



Business Briefing

Platform Engineering

Steigere Performance, Stabilität und Sicherheit für maximalen Business Value. Das Business Briefing hilft dir, den Mehrwert zu vermitteln und deine Peers zu überzeugen.

Business Value steigern – mit der richtigen Plattformstrategie

Wie Platform Engineering Unternehmen hilft, ihre Effizienz zu erhöhen

Platform Engineering ist mehr als ein technischer Ansatz – es ist eine **strategische Initiative**, die Unternehmen dabei unterstützt, **Anwendungen schneller bereitzustellen**, Wartungskosten zu senken und den **Business Value zu maximieren**. Durch **automatisierte Self-Service-Plattformen** wird die Entwickler-Erfahrung verbessert, was Produktteams ermöglicht, effizienter zu arbeiten und Innovationen schneller umzusetzen.

Mit einer **strategischen Analyse** und dem **Brownfield-Ansatz**, der bestehende Systeme integriert, lassen sich **schnelle Erfolge erzielen** – mit nachhaltigen Vorteilen für Entwicklungsteams und das gesamte Unternehmen.

Laut Gartner werden bis 2026 rund 80 % der grossen Softwareentwicklungsunternehmen dedizierte Plattform-Entwicklungsteams etablieren, um standardisierte, wiederverwendbare Services bereitzustellen.

Platform Engineering als strategische Investition

Die 4 wichtigsten Gründe, warum es sich für das Business langfristig lohnt



Time-to-Market

Produkte und Neuerungen werden schneller und kostengünstiger auf den Markt gebracht, standardisiert betrieben und den Kunden zur Verfügung gestellt.

Herausforderung

- Viele manuelle Tätigkeiten, komplexe, teamübergreifende Prozesse bei der Bereitstellung und Wartung von IT-Infrastruktur sowie Softwareentwicklungs- und Test-Tools beeinträchtigen die Effizienz der Entwickler.
- Dies führt zu langen Release-Zyklen und Verzögerungen bei der Einführung neuer Funktionen, die den Kunden- oder Geschäftsnutzen erhöhen.
- Neue Produkt-Launches und Marketingaktivitäten sind stark voneinander abhängig und können erst nach Releases erfolgen.

Nutzen

- Entwickler erhalten alle notwendigen Komponenten – von IT-Infrastruktur über Entwicklungs- und Test-Tools bis hin zu automatisierten Qualitätsprüfungen und Deployments.
- Durch CI/CD, Dokumentations-Templates und Infrastructure-as-Code können sie Features und Produkte schneller live schalten und direktes Kundenfeedback einholen.



Kundenzufriedenheit

Die Kundenzufriedenheit steigt dank beschleunigter Bereitstellung von gewünschten Funktionalitäten sowie besserer Qualität der Applikationen.

Herausforderung

- Ohne Platform Engineering werden sehr heterogene Applikationsstrukturen und Deployment Prozesse realisiert. Diese sind nicht nur zeitraubend sondern bergen auch ein hohes Qualitäts- und Sicherheitsrisiko.
- Applikationen können oft schon nach kurzer Zeit nicht mehr gewartet werden, wodurch Kundenwünschen nicht mehr nachgekommen werden kann.

Nutzen

- Dank Platform Engineering können sich Entwickler und Produktteams auf die Umsetzung der eigentlichen Business-Funktionalität fokussieren.
- Diese wird dadurch über den gesamten Lebenszyklus schneller realisiert und stabiler bereitgestellt – der Kundennutzen steigt.



Attraktiver Arbeitgeber

Entwicklern wird eine Cloud-basierte moderne Arbeitsumgebung geboten, die den Aufwand reduziert und ihre Arbeit spannender macht.

Herausforderung

- Entwickler verbringen zu viel Zeit mit der Einrichtung von Entwicklungsumgebungen, der Suche nach und dem Aktualisieren von verstreuter Dokumentation und dem Abstimmen mit anderen Teams.
- Fehlende standardisierte Deployment-Prozesse führen zu Verzögerungen und erhöhtem Arbeitsaufwand sowie Nacht- oder Wochenendeinsätzen.

Nutzen

- Entwickler können sich auf das Coden konzentrieren, da Infrastruktur, CI/CD-Pipelines und Softwarekomponenten automatisch bereitgestellt und gewartet werden – ohne manuelle Eingriffe oder lange Freigabeprozesse.
- Eine moderne Entwicklungsumgebung steigert die Attraktivität des Unternehmens und hilft, qualifizierte Fachkräfte zu gewinnen.



Governance und Security

Alle Sicherheitsstandards und Regularien werden zentral gesteuert und sind Governance-konform.

Herausforderung

- Die Einhaltung vorgeschriebener Sicherheitsstandards und einheitlicher Governance Strukturen ist oft schwierig. Dies führt dazu, dass nicht alle Produkte den Compliance- und Sicherheitsvorgaben entsprechen und Lücken erst spät oder gar nicht erkannt werden.
- Zukünftige Anforderungen zur Cyber Security (z.B. Cyber Resilience Act der Europäischen Union) müssen in allen Produkten umgesetzt werden.

Nutzen

- Dank Platform Engineering werden Sicherheitsprüfungen und Compliance-Überwachungen automatisiert. Dies minimiert Risiken und gewährleistet die Einhaltung von Vorschriften zentral.
- Integrierte Monitoring- und Sicherheitstools sorgen für konsistente Implementierungen und Behebung allfälliger Sicherheitslücken sowie eine effiziente und transparente Steuerung.

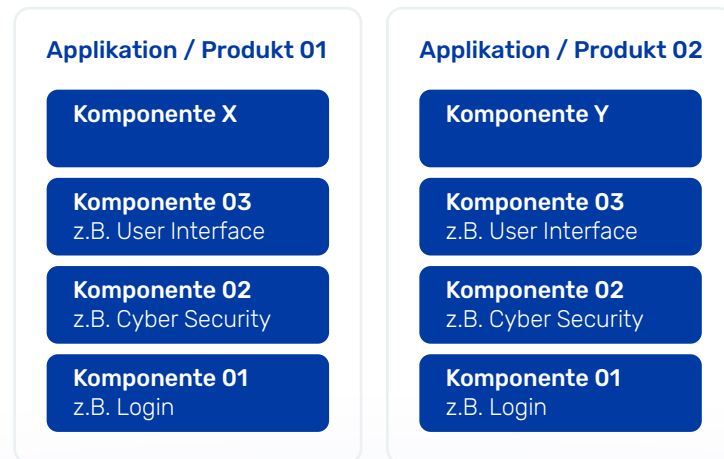
Plattformstrategie statt individuelle Entwicklung

Funktionalitäten und Prozesse werden als Self-Service bereitgestellt

Ein zentrales Plattform-Setup ersetzt mehrfach implementierte Basisstrukturen für jede einzelne Anwendung. Statt redundanter Setups profitieren alle Applikationen von einer einheitlichen, wiederverwendbaren Plattform – für mehr Effizienz, Konsistenz und Skalierbarkeit.

■ = Individuell entwickelt ■ = Individuell entwickelt und in Plattform aufgenommen ■ = «Re-Use» gebauter Komponenten

Individuelle Entwicklung

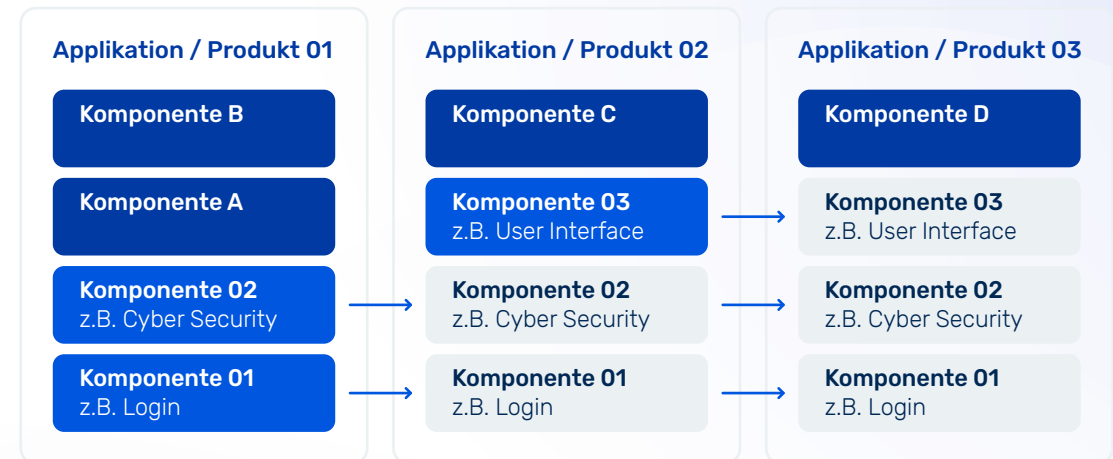


➔ Gleichbleibender Entwicklungsaufwand

Arbeitsschritte pro Applikation / Produkt

Dev > Build > Test > Release > Maintenance

Entwicklung mit einer Plattformstrategie



+ Repository & Funktionen auf Plattform wachsen

– Entwicklungsaufwand pro Applikation / Produkt nimmt ab

Arbeitsschritte pro Applikation / Produkt

werden effizienter durch Nutzung bereits gebauter Komponenten und Prozesse

Plattform als Self-Service



Nutzen für IT-Teams durch Plattform Engineering

So profitieren die Entwickler-Teams – Die 5 wichtigsten Vorteile

Zukunftsfähige Entwicklungsumgebung

- ✓ Eine standardisierte, benutzerfreundliche Plattform reduziert den Setup- und Wartungsaufwand, ermöglicht konsistente Entwicklungsprozesse und Dokumentation und fördert eine höhere Effizienz sowie schnellere Innovationszyklen.
- ✓ Self-Services reduzieren die kognitive Belastung, wodurch Entwickler sich auf Innovationen konzentrieren können.

Effiziente Deployment- und Entwicklungsprozesse

- ✓ Automatisierte CI/CD-Pipelines sorgen für eine zuverlässige, standardisierte und sichere Softwarebereitstellung.
- ✓ Integrierte Sicherheitsmechanismen und Qualitätsprüfungen reduzieren Risiken, während stabile und wiederholbare Prozesse die Time-to-Market verkürzen.

Entlastung der DevOps-Teams

- ✓ Self-Service Developer Portals ermöglichen es Entwicklern, benötigte Ressourcen eigenständig und standardisiert zu provisionieren.
- ✓ Dies reduziert Abhängigkeiten, beschleunigt Entwicklungsprozesse und entlastet DevOps-Teams, sodass sie sich auf strategische Aufgaben konzentrieren können.

Konsistente Sicherheits- und Compliance-Standards

- ✓ Durch zentral definierte Governance- und Sicherheitsrichtlinien werden Implementierungen standardisiert und automatisiert überprüft.
- ✓ Einheitliche Vorgaben und integrierte Sicherheitsmechanismen minimieren Risiken und erleichtern die Compliance-Einhaltung.

Effizientes Onboarding neuer Entwickler

- ✓ Durch standardisierte Entwicklungsumgebungen und klare, zentrale Dokumentationen können neue Entwickler schneller produktiv werden.
- ✓ Automatisierte Setups und Self-Service-Zugänge reduzieren den manuellen Aufwand und erleichtern den Einstieg ins Team.

Mit dem ipt-Vorgehen zur individuell zugeschnittenen Developer-Plattform

Brownfield-Ansatz: Bestehende Tools und Prozesse analysieren, optimieren und schrittweise transformieren

01

Analyse im Brownfield-Ansatz

- **Definition der Plattformstrategie**
Klare Zielsetzung und Ausrichtung der Plattformen im Einklang mit den Unternehmensanforderungen. Analyse fördert den Produktgedanken, weil sie die Plattform aus Nutzer- und Kundensicht betrachtet (Grundlage für *Platform-as-a-Product* im Betrieb).
- **Bedarfsanalyse**
Ermittlung der notwendigen Ressourcen, Tools und Technologien für die Umsetzung.
- **PMM-Bewertung* (falls notwendig)**
Priorisierung und Bewertung von Fähigkeiten zur effizienten Ressourcennutzung und Maximierung des Mehrwerts.
- **Identifikation von Verbesserungsbereichen**
Ermittlung der Prozesse oder Bereiche, die optimiert werden müssen, um die Plattform zu steigern.

02

Action mit Prototyping für schnellen Mehrwert

- **Prototyping**
Entwicklung eines Prototyps, um erste Feedbacks zu sammeln.
- **Implementierung**
Umsetzung der definierten Plattform Komponenten mit Fokus auf effiziente Bereitstellung und qualitativ hochwertige und schnelle Ergebnisse.
- **Integration und Enablement**
Nahtlose Integration der neuen Lösungen in bestehende Systeme und Prozesse, um den grösstmöglichen Nutzen zu erzielen.

03

Plattform als Produkt betreiben

- **Neue Anforderungen**
Kontinuierliche Aufnahme und Planung neuer Anforderungen, um die Plattformen fortlaufend zu verbessern und an die Bedürfnisse des Unternehmens anzupassen.
- **PMM nutzen***
Bewertung des Fortschritts durch regelmässige PMM-Bewertungen. Damit wird im Produkt Management das Level bewertet, daraus die Entwicklung gesteuert und Reifegrade iterativ gesteigert.
- **Optimierung**
Ständige Überwachung und Verbesserung der Plattform, um Effizienz, Skalierbarkeit und Sicherheit kontinuierlich zu erhöhen.

* **PMM:** Das Platform Engineering Maturity Model ist ein Framework, um die Reife von Plattform-Engineering-Prozessen zu bewerten und zu verbessern. Es hilft dabei, interne Plattformen durch strategische Planung, bewusste Reflexion und kontinuierliche Verbesserung effizienter und skalierbarer zu gestalten. Das Modell berücksichtigt verschiedene Reifegradstufen und fordert eine Balance zwischen Technologie, Menschen, Prozessen und Geschäftsergebnissen.



Kein PDF ersetzt das persönliche Gespräch.

Lassen Sie uns in einer kurzen Demo zeigen, wie
Platform Engineering echten Mehrwert schafft.

Jetzt Termin anfragen – wir freuen uns auf den Austausch!