

Walking Skeleton

Softwarearchitektur ohne Fleisch?

Falk Sippach



01.04.2025

embarc

0



embarc.de

Abstract

Walking Skeleton - Softwarearchitektur ohne Fleisch?

In den agilen Vorgehensweisen wird Softwarearchitektur iterativ inkrementell im Rahmen der Architekturvision entworfen und weiterentwickelt. Dabei empfiehlt sich ein risikobasiertes Vorgehen. Es wird versucht, die risikoreichen Ansätze möglichst früh zu verproben um so wertvolles Feedback zu sammeln und zeitnah nachjustieren zu können. Mit dem Walking Skeleton hat Alistair Cockburn dazu ein Konzept vorgestellt. Es umfasst es alle grundlegenden Elemente, die für die Entwicklung, die Auslieferung und den Betrieb eines Softwaresystems erforderlich sind. Dazu zählen neben der Entwicklungs- und Betriebsinfrastruktur auch die Integrationspunkte und grundlegende Architekturkonzepte.

Wir diskutieren die Vorteile und die Herausforderungen dieses Patterns und erarbeiten die Unterschiede zu einem MVP (Minimum Viable Product), Prototyp sowie Durchstich. Außerdem betrachten wir das Zusammenspiel mit anderen Vorgehensmustern in der agilen Architekturarbeit wie Architekturvision, letzter vernünftiger Moment und qualitative Tests.

1

1

Falk Sippach

embarc.de

- Softwarearchitekt, Berater, Trainer bei embark
- früher bei Orientation in Objects (OIO), Trivadis

Schwerpunkte

- Architekturberatung und -bewertung
- Cloud- und Java-Technologien

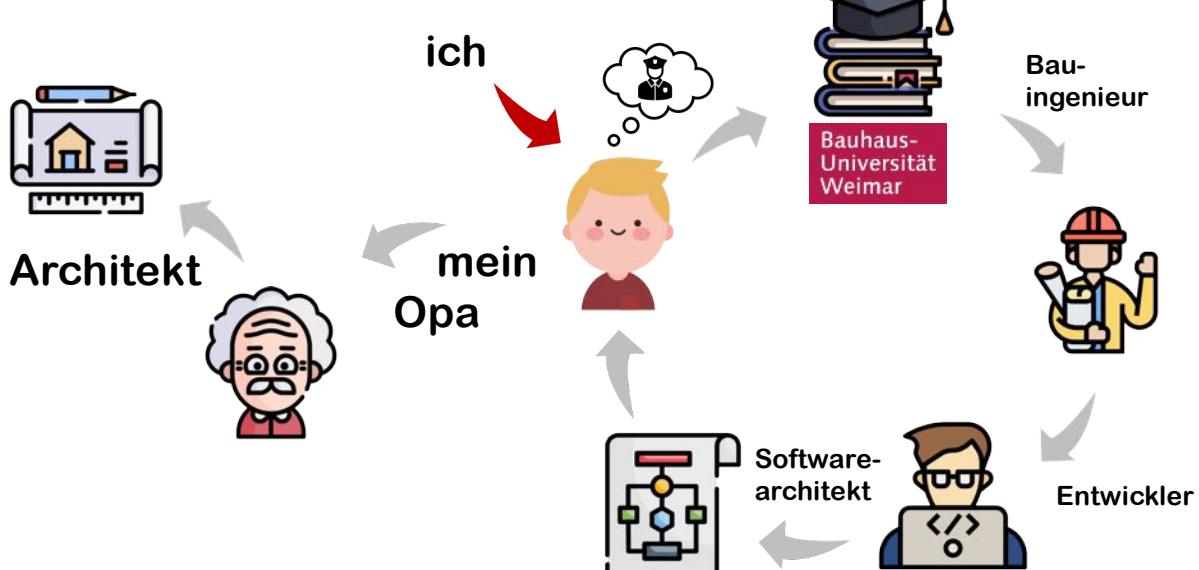
embarc 



2

embarc.de

Walking Skeleton



3

Softwarearchitektur

Eine etwas andere Definition



*“Softwarearchitektur ist die Menge der Entwurfsentscheidungen, die, wenn falsch getroffen, Dein Vorhaben **zum Scheitern bringen** kann.”**

(Eoin Woods)

* Im Englischen eigentlich: “Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled.”

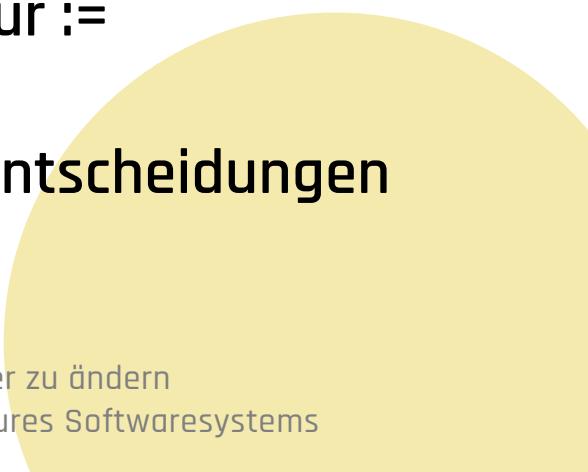
4

Was ist Softwarearchitektur?

Softwarearchitektur :=



wichtige Entscheidungen



wichtig =

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

5

Die große Herausforderung ...

Wie können wir den
Widerspruch auflösen, dass

- a) Architekturentscheidungen schwer zu ändern** sind
- b) Software**, die lange leben soll, **sich Veränderungen** im Umfeld **anpassen muss**

Was erwartet Euch in diesem Vortrag?



- Was ist **risikobasiertes Vorgehen** in der Softwarearchitektur?
- Wie können wir Softwarearchitektur **iterativ, kleinteilig und verzahnt** mit Entwicklung der **fachlichen Features** vorantreiben?
- Was sind die **Vor-/Nachteile** eines Walking Skeleton und wie unterscheidet es sich von einem **MVP** (Minimum Viable Product), **Prototyp** sowie **Durchstich**?

00.

Agenda

Was euch erwartet

Agile Architektur

Risikoorientiert

01

Walking Skeleton

02

Abgrenzung

03

Fazit & Ausblick

04

05

13

01.

Risikoorientiert



14

Cynefin-Framework

Softwareprojekte sind **heutzutage** eher **komplex**!



Aber wir versuchen nach **diesen Spielregeln** zu arbeiten!

Komplexitätstreiber: Unsicherheit

Risiken

Risiken für Projekte aus Architektursicht:

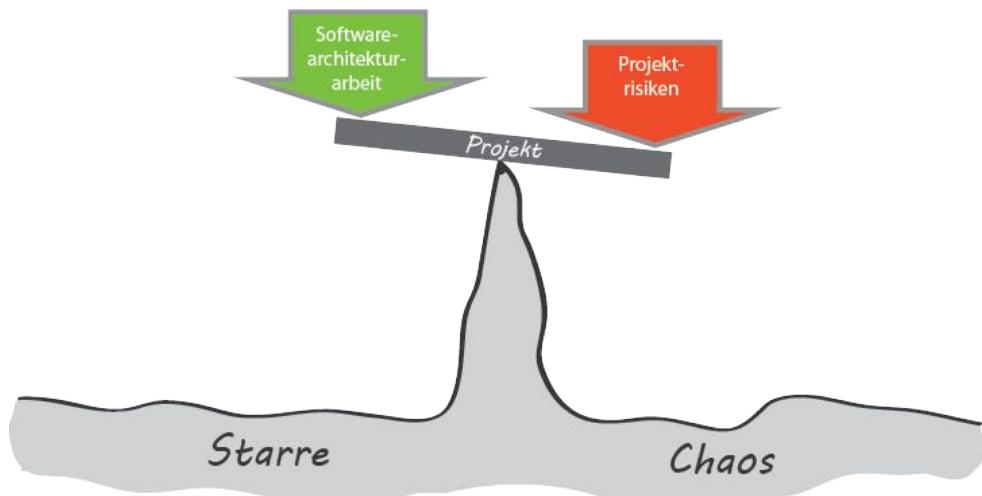
- Hohe Qualitätsanforderungen
- Enger Projektrahmen (Zeit, Budget)
- Viele Projektmitglieder
- Hoher räumlicher Verteilungsgrad
- Neue Technologien
- Wenig Erfahrung im Lösungsspektrum
- Dünner technischer Rahmen
- Viele Abhängigkeiten zu (externen) Projekten
- Vorhandene Zielkonflikte
- ...

Generell...

embarc.de

Walking Skeleton

17



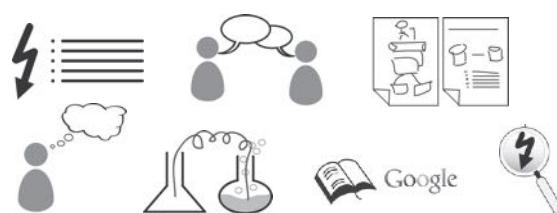
17

Architekturarbeit ist ...

embarc.de

Walking Skeleton

18



- Aufwand
- Kein Code
- Kein direkter Kundennutzen

Risikominderungsmaßnahme

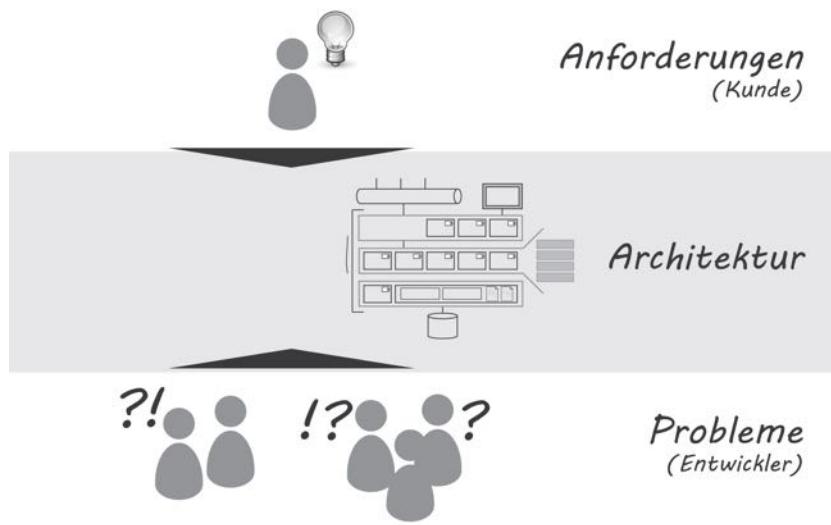
für nicht triviale Problemstellungen

Gute Prinzipien zum Umgang mit Risiken

- Fragestellungen **früh erkennen** und analysieren
- Möglichst **spät entscheiden** (bei mehr als einer sinnvollen Alternative)

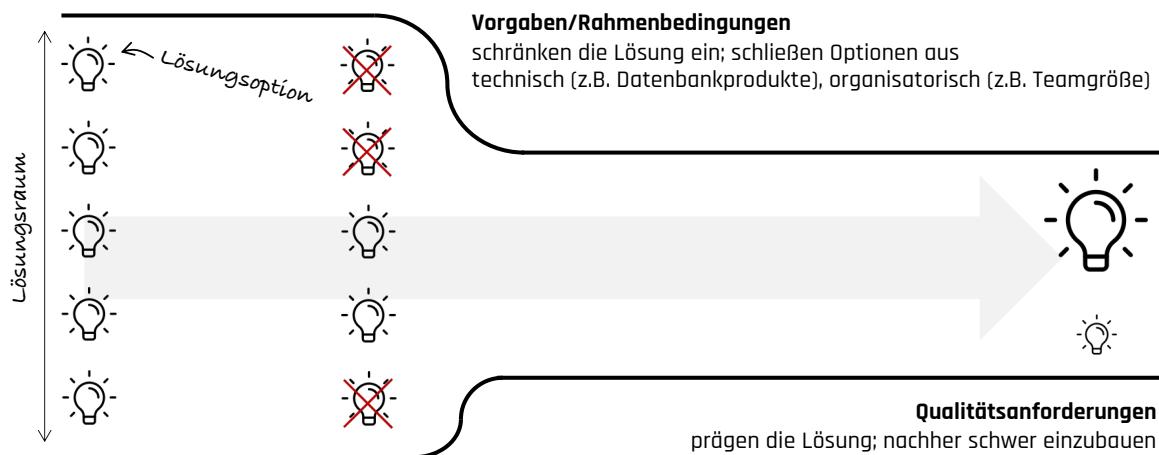
Das schafft ein großes **"Lernfenster"**
(bzw. Zeit zur Risikominderung)

Architektonische Fragestellungen

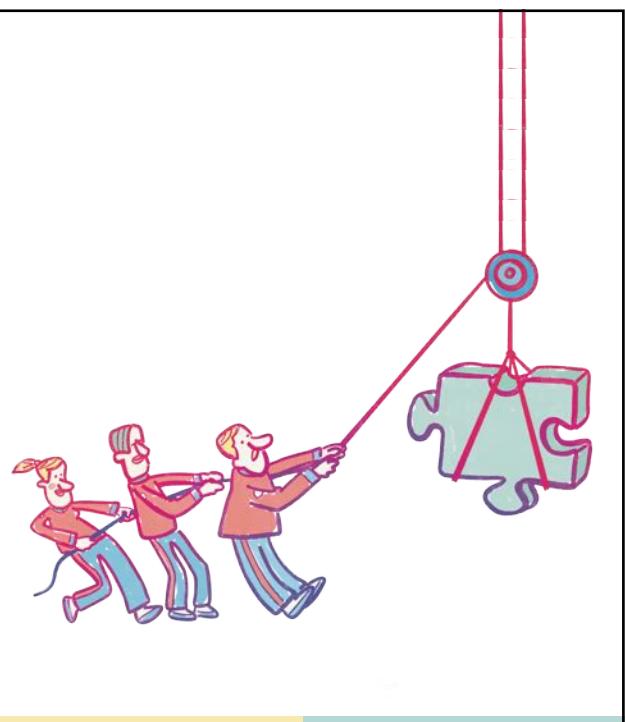


Einflüsse auf Entscheidungen

Wir stehen vor einer Architekturentscheidung ...



21



22

Was erwartet man sich von „Agil“?

- Flexibilität
- Schnellere T2M
- Nachvollziehbareren Fortschritt
- Bessere Termintreue (nicht unbedingt inhaltstreue)
- Bessere Produkte (es wird gebaut, was gewollt wird)
- Weniger “Waste”
- ...



Das agile Manifest

Individuen und Interaktion	vor	Prozessen und Werkzeugen
Funktionierende Software	vor	Umfassender Dokumentation
Zusammenarbeit mit dem Kunden	vor	Vertragsverhandlungen
Reagieren auf Veränderung	vor	Befolgen eines Plans

Das agile Manifest umgedreht

Prozesse und Werkzeug **für** Individuen und Interaktion

UNnützliche Dokumentation **für** Funktionierende Software

Vertragsverhandlungen **für** Zusammenarbeit mit dem Kunden

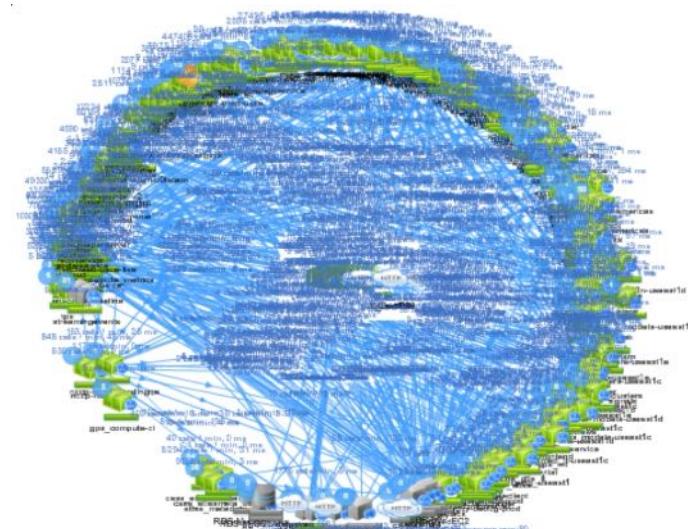
Ausreichend Planung **für** Reagieren auf Veränderung

Ref

Arbeit in komplexen Umfeldern



Netflix Architekturüberblick (?)



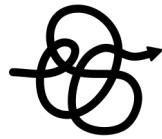
Einige Netflix Ingenieure zitiert...

“[...] we quickly see that a distributed system of any meaningful size becomes too complex for a human [to understand]. There are simply too many parts, changing and innovating too quickly, interacting in too many unplanned and uncoordinated ways for a human to hold those patterns in their head.”

"The same is true in other complex systems, including monoliths (usually with many, often unknown, downstream dependencies) that become so large that no single architect can understand the implications of a new feature on the entire application."

Aus "Chaos Engineering" von Casey Rosenthal, Lorin Hochstein, Aaron Blohowiak, Nora Jones, Ali Basiri

Agile Weltsicht / Erkenntnis



Softwareentwicklung ist
komplexes Problem



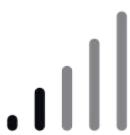
Interaktion, Feedback &
Zusammenarbeit



Werkzeuge sind
kein Selbstzweck



Probleme sind Quelle für
Wettbewerbsvorteil



Akzeptable statt
die beste Lösung

Agile Prinzipien



Inkrementelle Entwicklung zur
Verifikation von Annahmen



Risiken früh
bearbeiten



Kleine, fokussierte
Entwicklungsschritte

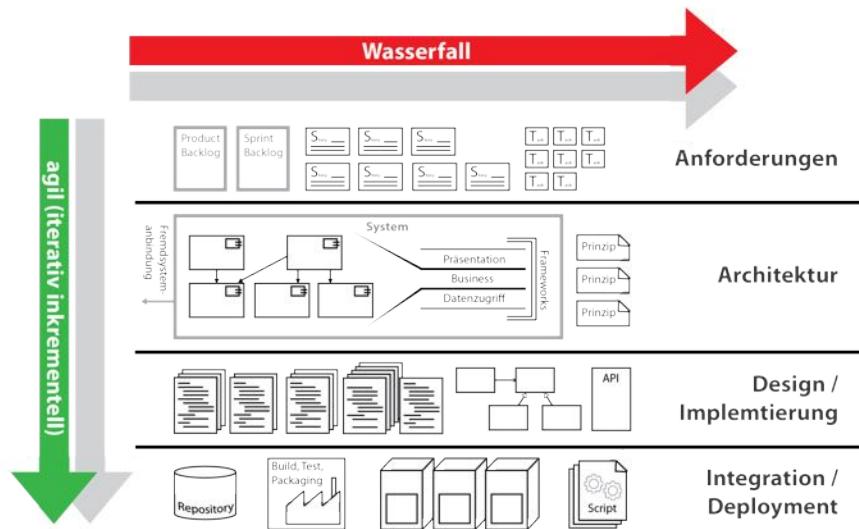


Entscheidungen zum
LVM treffen



Risikobearbeitung, Wissens-
aufbau und Wertschöpfung

Was macht Agile Architektur anders?



Agile Architektur



Architekturaufwand dem Problem **angemessen**

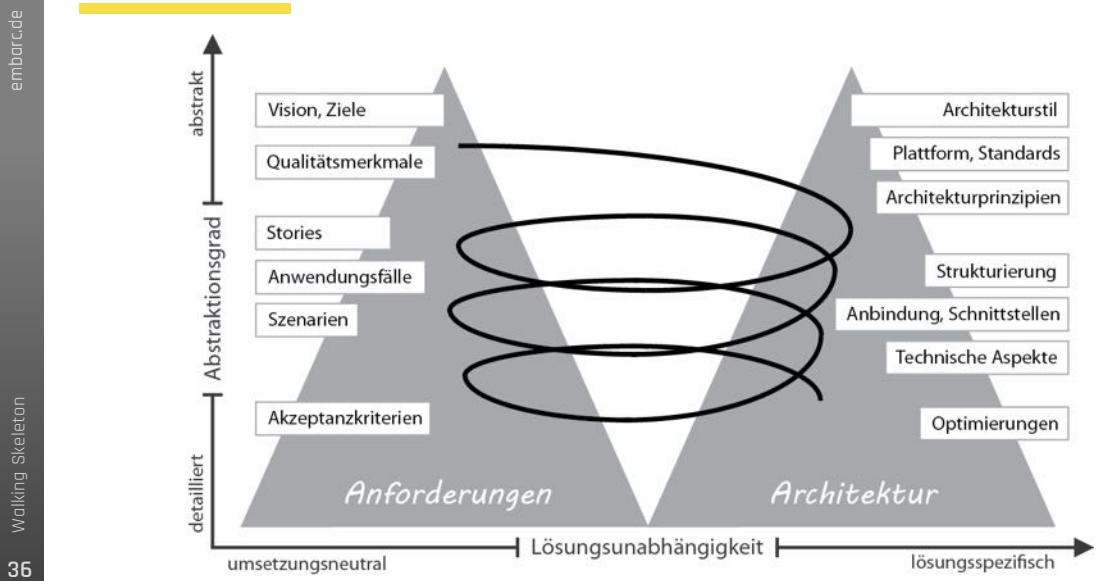


Architektur entsteht **iterativ** (kein **BUFD**)



Crossfunktionale Teams (transparent/reflektiert)

Iterative Architekturarbeit



36



37

”

 statt **Big Design Up Front**

Eine Architekturvision ist eine schlanke, sich stetig weiterentwickelnde Übersicht zur aktuellen Architekturidee und deren Motivation.



Stefan Toth

Architekturvision

Architekturvision

Fokus für den Projekterfolg kritische Architekturaspekte für die Entwicklung unumstößlich wichtiger Rahmen

Typische Inhalte

- ▶ **Systemkontext** (Abgrenzung)
- ▶ **Rahmenbedingungen**
- ▶ **Qualitätsanforderungen** (priorisiert)
- ▶ **Technologien** (inkl. Plattform, Programmiersprache)
- ▶ **Architekturstil** (bestimmende Muster)
- ▶ **Architekturprinzipien**
- ▶ **Große Gliederung** (Systemteile auf oberster Ebene)

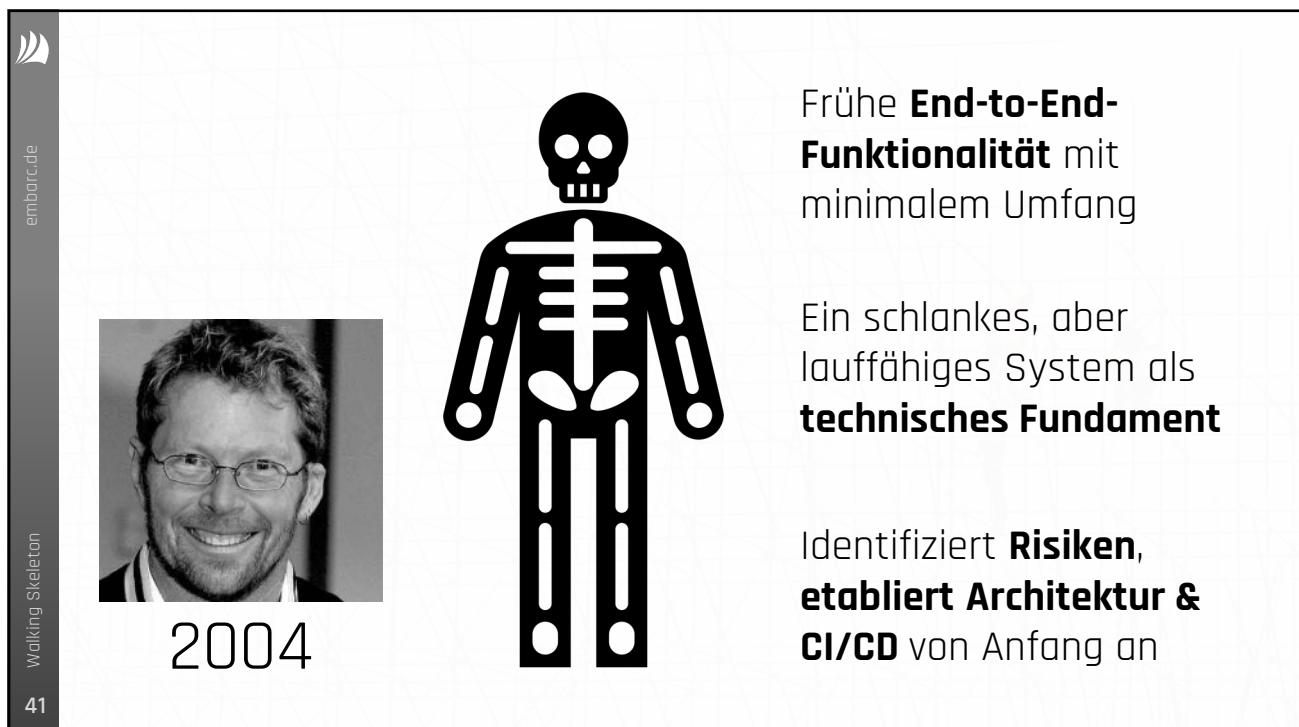
Was?

Wie?

Als Idee, nicht als irreversible Entscheidung!



40



Walking Skeleton

2004

Frühe **End-to-End-Funktionalität mit minimalem Umfang**

Ein schlankes, aber lauffähiges System als **technisches Fundament**

Identifiziert **Risiken**, etabliert **Architektur & CI/CD** von Anfang an

41

41

Walking Skeleton



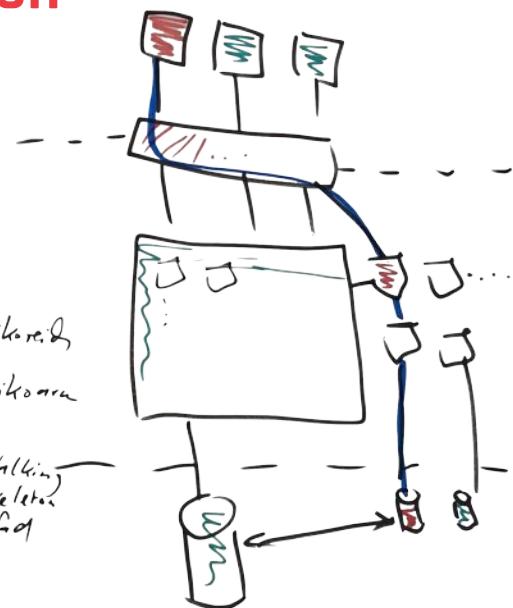
„A **Walking Skeleton** is a tiny implementation of the system that performs a small end-to-end function. It need **not** use the **final** architecture, but it should **link together** the **main architectural components**. The architecture and the functionality can then evolve in parallel.“

- Alistair Cockburn

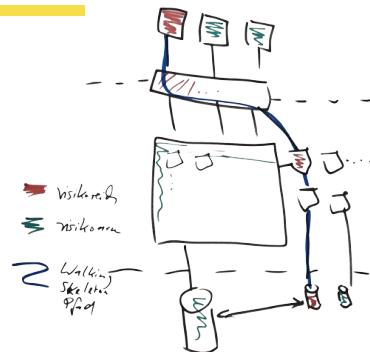
Walking Skeleton in Aktion

Die **momentan beste Idee** möglichst **früh** widerlegen!

Risiken
früh angehen!
kein MVP,
kein Durchstich/Prototyp



Möglichst effektiv beweisen dass es Müll ist.



Möglichst schnell **widerlegen**. Und zwar richtig.

- In Risiken stechen
- Unsicherheiten ausnutzen
- zentrale Ziele in Gefahr bringen

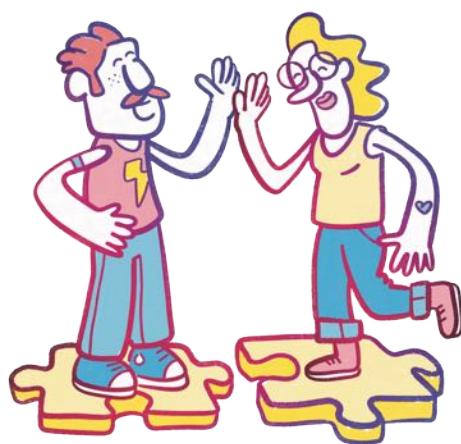


Entsprechende Anforderungen suchen

44

04.

Abgrenzung



45

Abgrenzung zu ...

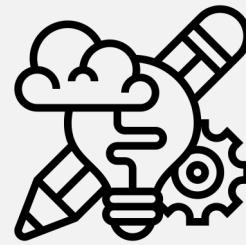
embarc.de

Walking Skeleton

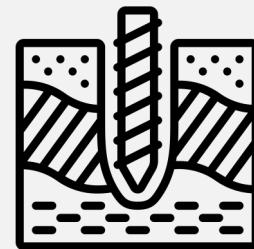
46



MVP



Prototyp



Durchstich

46



Architektur-vision



Walking Skeleton



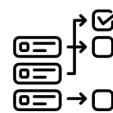
Architektur-überblick



Umgang mit Risiken



Letzter vernünftiger Moment



Architekturarbeit sichtbar machen



Technische Schulden



Auf den Prüfstand stellen

Walking Skeleton

50

50

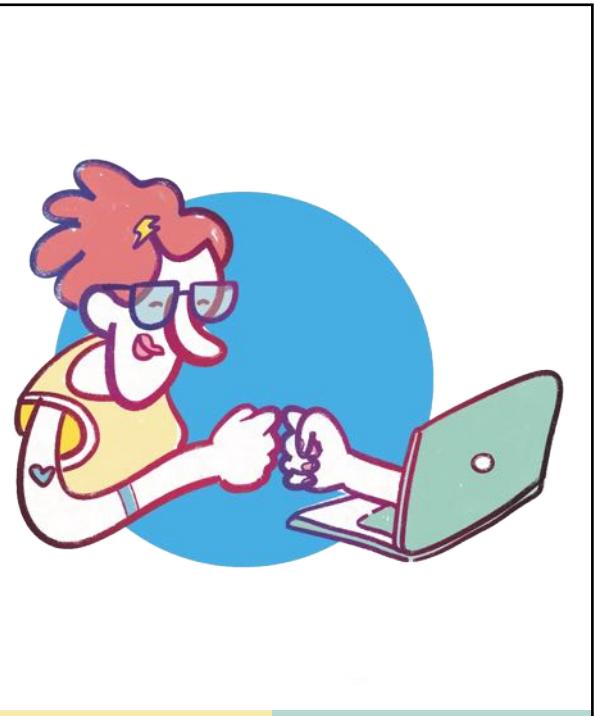
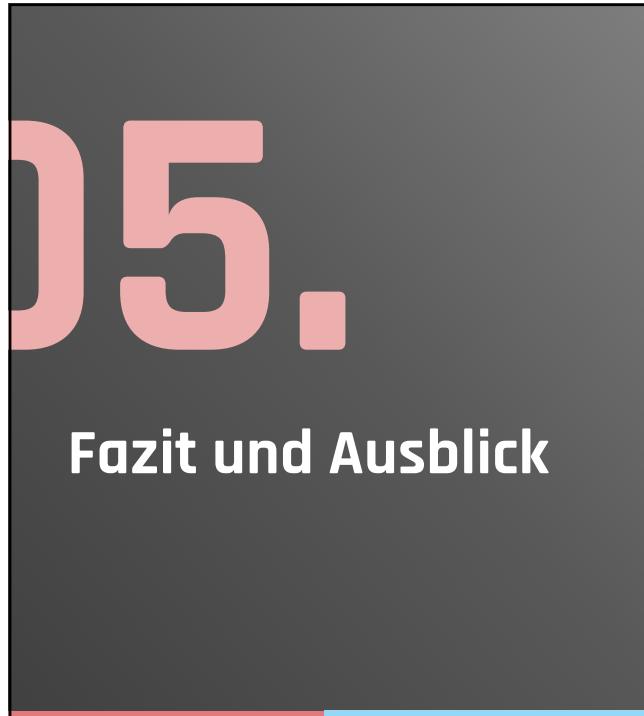
04/2025

Vorgehensmuster für Softwarearchitektur

Kombinierbare Praktiken in Zeiten von Agile und Lean

Stefan Toth

4. Auflage
Hanser Verlag



Wie hilft uns das Walking Skeleton?

Frühes Feedback & Risikominimierung

Iterative Entwicklung & Evolutionäre Architektur

DevOps & Automatisierung von Anfang an

Klare Kommunikation & Transparenz

Architektur-Spicker



„Mit unseren Architektur-Spickern beleuchten wir die konzeptionelle Seite der Softwareentwicklung.“

Architektur-Spicker #6
Agile Architektur



PDF, 4 Seiten
Kostenloser Download.

→ embarc.de/architektur-spicker

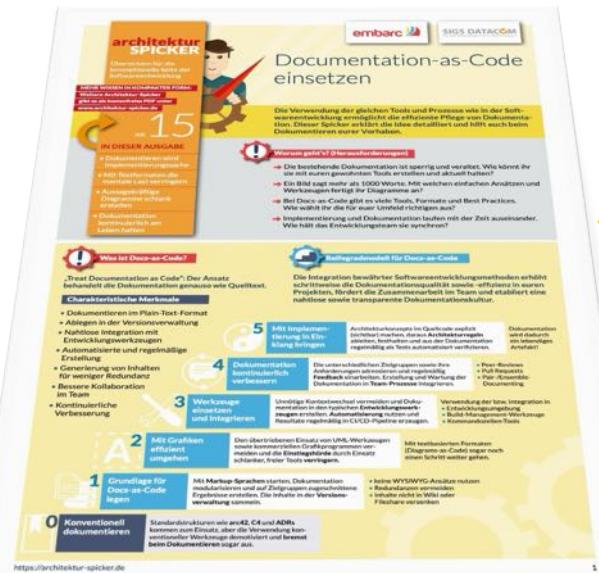
Architektur-Spicker

NEU

embarc.de

Walking Skeleton

59



„Mit unseren Architektur-Spickern beleuchten wir die konzeptionelle Seite der Softwareentwicklung.“

Architektur-Spicker #15 Documentation-as-Code einsetzen



PDF, 4 Seiten
Kostenloser Download.

→ architektur-spicker.de/

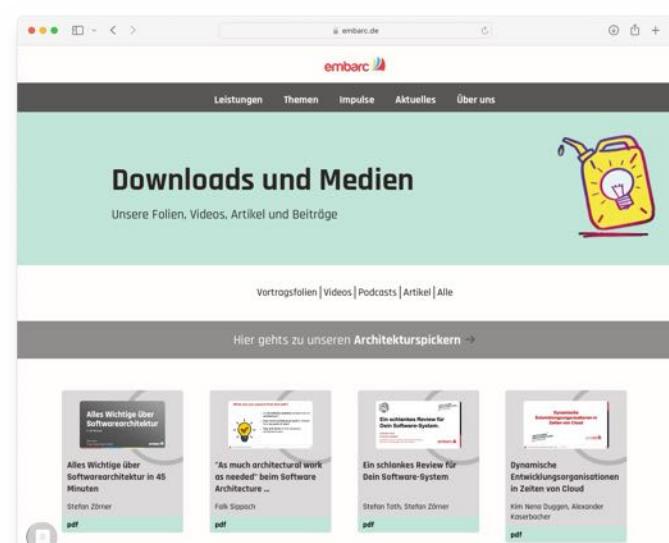
59

Folien als PDF zum Download

embarc.de

Walking Skeleton

60




→ embarc.de/download/

60

Feedback & Fragen?

Wir freuen uns auf Fragen,
Diskussionen, Anregungen!



Vielen Dank.

Ich freue mich auf Eure Fragen!

-  falk.sippach@embarc.de
-  linkedin.com/in/falk-sippach
-  @sippssack
-  @sippssack@ijug.social

