail | ↗ Chat

Christine Mohn | ↗ E-Mail | ↗ Chat

mit Unterstützung von

DB Systel	**Gualter Baptista**	↗ E-Mail | ↗ Chat
	Stefan Gerberding	↗ E-Mail | ↗ Chat
	Manfred Rieck	↗ E-Mail | ↗ Chat
	Cornelius Schumacher	↗ E-Mail | ↗ Chat
	Ulrich Vogler	↗ E-Mail | ↗ Chat
	Daniel Woithe	↗ E-Mail | ↗ Chat
DB Vertrieb	**Jonas vor dem Berge**	↗ E-Mail | ↗ Chat
	Stefan Schmeißer	↗ E-Mail | ↗ Chat
sowie	**Uwe Friedrichsen, CTO codecentric AG**	↗ E-Mail

… sowie freundlicher Unterstützung durch die strategischen Partner der DB Systel

Thorsten Gau
CTO IBM Consulting

Markus Eisele
Developer Strategy

Bernd Drothen
Deutschland-Verantwortlicher Solution Engineering

Dave Wright
Chief Innovation Officer

Sie haben Fragen oder möchten mit uns zum Thema „Softwareentwicklung 2030“ 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.