

# Platform-based Software Engineering

## Beschleunigte Entwicklung durch vorgefertigte Entwicklungsplattformen

Anwendungssoftware wird nicht mehr von Grund auf neu entwickelt, sondern auf der Grundlage von Softwareplattformen und als Teil von Softwareökosystemen. Das führt zu einer beschleunigten Lösungsentwicklung, wirkt sich aber massiv auf die Art und Weise der Entwicklung aus.

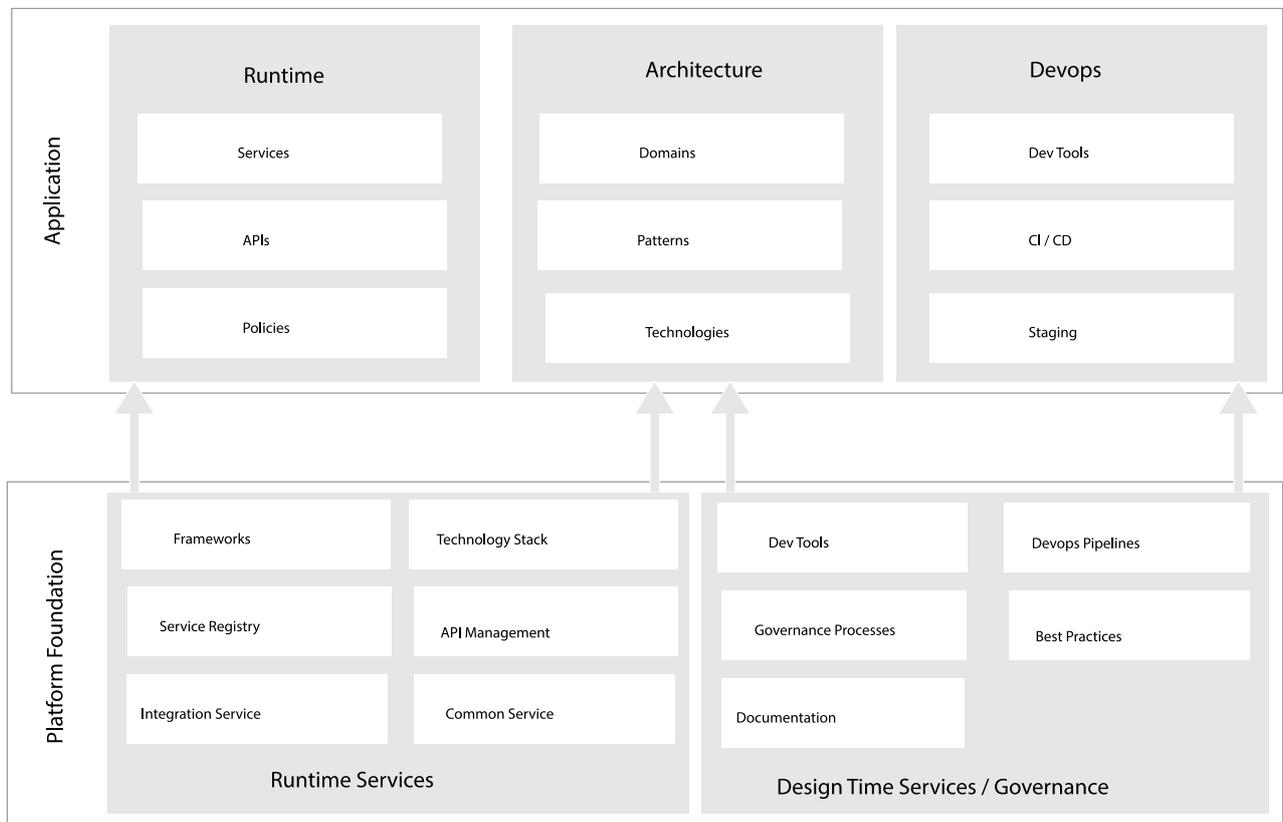
### Definition

Bei Platform-based Software Engineering sind Laufzeitumgebung, Technologiestack und Entwicklungsrichtlinien strikt vorgegeben. Nur innerhalb dieses Rahmens dürfen Entwickler Entscheidungen treffen, die für ihre Anwendung notwendig sind. Sie profitieren dafür von einer einheitlichen Vorgehensweise, kurzen Entwicklungszyklen und Fokus auf die Umsetzung des fachlichen Codes. Zwei Perspektiven sind relevant: die Laufzeitsicht und die Entwicklungssicht.

Laufzeitplattformen stellen für eine darauf basierende Anwendung die Laufzeitumgebung bereit. Die Anwendung kann also nicht außerhalb der Plattform betrieben werden. Die Plattform stellt der Anwendung eine Reihe von Services zur Verfügung, auf die diese aufbaut. Hierzu gehören Common Services wie Monitoring und Logging, aber auch Integration Services, etwa Authentifizierung. Zugleich liefert die Plattform eine Auswahl von Frameworks und Technologiestacks, auf die die Anwendung aufsetzen

kann. Um die Integration von Anwendungen innerhalb der Plattform zu vereinfachen, finden sich hier typischerweise ebenso eine Service Registry und ein vereinheitlichtes API-Management. Die Plattform definiert zugleich Security- und Compliance-Standards und forciert deren Umsetzung durch entsprechende Policies.

Aus Entwicklungssicht sind vor allem die Plattform-Governance-Prozesse zu nennen, die das On- und Offboarding einer



Anwendung regeln. Die Plattform liefert Muster für die Umsetzung typischer Anforderungen, die sich in Best Practices und Dokumentationen niederschlagen. Typischerweise stellt die Plattform auch eine DevOps-Pipeline sowie für die plattformbasierte Entwicklung optimierte Entwicklungswerkzeuge zur Verfügung. Die plattformbasierte Entwicklung selbst wird durch Competence-Teams unterstützt, die spezifische Erfahrungen mit Technology Stacks oder Plattform Services haben und als Know-how-Träger zur Verfügung stehen.

Für die Anwendungen ergeben sich zahlreiche Abhängigkeiten. Es beginnt bei der Architektur der Anwendung, die die Laufzeitumgebung der Plattform berücksichtigen und beispielsweise Common- und Integration Services nutzen muss. Um eine Wiederverwendung von bereits existierenden fachlichen Services zu ermöglichen, ist bei der Strukturierung der Anwendung ein domänenbasierter Ansatz empfehlenswert. Dieser Ansatz identifiziert oft Funktionalitäten, die eventuell anderen Projekten auf der Plattform als Services zur Verfügung gestellt werden können. Für die Verwendung von Plattform Technology Stacks und Frameworks müssen beim Anwendungsdesign Muster und Technologien identifiziert und auf das Plattformangebot und unterstützte Best Practices übertragen werden. Auch für die Interoperabilität der Anwendung muss im Design das API Management und Service Registry der Plattform Berücksichtigung finden. Der Entwicklungsprozess selbst hat eine enge Nähe zur Plattform, da es gilt, die für die Plattform optimierten Werkzeuge und die integrierten DevOps-Pipelines zu benutzen. Spätestens jedoch beim Staging-Konzept sind die Vorgaben der Plattform bindend. Zur Laufzeit ist die Anwendung dann eng an

### Plattform Entwicklung

- Entwicklung neuer Plattformen
- Weiterentwicklung der Plattform
- Plattform Abhängigkeiten

### Design Prinzipien und Methoden

- Domain Driven Design (DDD)
- Service Oriented Design (SOA)
- Functional Domain Modelling (FDM)

die Plattform gekoppelt, da sie deren APIs und Services benutzt.

Das Entwicklungsteam der Anwendung muss die Plattform selbst verstehen und dabei das Know-how der Plattform Competence Teams nutzen. Nicht zuletzt sollte frühzeitig im Projekt eine Kostenbetrachtung stattfinden, die sowohl die Onboarding-Kosten als auch die Betriebskosten der Anwendung auf der Plattform berücksichtigt.

### Referenzszenario

Bisher bildeten die Geschäftsanforderungen die Grundlage für eine individuelle Wahl der Entwicklungsumgebung, der Programmiersprachen, der Werkzeuge und des Entwicklungsprozesses. Dies führte zu isolierten Anwendungen, unterschiedlichen Architekturen und Erfahrungsergebnissen von Projekt zu Projekt. Das und die zudem schwierige Integration mit anderen Anwendungen will das Unternehmen vermeiden und die Wiederverwendung von Komponenten ermöglichen.

Es setzt auf plattformbasierte Entwicklung. Dabei akzeptiert es bewusst, die Projekte in ihrer Freiheit einzuschränken, senkt aber die Entwicklungskosten durch bewährte Verfahren, vereinheitlichte Arbeitsweisen und innovative Erweiterungen der Plattform. Das Unternehmen

### Plattform Betrieb

- Plattform Prozess
- Plattform Kosten
- Plattform Teams

### Software Entwicklungsprozess

- synchron mit Plattform-Weiterentwicklungsprozess
- Wasserfall
- Agil (Scrum / Safe)
- Hybrid

profitiert von den implementierten Best Practices, Entwicklungsprozessen, Werkzeugen und Technologien. Zudem können die Anwendungen übergreifend miteinander interagieren.

### Potenzial

Konsumenten von plattformbasierten Anwendungen profitieren von einheitlichen Gestaltungen und Verhalten der verschiedenen Anwendungen, weil sie nach denselben Prinzipien entwickelt wurden und idealerweise gut miteinander integriert sind.

Das Unternehmen, das plattformbasierte Softwareentwicklung einsetzt, profitiert vor allem durch die Standardisierung von Entwicklungsprozessen, die zentrale Governance für kritische Bereiche wie Anwendungssicherheit, die Wiederverwendung allgemeiner Konzepte und Services sowie die Bündelung von Kompetenzen und deren Nutzung über Projektgrenzen hinweg. Die Innovationsgeschwindigkeit und -breite steigt, weil Innovationen in die Plattform einfließen und damit allen Projekten zur Verfügung stehen.

Ein Projekt, das seine Anwendung auf der Basis der Plattform aufbaut, profitiert von der Wiederverwendung bestehender Komponenten, Prozesse und dem Wissen der Plattform-Teams.

PSE

## Reifegrad

Alle großen Unternehmen haben den Trend erkannt und nutzen seit mehr als zehn Jahren Basisplattformen zur Umsetzung ihrer Cloud-Strategie. In den letzten fünf Jahren haben diese Unternehmen ihre eigenen internen Entwicklungsplattformen entwickelt, die in der Regel auf bestehenden Basisplattformen aufsetzen. Gleichzeitig hat die Verfügbarkeit von Softwareökosystemen auf dem Markt die Möglichkeit für junge Unternehmen geschaffen, ihre Anwendungslandschaft vollständig auf diesen Ökosystemen aufzubauen, ohne eine eigene Infrastruktur und eigene Dienste betreiben zu müssen.

## Marktübersicht

Basisplattformen sind Allzweckplattformen, wie sie durch die bekannten Cloud-Provider Amazon, Microsoft und Google bereitgestellt werden. Diese bieten Laufzeitumgebungen auf verschiedenen Diensteebenen wie PaaS und FaaS an und kommen mit einem breiten Entwick-

lersupport und DevOps-Werkzeugen.

Interne Entwicklungsplattformen (IDP) werden in der Regel von größeren Unternehmen betrieben, um die unternehmensinterne Anwendungsentwicklung auf eine gemeinsame Basis zu stellen. Diese setzen oft auf Basisplattformen auf, werden von Plattformentwicklungsteams im Unternehmen entwickelt und betrieben. Charakteristisch für diese IDPs ist deren Integration in unternehmensinterne Lösungen wie Monitoring und Identity-Management.

Softwareökosysteme sind Plattformen, die in der Regel auf eine bestimmte Anwendungsdomäne wie Banken oder Versicherungen oder eine bestimmte Technologie ausgerichtet sind. Sie berücksichtigen in hohem Maße die Spezifika der entsprechenden Märkte und bieten Marktplätze für Lösungen, die innerhalb des Ökosystems entwickelt wurden. Auch Softwareökosysteme setzen in der Regel auf Basisplattformen

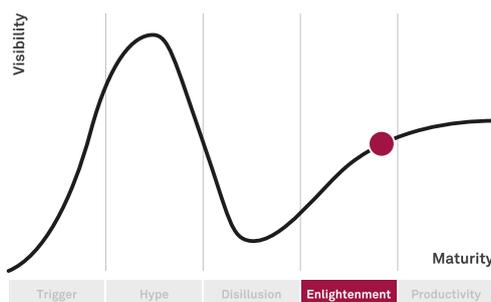
auf. Typische Beispiele sind hier Salesforce, Servicenow, SAP und Metaverse.

## Alternativen

Klassische Vorgehensweisen, etwa Anwendungen von Grund auf neu erstellen, sind weiterhin eine Option.

## Fazit

- + fokussiert Funktionalität
- + einheitliches Applikationsdesign
- + setzt bewährte Verfahren und nicht-funktionale Anforderungen durch
- + gemeinsames Modell für die Zusammenarbeit zwischen den Anwendungen
- + fördert Wissensaustausch
- weniger Freiheit bei der Framework-Wahl
- eingegrenzter Lösungsraum
- Plattform-lock-in
- Plattform-Abhängigkeiten zur Laufzeit
- schwierige Analyse von Laufzeitproblemen



### Buzzword Factor (Ent./Customer)

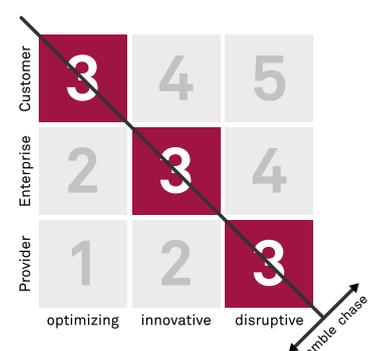
1 low	2 medium	3 high
----------	-------------	-----------

### Entry Barrier (Provider)

1 low	2 medium	3 high
----------	-------------	-----------

### Benefit Level (Provider)

1 low	2 medium	3 high
----------	-------------	-----------



<https://msg.direct/techrefresh>

Stand: Dezember 2023

## msg systems ag

Robert-Bürkle-Straße 1 | 85737 Ismaning/München | Telefon: +49 89 96101-0 | Fax: +49 89 96101-1113 | [www.msg.group](http://www.msg.group) | [info@msg.group](mailto:info@msg.group)