



Alles Wichtige über Softwarearchitektur in 40 Minuten

STEFAN ZÖRNER, EMBARC

JavaLand 2023 im Phantasialand Brühl

Dienstag, 21.03.2023, 14:00-14:40 Uhr



Alles Wichtige über Softwarearchitektur ...



Zusammenfassung

"Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams." -- so sagte es das agile Manifest bereits vor 20 Jahren. Und tatsächlich sind die überzeugendsten Softwarearchitekturen heute kein Werk einzelner mehr, und kein lemminghaftes Verfolgen von Trends. Softwarearchitektur ist Teamsache geworden -- sie geht alle an.

Du bist Softwareentwickler:in und möchtest verstehen, was Architektur heute ausmacht? Was Du und Dein Team mindestens tun solltet, um passende Lösungen zu entwerfen, sie überzeugend zu kommunizieren und mitunter auch zu hinterfragen?

In diesen Vortrag vermittele ich das meiner Meinung nach Wichtigste, was jedes Team zum Thema wissen, beachten und tun muss. Zumindest dann, wenn scheitern keine Option ist. Viel ist es nicht. Wir sprechen über Ziele, Architekturentscheidungen, Risiken und Kompromisse ...

Ihr nehmt ein verdichtetes Bild mit, mit dessen Hilfe Ihr eure eigene Situation einordnen und bei Bedarf nachjustieren könnt.



Stefan Zörner

- Softwareentwickler + -architekt bei embarc in Hamburg
- Vorher oose, IBM, Mummert + Partner, Bayer AG, ...

Schwerpunkte:

- Softwarearchitektur (Entwurf, Bewertung, Dokumentation)
- Cloud- und Java-Technologien

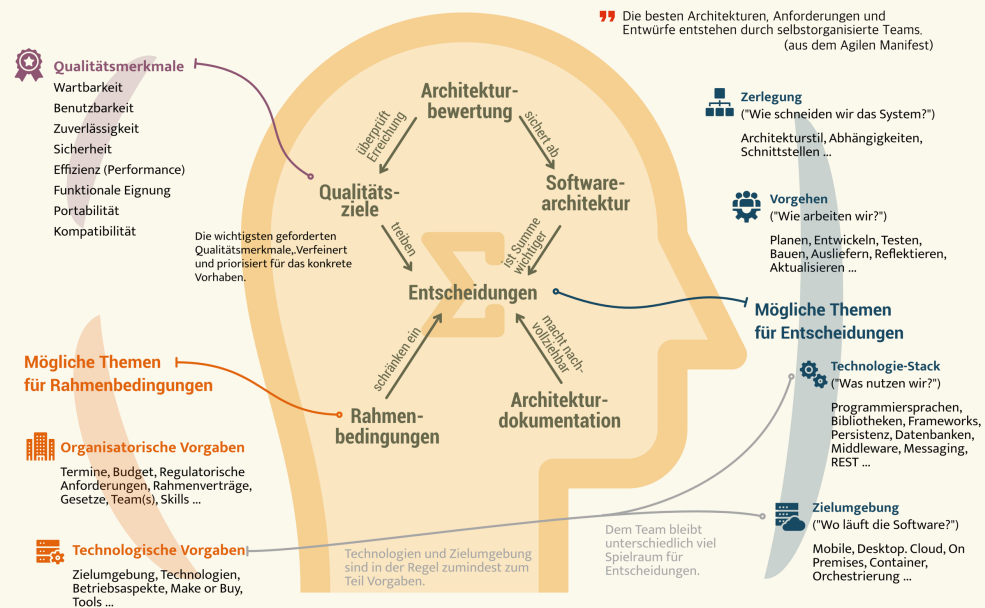


✉ Stefan.Zoerner@embarc.de
 🐦 [@StefanZoerner](https://twitter.com/StefanZoerner)
 🌐 [linkedin.com/in/stefan-zoerner](https://www.linkedin.com/in/stefan-zoerner)
 📧 [xing.to/szr](https://www.xing.to/szr)
 🐘 [@StefanZoerner@mastodon.social](https://mstdn.social/@StefanZoerner)



EIN KOPF VOLLER SOFTWAREARCHITEKTUR

Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams. (aus dem Agilen Manifest)



Agenda

- 1 Was (genau) ist eigentlich Softwarearchitektur?**
- 2 Entscheidungen treffen
- 3 Lösungsansätze kommunizieren
- 4 Lösungen bewerten
- 5 Fazit und Weitere Informationen



Agenda

- 1 Was (genau) ist eigentlich Softwarearchitektur?**
- 2 Entscheidungen treffen
- 3 Lösungsansätze kommunizieren
- 4 Lösungen bewerten
- 5 Fazit und Weitere Informationen

1





Ein Beispiel: Die Website embarc.de



Steckbrief:

- Unternehmensauftritt von embarc
- Stellt uns und unsere Leistungen vor, im Detail
 - Team-Mitglieder
 - Fachliche Infos (z.B. Architektur-Spicker)
 - Blog-Beiträge und Termine (Vorträge, Workshops, ...)
 - Kontaktformular
 - Anmeldung zum Newsletter
 - ...
- Entwicklung in 2020, Live seit Januar 2021

→ <https://www.embarc.de>



Kernelemente der Lösung (Entscheidungen) (1/2)

- **Hugo** als statischer Website-Generator („Buy“)
- Verwaltung der Seiten-Inhalte („Eingabe“) auf **GitHub**
 - Inhalte (Leistungen, Blog-Beiträge ...) als **Markdown**
 - Abbildungen (falls klein)
- Theme **selber entwickelt** (Design mit externer Unterstützung) („Make“)
- Lokales „Entwickeln“ und „Testen“ mit sehr kurzen Roundtrip-Zeiten (Hugo generiert im Hintergrund)



Kernelemente der Lösung (Entscheidungen) (2/2)

- Kollaboration, Teilen der Änderungen über **GitHub**
- Automatisches Generieren mit **Hugo** auf Serverseite
- Auslieferung der produktiven Seite auf Cloud-Dienst **Netlify**
- Größere Downloads über Amazon Web Services **S3** / **Cloudfront** CDN
- ...



Was ist Softwarearchitektur?

Softwarearchitektur :=

Σ wichtige Entscheidungen

- fundamental (betrifft viele)
- im weiteren Verlauf nur schwer zu ändern
- entscheidend für den Erfolg Eures Softwaresystems



Kleine Zeitreise: Die Java-Welt 2002



Java um die Jahrtausendwende

- Technologie ist ausgewachsen (Sprache seit 1995 verfügbar)
- Unternehmen beginnen geschäftskritische Anwendungen in Java zu realisieren
- Standard-Java dazu: J2EE („Enterprise Edition“)

Kritik an J2EE

- Geringe Entwicklerproduktivität
- Einfache Dinge nicht angemessenen leicht, das galt insbesondere für das Komponentenmodell EJB
- Prominente Stimme: Rod Johnson



Ein zweites Beispiel: Das Spring Framework

The screenshot shows the Spring Framework website. The main heading is 'Spring Framework 5.0.6'. Below the heading are three tabs: 'OVERVIEW', 'LEARN', and 'SUPPORT'. The 'OVERVIEW' tab is active. The text describes the Spring Framework as a comprehensive programming and configuration model for modern Java-based enterprise applications. It mentions that a key element is infrastructural support at the application level, focusing on the 'plumbing' of enterprise applications. There is also a section for 'Support Policy and Migration' and a 'Features' section.



S. Zörner: "Alle

13

Kernelemente der Lösung (Auswahl)

Lösungsansätze in Spring, Entscheidungen ca. 2002 getroffen

- POJOs (Plain Old Java Objects) als Komponenten
- Konfiguration über XML (später Annotationen)
- Dependency Injection
- AOP (Aspektorientierte Programmierung) für querschnittliche Themen wie Transaktionen, Security, ...
- Unterstützung zahlreicher Technologien (Persistenz, Messaging) nach gleichem Schema ...
- ...

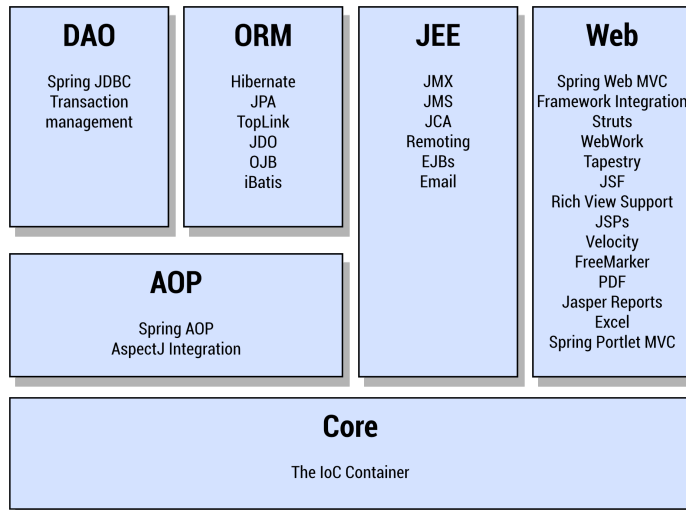


S. Zörner: "Alles Wichtige über Softwarearchitektur"

embarc.de

14

Das Spring Framework ca. 2004



Abkürzungen (Auswahl)

DAO	Data Access Object
ORM	Objekt-/Relationales Mapping
AOP	Aspekt-orientierte Programmierung
JEE	Java Enterprise Edition
JPA	Java Persistence API
JMS	Java Message Service
EJB	Enterprise JavaBeans
MVC	Model View Controller
JSF	JavaServerFaces
JSP	JavaServerPages
IoC	Inversion of Control

Spring Framework



Themen für Entscheidungen

Zerlegung

Welcher Architekturstil?
Wie zerfällt die Anwendung?
Teilsysteme, Module,
Komponentenbildung,
Abhängigkeiten ...



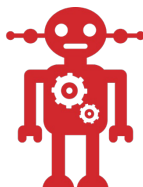
Zielumgebung

Wo läuft die Software?
Beim Endanwender, im
eigenen Rechenzentrum,
Cloud, Verteilung,
Virtualisierung ...



Technologie-Stack

Was setzen wir ein?
Programmiersprache(n)
Bibliotheken, Frameworks,
Middleware,
Querschnittsthemen

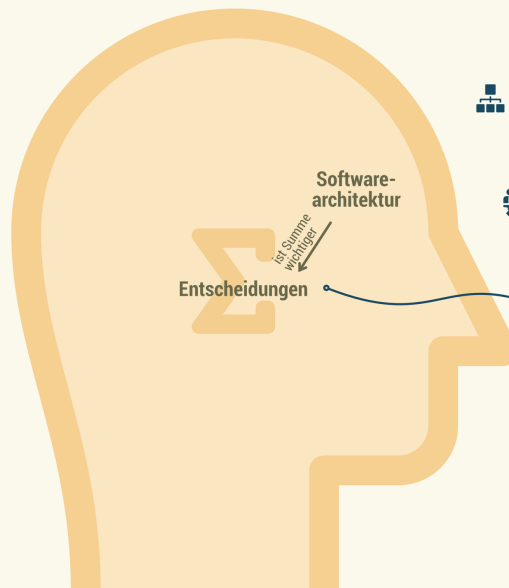


Vorgehen

Wie arbeiten wir?
Planen, Entwickeln, Testen,
Bauen, Dokumentieren,
Ausliefern, Nachjustieren,
...



EIN KOPF VOLLER SOFTWAREARCHITEKTUR



Zerlegung
("Wie schneiden wir das System?")

Architekturstil, Abhängigkeiten,
Schnittstellen ...



Vorgehen
("Wie arbeiten wir?")

Planen, Entwickeln, Testen,
Bauen, Ausliefern, Reflektieren,
Aktualisieren ...

**Mögliche Themen
für Entscheidungen**



Technologie-Stack
("Was nutzen wir?")

Programmiersprachen,
Bibliotheken, Frameworks,
Persistenz, Datenbanken,
Middleware, Messaging,
REST ...



Zielumgebung
("Wo läuft die Software?")

Mobile, Desktop, Cloud, On
Premises, Container,
Orchestrierung ...

© embarc Software Consulting GmbH

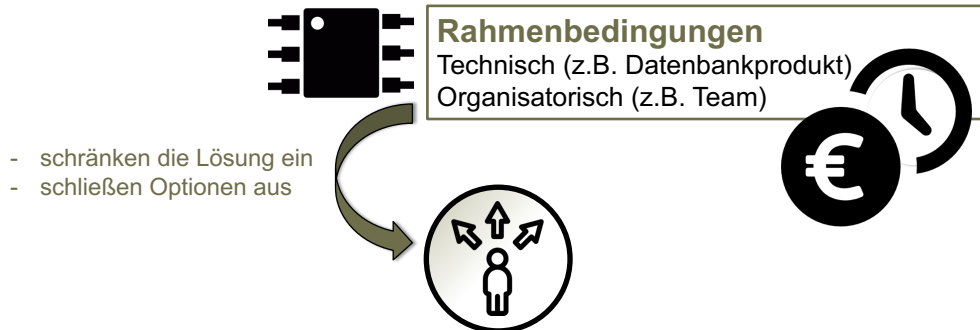
Agenda

- 1 Was (genau) ist eigentlich Softwarearchitektur?
- 2 Entscheidungen treffen**
- 3 Lösungsansätze kommunizieren
- 4 Lösungen bewerten
- 5 Fazit und Weitere Informationen

2



Einflüsse auf Entscheidungen



Rahmenbedingungen für embarc.de (Auswahl)



Organisatorische Vorgaben

- gesetzliche Aspekte (z.B. §5 TMG, DSGVO ...)
- beschränktes Budget (nicht explizit)
- Kein echter Termin (nur eigene Ungeduld)
- Team + dessen Skills (zum Bauen und später zum Pflegen)

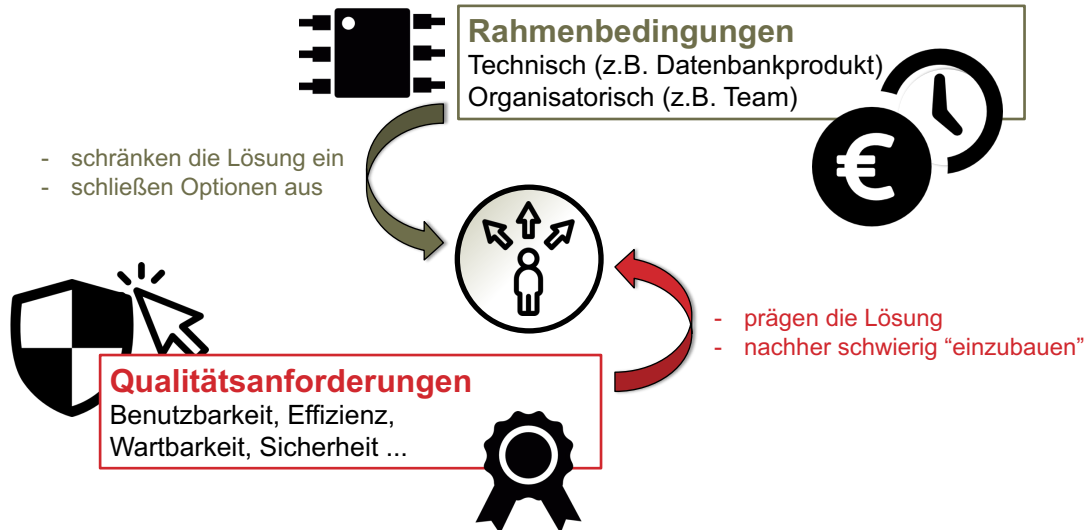


Technische Vorgaben

- Notebooks des Teams (Mac OS X)
- Übliche Webstandards
- SSL-Verschlüsselung der Seite
- Übernahme der Domäne embarc.de
- Migration bestimmter Inhalte (alte WordPress-Seite)



Einflüsse auf Entscheidungen



Qualitätsmerkmale

Begriffe
nach
ISO 25010




Eingabe eines Kennwortes

Username

StefanZ

Password

.....



Username

StefanZ


Password

geheim!123




Typische Wechselwirkung

Qualitätsmerkmale hängen voneinander ab.
Entscheidungen führen zu Kompromissen (engl. „Trade-off“).



Sicherheit
(Security)

Ist das System sicher vor Angriffen?
Sind Daten und Funktion vor unberechtigtem Zugriff geschützt? ...

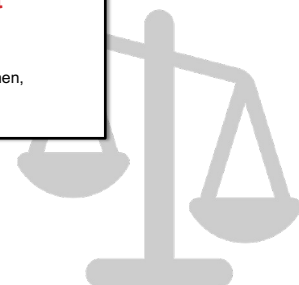


Benutzbarkeit
(Usability)






Ist die Software intuitiv zu bedienen,
leicht zu erlernen, attraktiv?



Benutzer empfinden Sicherheitsmaßnahmen als störend (z. B. Passworteingabe, CAPTCHAs)








Wichtige Architekturziele des Spring Frameworks

-  Hohe Entwicklerproduktivität
-  Unternehmenskritische Anwendungen ermöglichen
-  Auf lange Sicht wartbare Lösungen
-  Portable und kompatible Arbeitsergebnisse
-  Zukunftsfähiges Framework

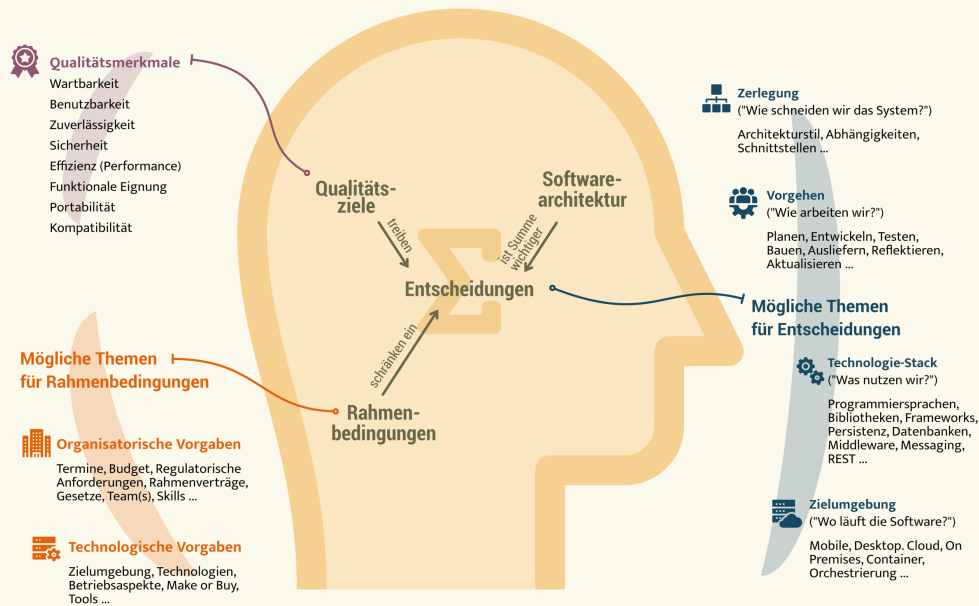


Wichtige Architekturziele des Spring Frameworks

Ziel	Beschreibung (und zugehöriges Software-Qualitätsmerkmal)
 Hohe Entwicklerproduktivität	Anwendungen auf Basis von Spring lassen sich effizient entwickeln. Java-Entwickler kommen gut mit dem Framework zurecht. Die dahinterliegenden Konzepte sind schnell verstanden und kein Hexenwerk. Entwickler konzentrieren sich auf die Fachlogik. <i>(Wartbarkeit, Benutzbarkeit (aus Entwicklersicht))</i>
 Unternehmenskritische Anwendungen ermöglichen	Spring erlaubt die Entwicklung von zuverlässigen und sicheren Applikationen, wie sie Unternehmen und Organisationen zur Unterstützung wichtiger Geschäftsfähigkeiten erwarten. <i>(Zuverlässigkeit, Sicherheit)</i>
 Auf lange Sicht wartbare Lösungen	Mit Spring realisierte Anwendungen sind leicht erweiterbar und änderbar. Genutzte Technologien (z.B. Bibliotheken, Middleware, Persistenz) lassen sich einfach aktualisieren und zu einem gewissen Grad auch austauschen. Das gilt insbesondere für Spring selbst. <i>(Erweiterbarkeit, Änderbarkeit)</i>
 Portable und kompatible Arbeitsergebnisse	Die Anwendungen sind auf allem relevanten Zielplattformen lauffähig (Applikationsserver wie z.B. WebSphere, Web-App-Server wie Tomcat, Standalone als Java-Prozess ...). Neue Java- oder Spring-Versionen brechen eine vorhandene Anwendung nicht. <i>(Kompatibilität, Portierbarkeit)</i>
 Zukunftsfähiges Framework	Spring kann dem technologischen Fortschritt folgen. Neue Produkte, Entwicklungsmethoden und Zielumgebungen sind rasch adaptiert. Die Entscheidung für Spring ist keine Sackgasse. <i>(Wartbarkeit (des Frameworks selbst))</i>



EIN KOPF VOLLER SOFTWAREARCHITEKTUR



Agenda


- 1 Was (genau) ist eigentlich Softwarearchitektur?
- 2 Entscheidungen treffen
- 3 **Lösungsansätze kommunizieren**
- 4 Lösungen bewerten
- 5 Fazit und Weitere Informationen

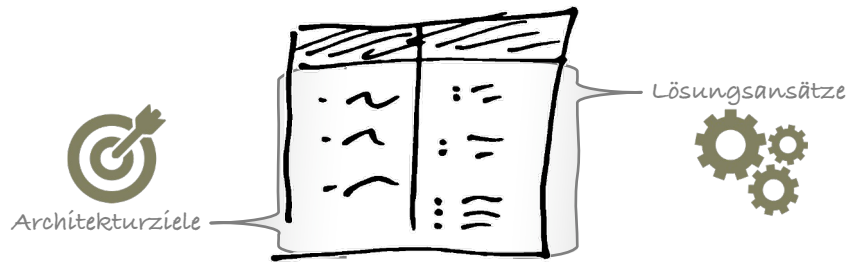
3








Lösungsstrategie als Mapping

Architekturziele und zugeordnete high-level Lösungsansätze.

 Form: z.B. Tabelle, [Ziele | Lösungsansätze].

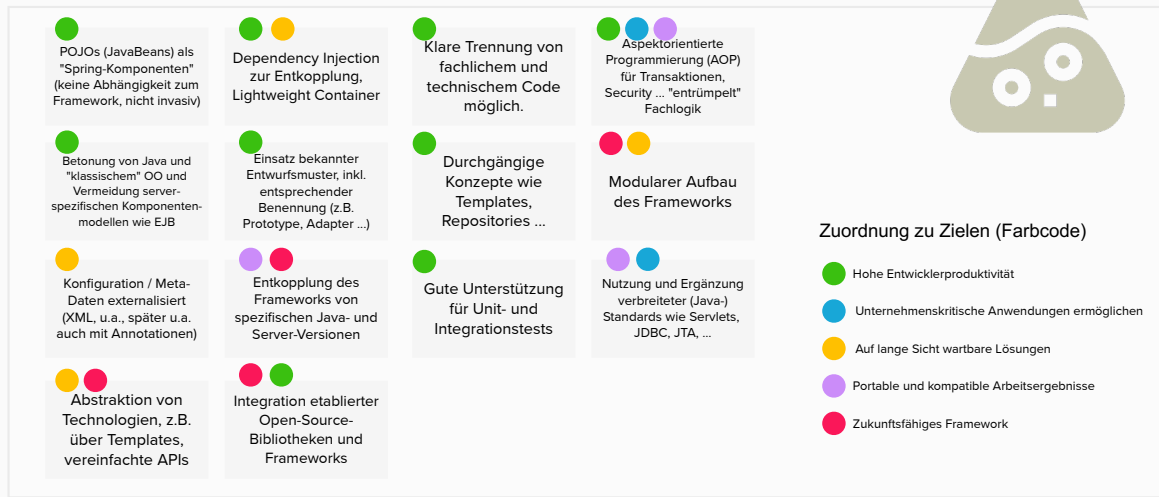


Wichtige Architekturziele des Spring Frameworks

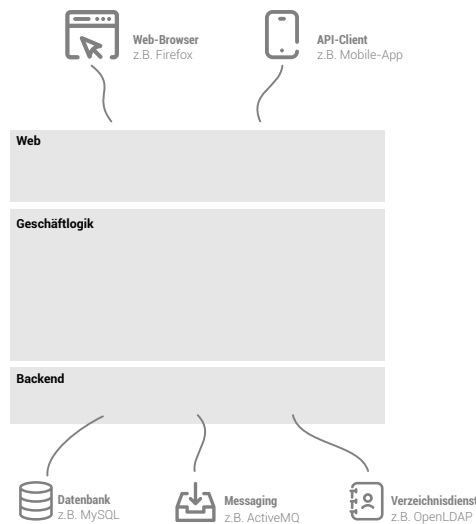
-  Hohe Entwicklerproduktivität
-  Unternehmenskritische Anwendungen ermöglichen
-  Auf lange Sicht wartbare Lösungen
-  Portable und kompatible Arbeitsergebnisse
-  Zukunftsfähiges Framework



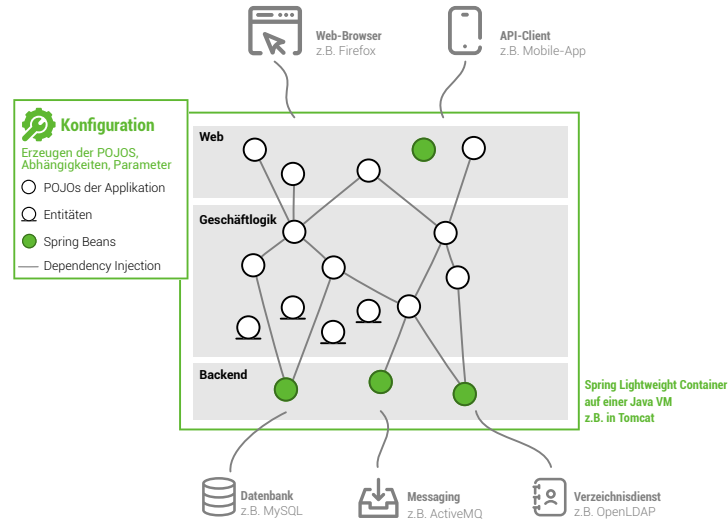
Zentrale Lösungsansätze von Spring



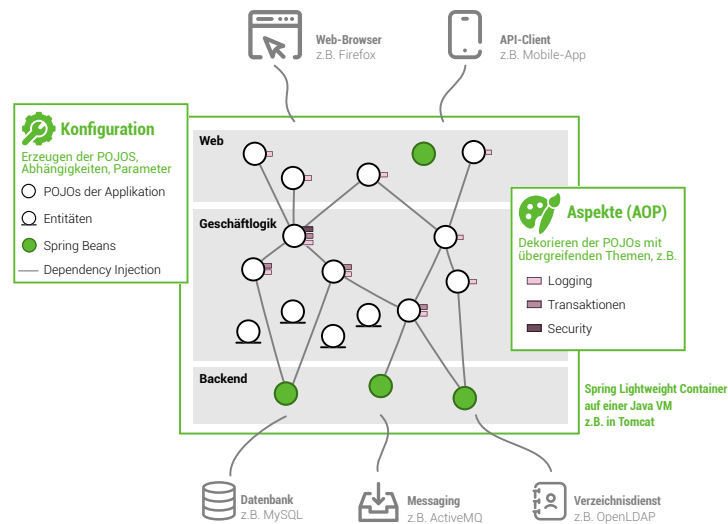
Eine Web-Anwendung im Rahmen von Spring



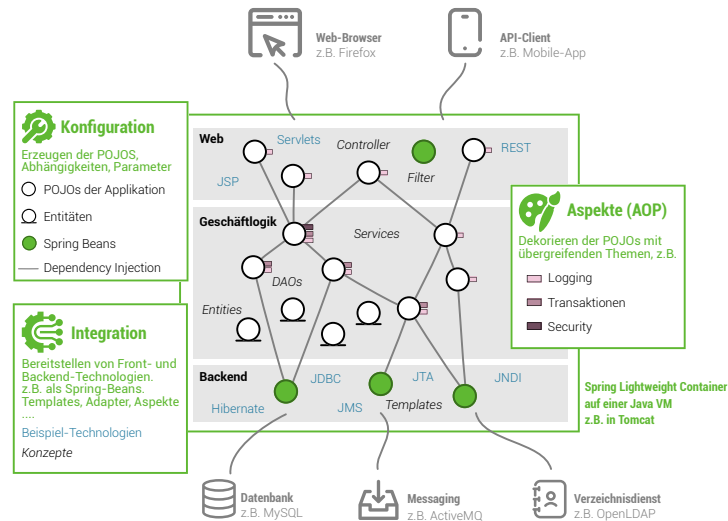
Eine Web-Anwendung im Rahmen von Spring



Eine Web-Anwendung im Rahmen von Spring



Eine Web-Anwendung im Rahmen von Spring



S. Zörner: "Alles Wichtige über Softwarearchitektur"

embarc.de

35



WAS sollen wir bauen?

Was wollen wir erreichen?
Wozu ist es da?
Wem nützt es?
Was soll es tun?
Was braucht es nicht tun?
Woran halten wir uns?
Was soll es exzellent können?

ANFORDERUNGEN

Architekturüberblick

Aufgabe

- Mission Statement
- Kontextabgrenzung

Einflüsse

- Rahmenbedingungen
- Qualitätsziele

Lösungsansätze

- Architekturstil
- Technologie-Stack
- Konzepte
- Prinzipien
- Zerlegung
-



Lösungsstrategie
(plus Überblicksbild)
als Brücke



WIE sieht die Lösung aus?

Wie erreichen wir das?
Welchen Mustern folgen wir?
Was verwenden wir?
Was leitet uns?
Woraus besteht es?
Wie ist es strukturiert?

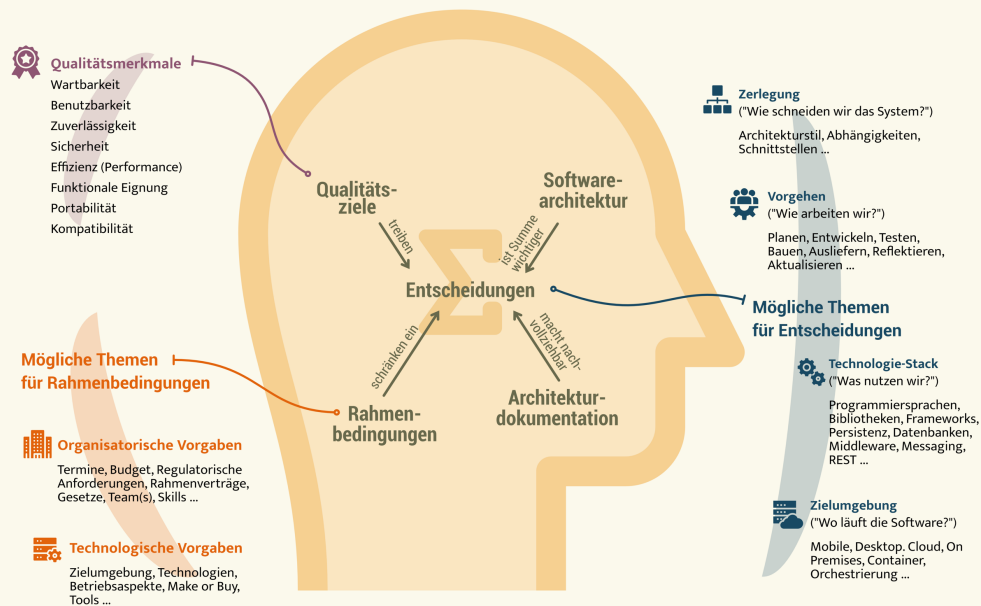
ENTSCHEIDUNGEN



S. Zörner: "Alles Wichtige über Softwarearchitektur"

Quelle: S. Zörner: Architektur ohne Firlefanz – Ihre Lösung auf einem Bierdeckel

EIN KOPF VOLLER SOFTWAREARCHITEKTUR



Agenda

- 1 Was (genau) ist eigentlich Softwarearchitektur?
- 2 Entscheidungen treffen
- 3 Lösungsansätze kommunizieren
- 4 **Lösungen bewerten**
- 5 Fazit und Weitere Informationen

4



Was sind Architektur-Reviews?



Wörtlich:
Review == etwas (über)prüfen, besprechen, ...

Gegenstand („etwas“) in unserem Fall:
Die Architektur(-entscheidungen) eines Softwaresystems

Typischer Begriff in der Fachwelt / -literatur auch:
Architekturbewertung (englisch „Evaluation“)

Kann durch Außenstehende erfolgen – muss aber nicht.



Kernelemente: Anlass, Gegenstand und Maßstab

Für eine Bewertung oder ein Review braucht es mindestens



Einen Anlass

Warum bewerten wir?



Einen Gegenstand

Was bewerten wir?



Einen Maßstab

Wonach (wo gegen) bewerten wir?

Typische Fragestellungen

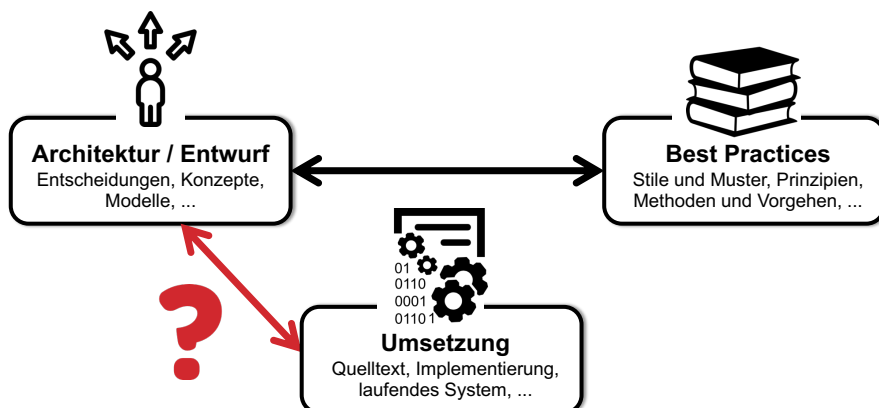
- Abgrenzung des Gegenstandes: Was gehört dazu, was nicht?
- „Reife“ des Gegenstandes: Bewerten wir eine Idee oder eine Umsetzung?
- Bewerten wir gegen die ursprünglichen Ziele oder gegen aktuelle?



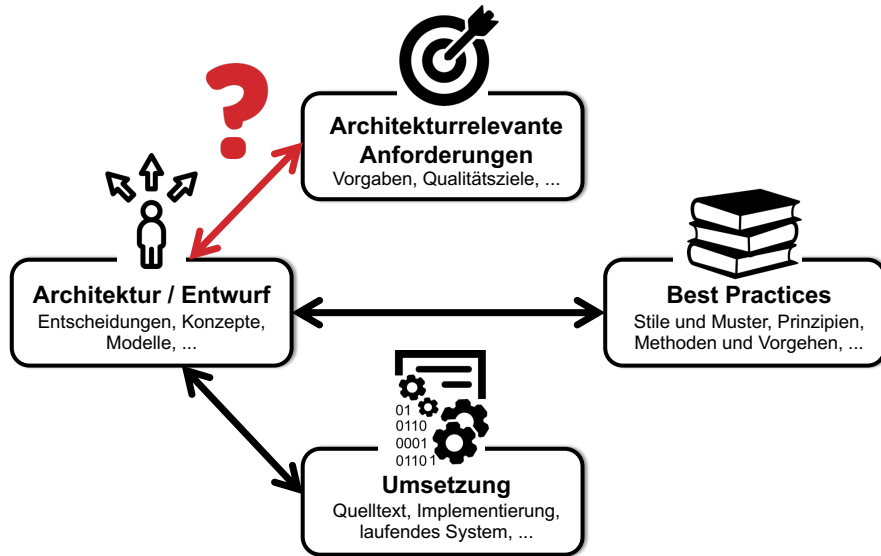
Gegeneinanderhalten ...



Gegeneinanderhalten ...



Gegeneinanderhalten ...

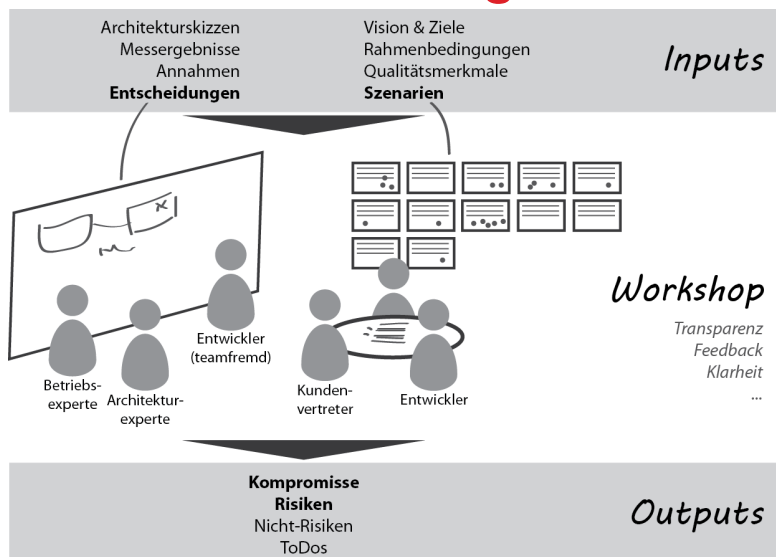


S. Zörner: "Alles Wichtige über Softwarearchitektur"

embarc.de

43

Qualitative Bewertung in einem Workshop



Quelle der Abbildung: S. Toth, Vorgehensmuster für Softwarearchitektur, 3. Auflage, Hanser 2019

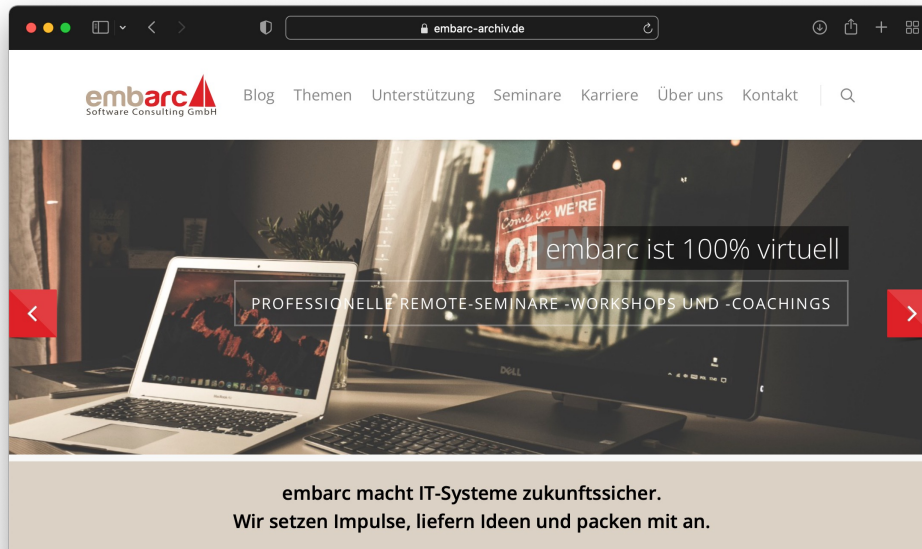


S. Zörner: "Alles Wichtige über Softwarearchitektur"

embarc.de

44

Beispiel: Die „alte“ embarc-Seite (2013-2020)



Risiken mit der alten embarc-Seite (Auswahl)

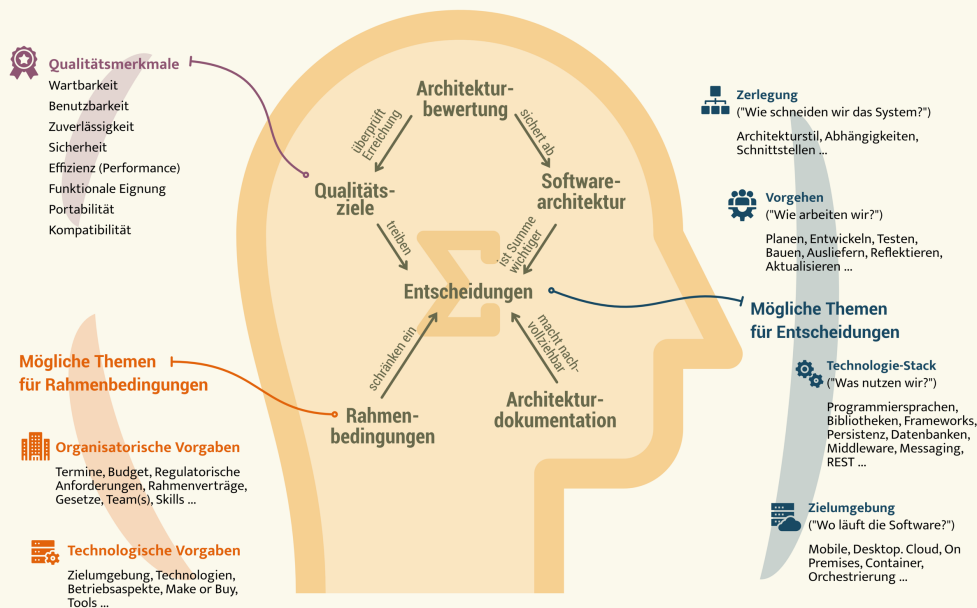
Die alte Seite hatten wir auf Basis von WordPress 2013 in kürzester Zeit gebaut und bis Anfang 2020 weiter betrieben und gepflegt.



Risiken (Probleme, die auftreten können, nicht müssen)

- Neue Theme- oder Plugin-Versionen verändern Aussehen und/oder Funktionalität der Seite
- Neues Theme erfordert kompletten Neubau der Seite
- Neues Feature zerlegt die Live-Seite, da Test-System nicht 100% in Sync und repräsentativ
- Mitarbeitende schreiben keine Beiträge, da zu umständlich (z.B. keine Offline-Fähigkeit)
- ...

EIN KOPF VOLLER SOFTWAREARCHITEKTUR



Agenda

- 1 Was (genau) ist eigentlich Softwarearchitektur?
- 2 Entscheidungen treffen
- 3 Lösungsansätze kommunizieren
- 4 Lösungen bewerten
- 5 **Fazit und Weitere Informationen**

5



TL;DR – Too long; didn't read.



Softwarearchitektur ist die Summe wichtiger **Entscheidungen**. Die gibt es immer, unabhängig davon, ob Du und Dein Team sie explizit treffen, oder es einfach so passiert.

Bei den Entscheidungen geht es darum die **Rahmenbedingungen** einzuhalten und die **Architekturziele** zu erreichen – ggf. mit **Kompromissen**.

Architekturdokumentation macht Eure Entscheidungen **nachvollziehbar** und leichter erklärbar. **Architekturbewertung** sichert sie ab und zeigt **Risiken** und Kompromisse auf.



S. Zörner: "Alles Wichtige über Softwarearchitektur"

embarc.de

49

Folien als PDF zum Download



→ embarc.de/download/



50

Artikel-Reihe in IT-Spektrum

Alle zwei Monate ein neues Porträt seit Ausgabe 04 | 2022 ...



Zeitversetzt auch online im embarc-Blog



Architektur-Porträt: Die Programmiersprache Go

Zielsetzung, zentrale Ideen und Konzepte



Architektur-Porträt: Corona-Warn-App

Architekturziele und zentrale Lösungsansätze



Architektur-Porträt: Das Spring Framework

Architekturziele und zentrale Lösungsansätze

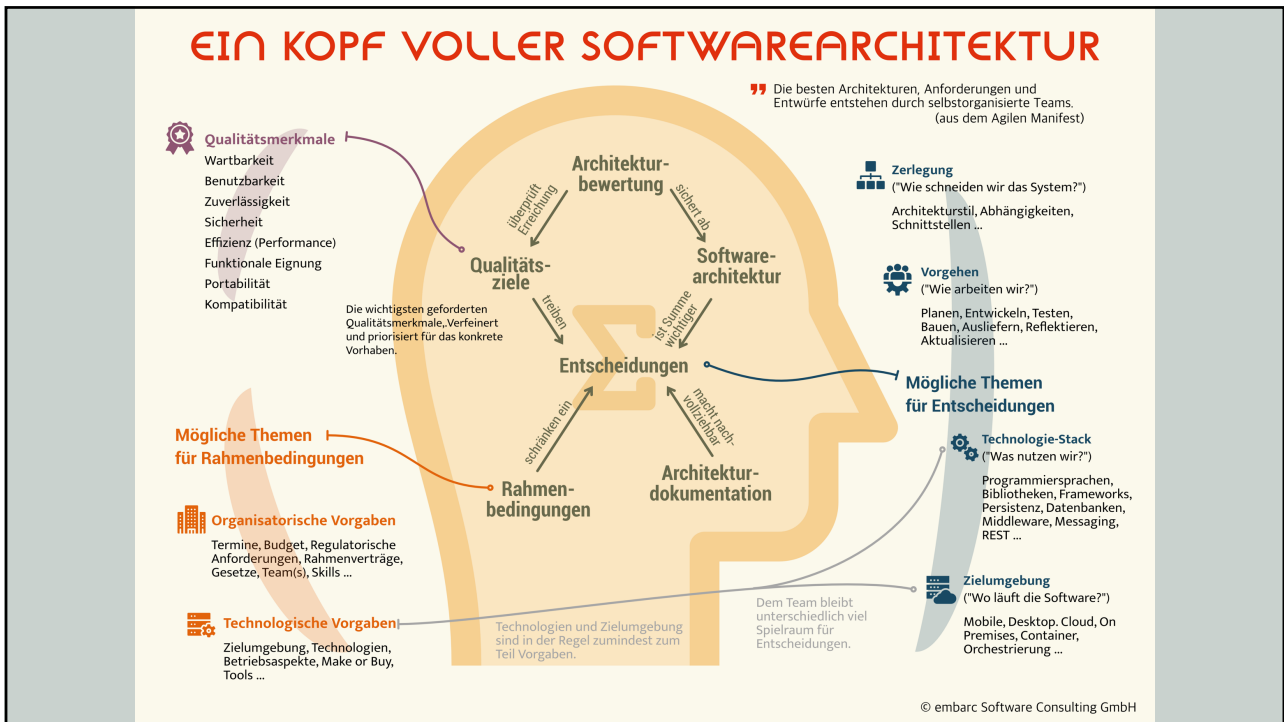


Architektur-Porträt: Visual Studio Code

Architekturziele und zentrale Lösungsansätze

➔ embarc.de/architektur-portraits/





Blog-Beitrag zu Concept-Maps

→ embarc.de/kopf-voller-softwarearchitektur-begriffsbild/



S. Zörner: "Alle

Architekturdokumentation als Textwüste? Die zentralen Lösungsideen Eures Softwarevorhabens könnt Ihr auch mit einer

Vielen Dank.

Ich freue mich auf Eure Fragen!

embarc 

- ✉ Stefan.Zoerner@embarc.de
- 🐦 [@StefanZoerner](https://twitter.com/StefanZoerner)
- in [linkedin.com/in/stefan-zoerner](https://www.linkedin.com/in/stefan-zoerner)
- X [xing.to/szr](https://www.xing.to/szr)
- 🐙 [@StefanZoerner@mastodon.social](https://mastodon.social/@StefanZoerner)



 **SOCREATORY**
The Software Creators' Academy

EINE KOLLABORATION VON

INOQ 

Angemessene Dokumentation unterstützt Dich im Austausch mit Deinem Team und gegenüber Dritten. Aber **ballastfreie Architekturüberblicke ohne Firlefanz** – gibt es nicht? Oder doch?! Was gehört hinein (und was nicht)? Wie fertigst Du einen an? Nutze Praxistipps, Erfahrungsaustausch und echte Beispiele in unserem **iSAQB® CPSA-A Modul ADOC** mit **Stefan Zörner!**



Teilnehmerstimmen:
„Der Referent holt die Teilnehmer sehr gut ab und schafft es, ein trockenes Thema spannend und kurzweilig rüber zu bringen.“



Nächster Termin
in München:
04.-05. Mai 2023
socreatory.com

Lernen von den Besten.

