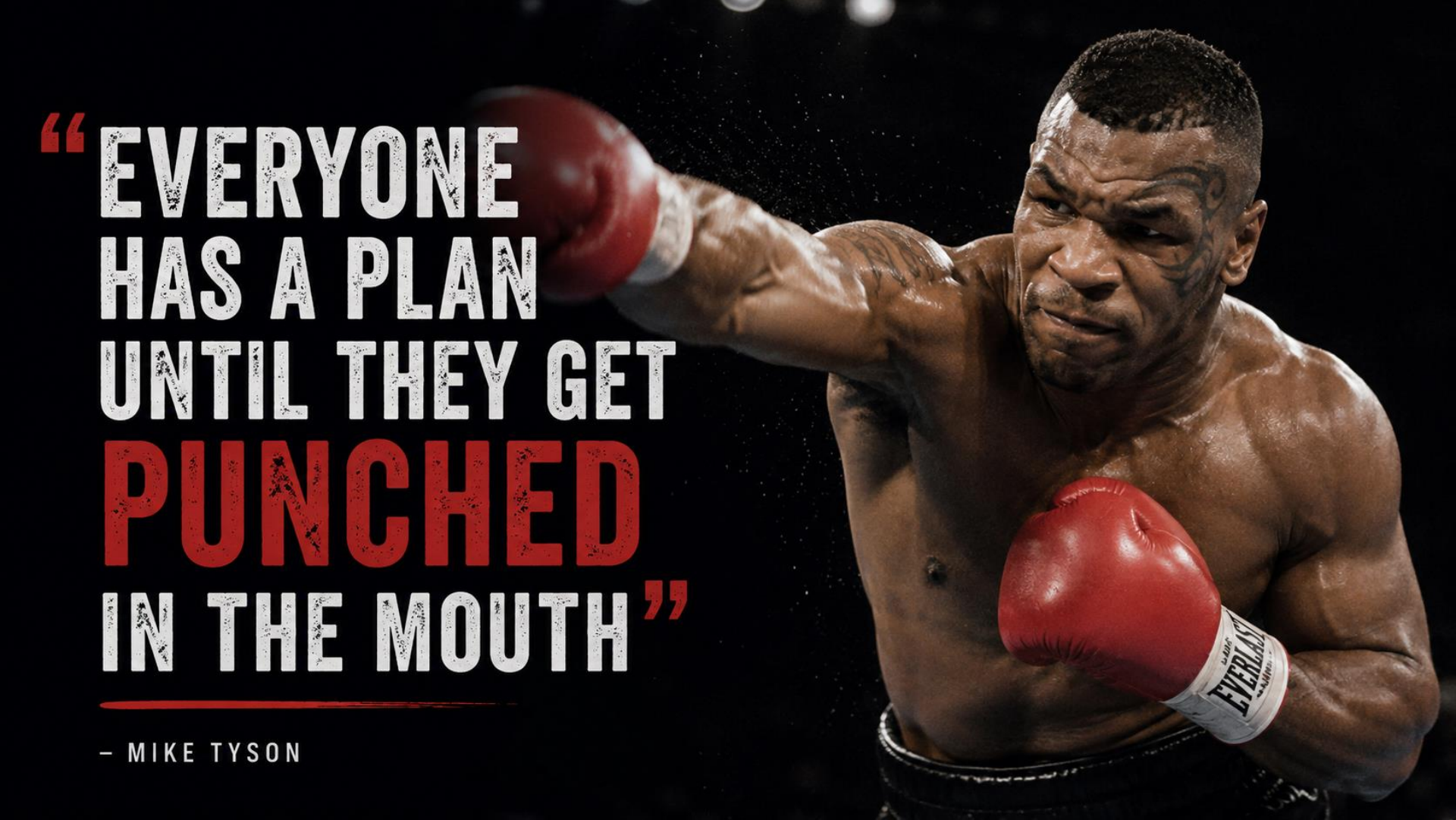
Das Manifest agiler Architekturarbeit

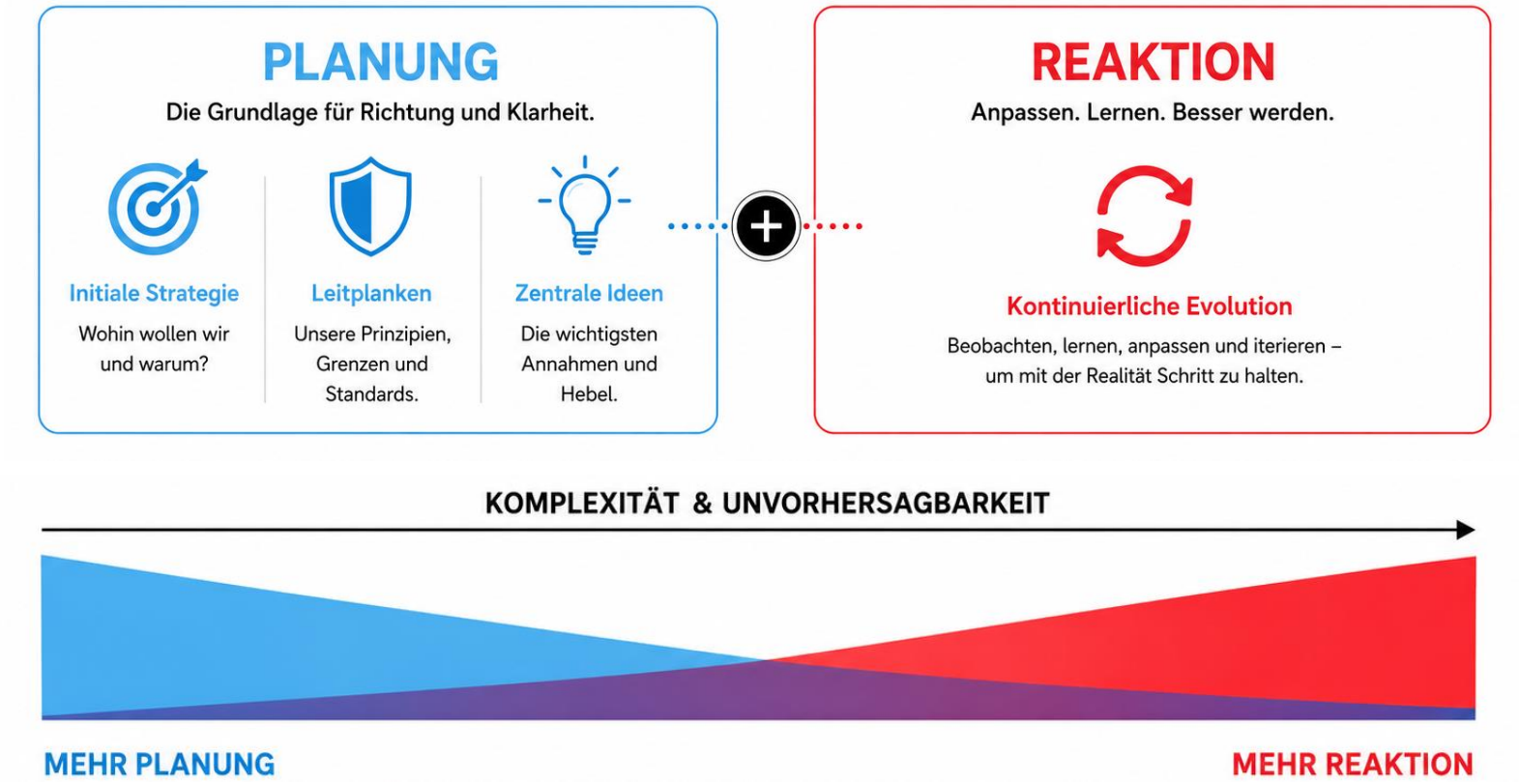
Stefan Toth

Software Architecture Forum – 16.06.2026, München


In der Eile verlieren wir am Flughafen „Eli“ – DAS Kuscheltier meines Sohnes…
40onL Apartments
to Stefan
Today at 09:41
Dear Stefan,
I'm sorry to inform you that we have renovation trouble in the apartment you booked. As a consequence, we have to cancel your booking with us. We are very sorry for any inconvenience.
Unser Urlaub in
KAPSTADT
SÜDAFRIKA
Das Taxi verspätet sich und ist dann nicht der bestellte Kombi. Unser Gepäck passt nicht in den Kofferraum…
Unser Kindersitz ist vom Flug beschädigt. Wir sind müde aber müssen reklamieren.
TAXIS
Vollkasko
GPS inklusive
Abholung am Flughafen CPT
UNSERE EIGENEN KINDERSITZE MIT
Für den Flug zugelassene
Sitze für be
CARES-Gurt Flugzeug nutzen
Am Gate mit Transporttaschen aufgeben
WETTERBERICHT
28° / 18°
30° / 19°
28° / 17°
27° / 16°
26° / 15°
Zwiebellook, Sonnencreme & bequeme Schuhe!


“EVERYONE HAS A PLAN UNTIL THEY GET PUNCHED IN THE MOUTH”
– MIKE TYSON

Jedes Unterfangen hat…



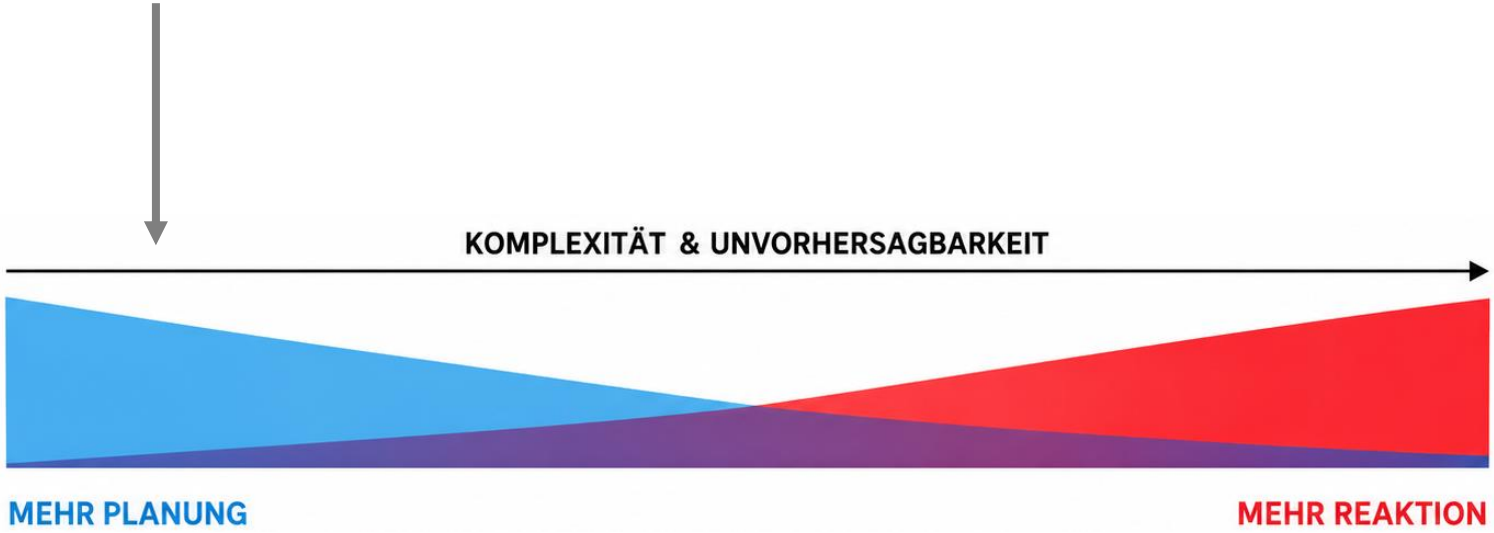
Über Software-Architektur

Sichtweise von 1990

architecture. The organizational structure of a system or component. *See also:* **component; module; subprogram; routine.**

Approved September 28, 1990

IEEE Standards Board



Was passiert in den 1990ern?

1. Sprachen & Entwicklungswerkzeuge

- Python (1991)
- Visual Basic (1991)
- Java (1995)
- JavaScript (1995)
- PHP (1995)
- Ruby (1995)
- Delphi (1995)

2. Plattformen & Systeme

- Windows NT (1993)
- Linux (1991)
- Solaris (1992)
- Windows 95 (1995)
- Windows NT 4.0 (1996)

3. Web-, Server- & Enterprise-Technologien

- Apache HTTP Server 1.0 (1995)
- IIS 1.0 (1995)
- CSS1 (1996)
- HTML 4.0 (1997)
- JDBC (1997)
- Java Servlets (1997)
- RMI (1997)
- CSS2 (1998)
- XML 1.0 (1998)
- EJB 1.0 (1998)
- JSP 1.0 (1999)
- J2EE (1999)

4. Architektur- & Verteilungskonzepte

Klare Technologie-/Standardeinführungen

- CORBA 1.0/1.1 (1991)
- COM (1993)
- DCOM (1996)

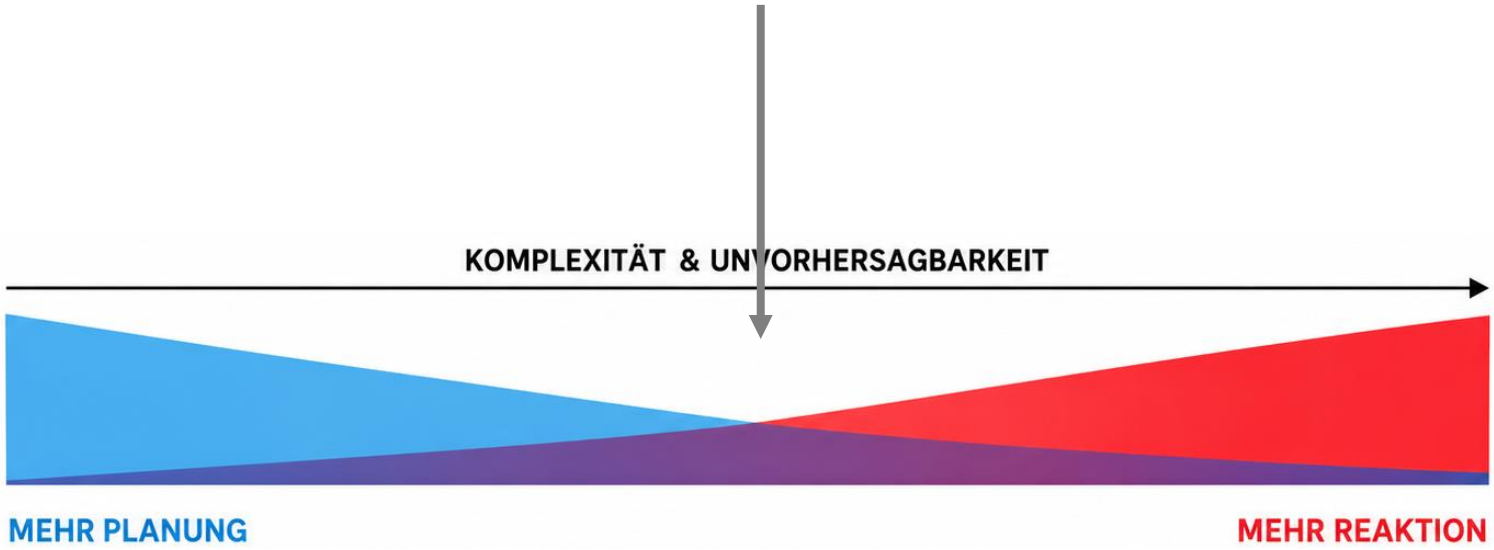
Prägend bzw. stark verbreitet in den 1990ern

- Client-Server
- 3-Tier
- N-Tier
- Component-Based Architecture
- Distributed Systems
- Middleware
- Messaging
- Thin Client

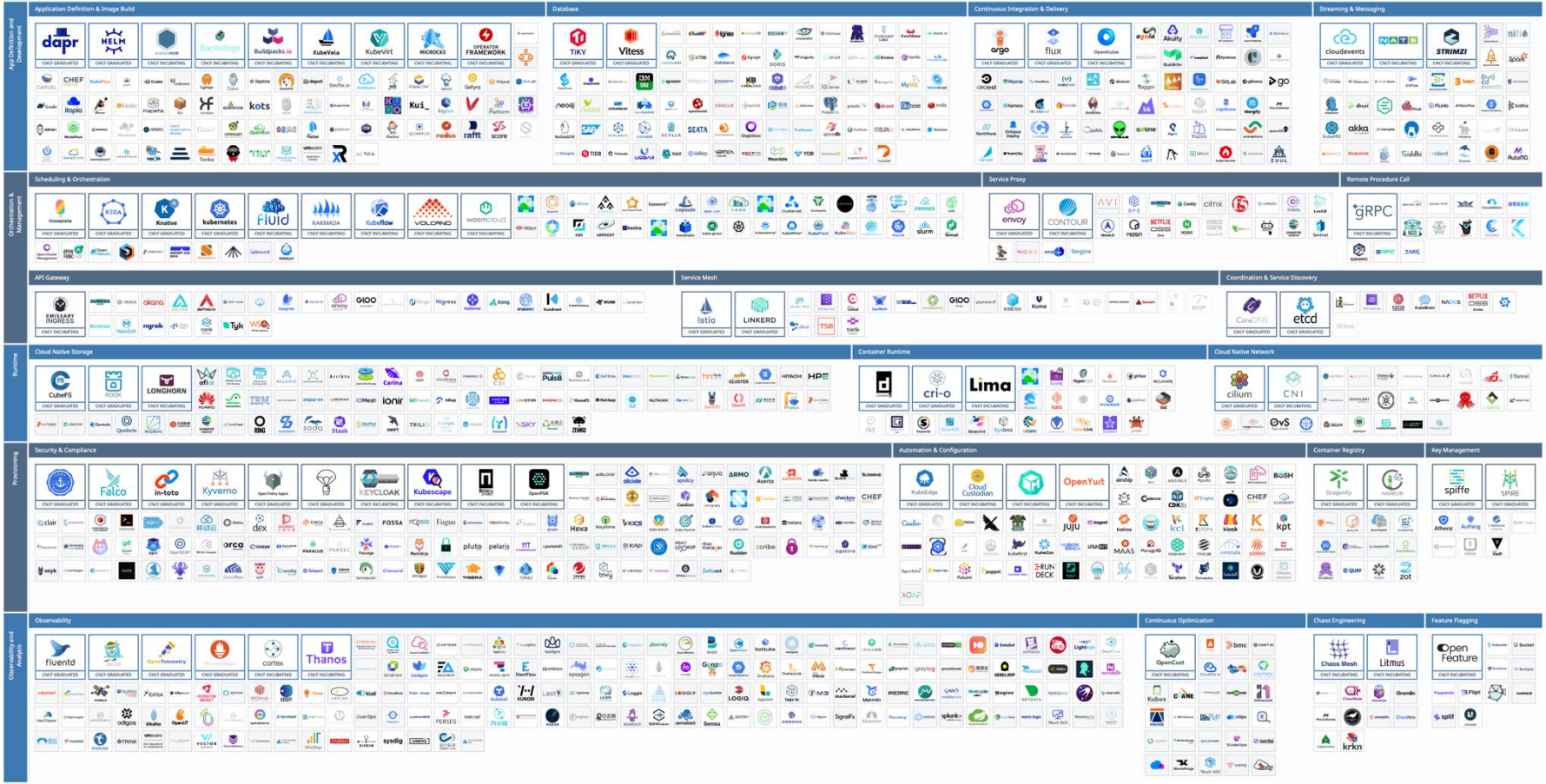
Sichtweise der 2000er Jahre

IEEE Std 1471-2000

3.5 architecture: The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.



CLOUD NATIVE LANDSCAPE

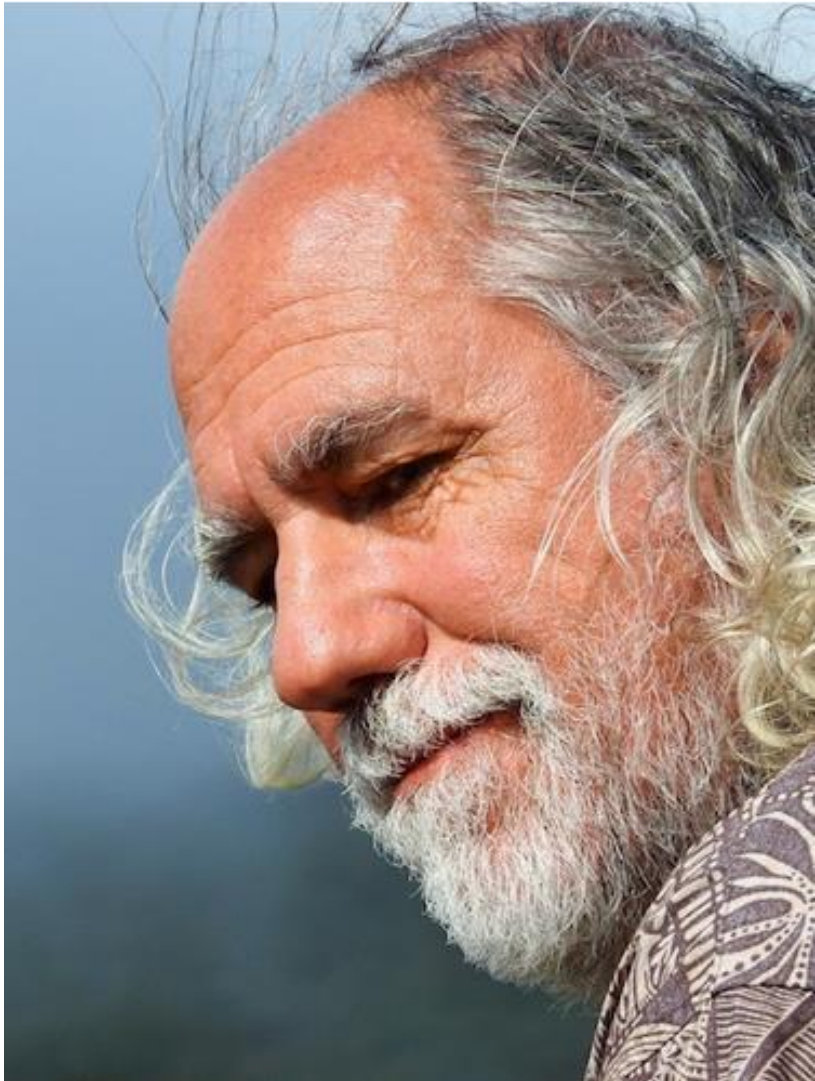


Architektur

“The journey between vision and ultimate executable system is **complex**.

[…] myriad decisions, some […] advance progress while others represent dead ends or trigger points for scrap and rework.”

(Grady Booch)



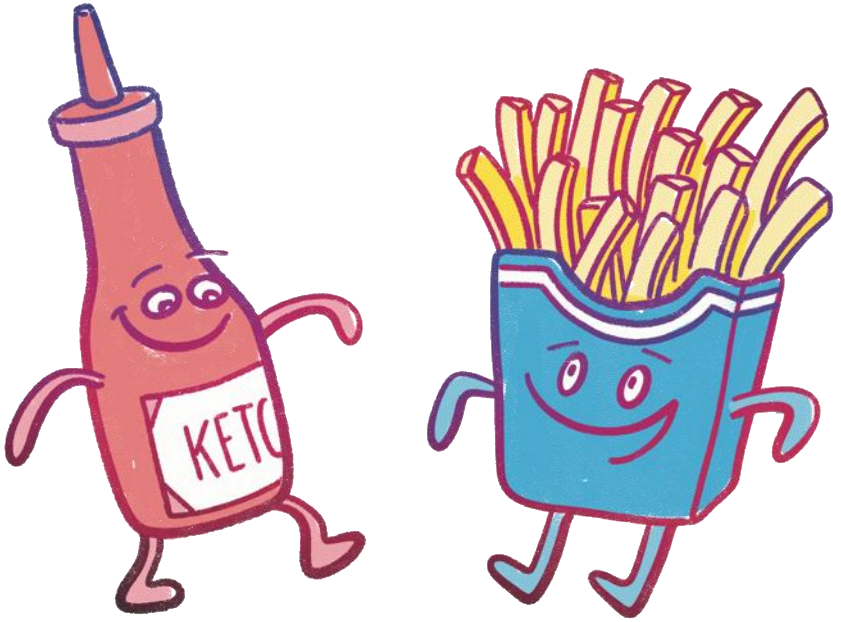
Architektur

“The life of a software architect is a long and rapid succession of suboptimal design decisions taken partly in the dark.”

(Philippe Kruchten)

Architektur und Agilität

A Perfect Match?



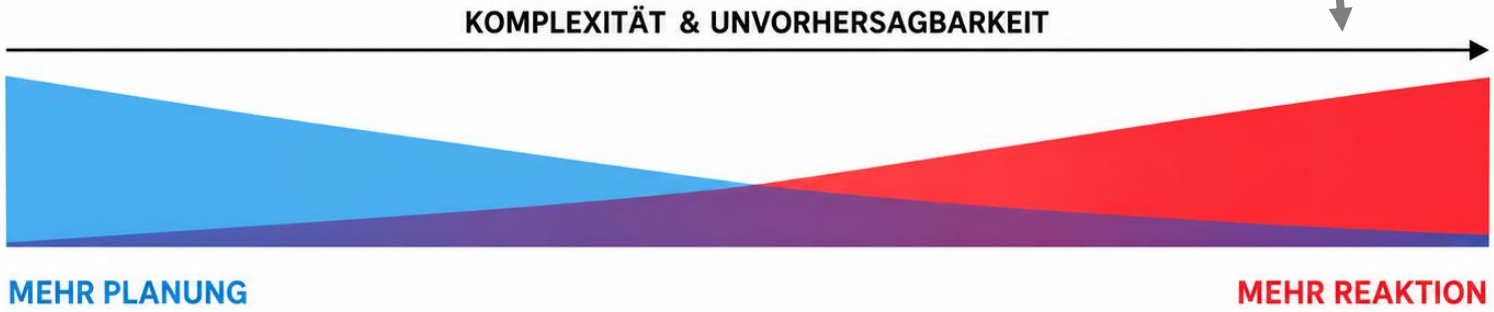
Was macht „agil“ aus?

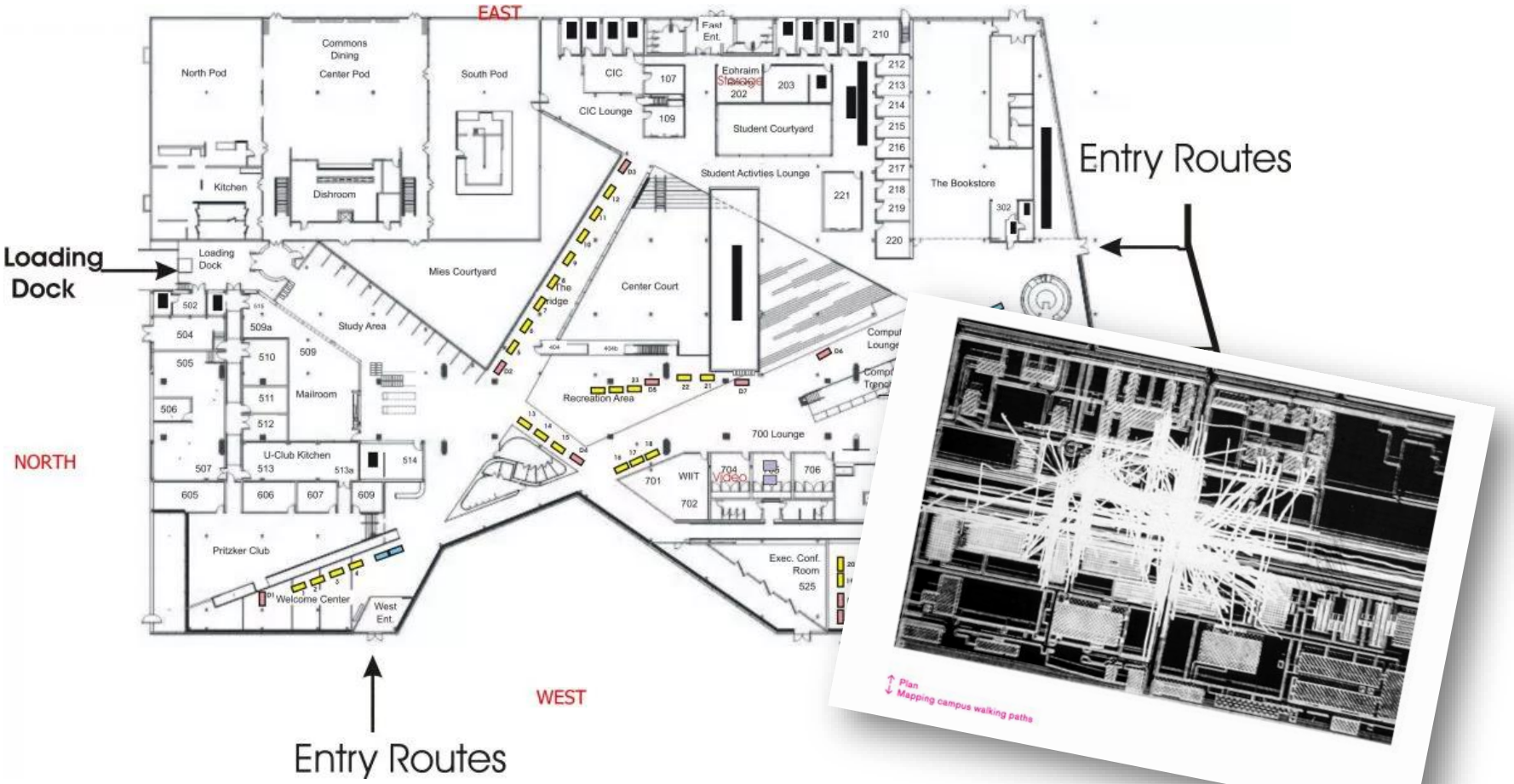
ANNAHMEN

- Es liegt keine Vollinformation vor
- Was wir wissen, wird sich ändern
- Wir lernen stetig hinzu
- Es gibt keine perfekte Lösung – nur gültige Lösungen

KERNIDEEN

- Kleine Entwicklungsschritte
- Gute Kommunikation und Transparenz sind zentral
- Risiken früh angehen
- Stetiges, reales Feedback ist wichtig
- Fehler und Probleme sind Lernchancen



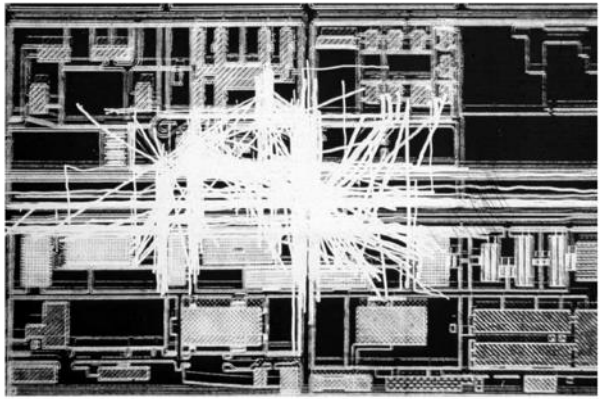


McCormick Tribune Campus Centers auf der Illinois Tech Universität

Bottom-Up…

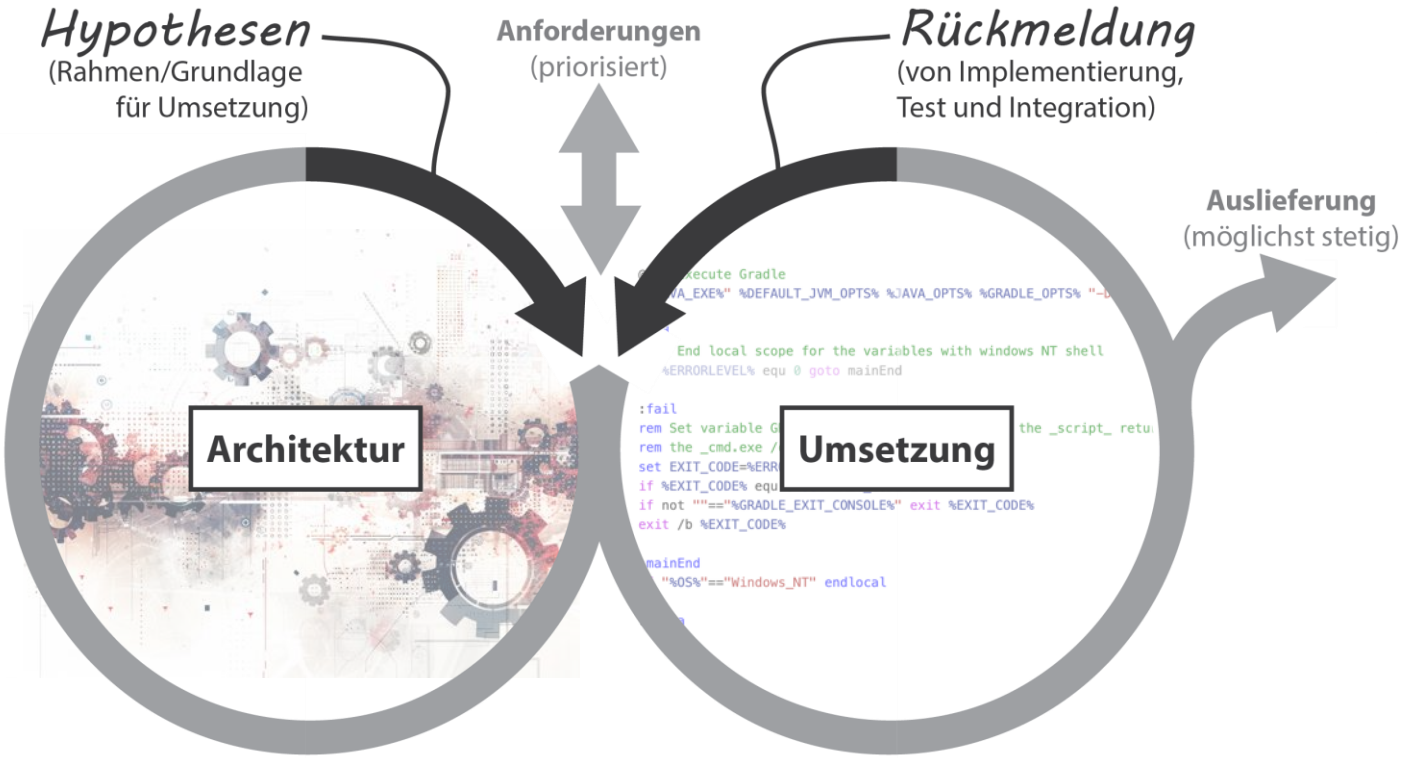
- **Ziele** von Nutzerinnen und Designer sind **homogen**
- Es gibt **Offenheit** aber auch einen **starken Rahmen**
  (Stundenpläne, Gebäudezwecke, …)

- Was wenn der Kontext anders wäre?

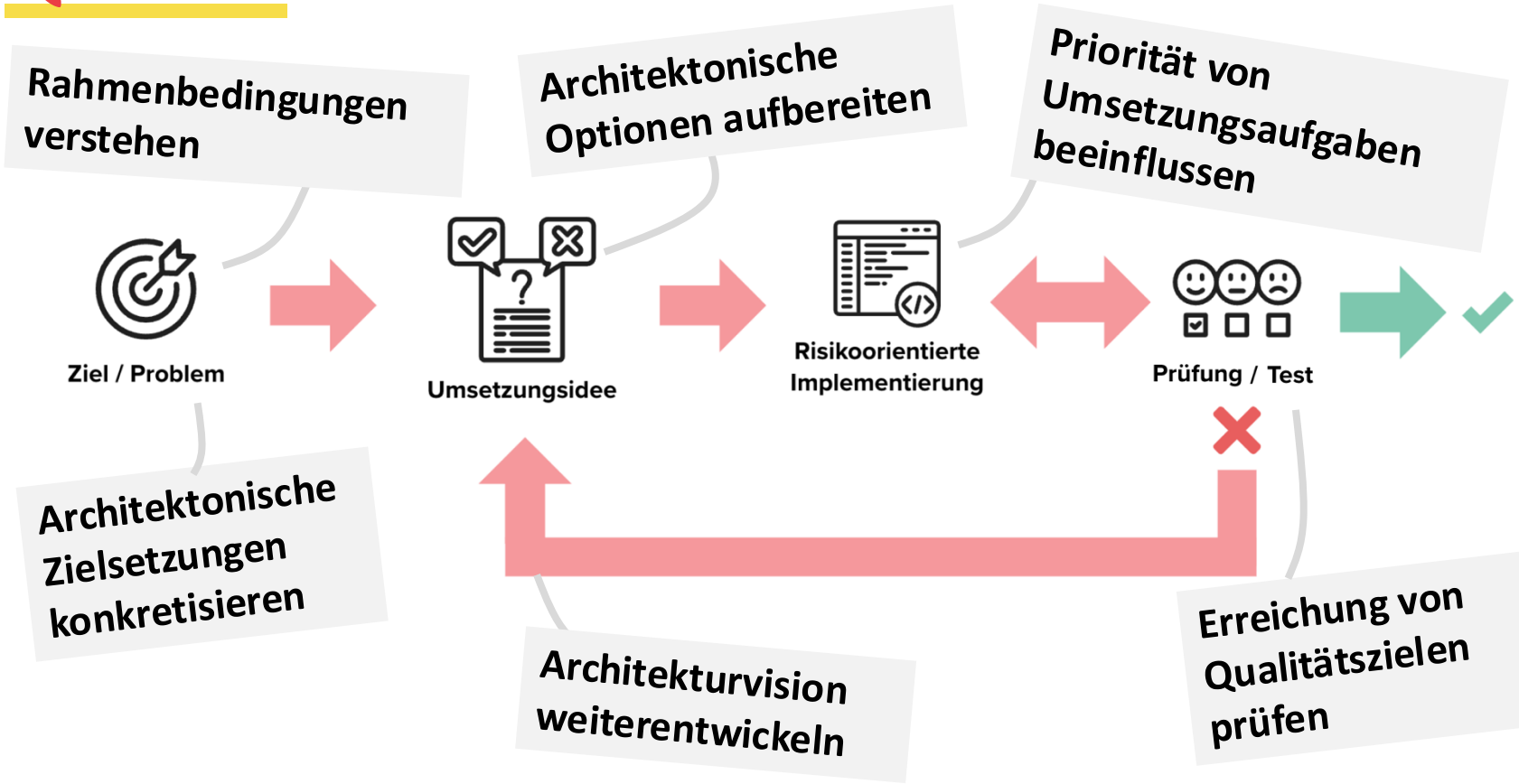


↑ Plan
↓ Mapping campus walking paths

Architektur als Entwicklungsmodus



Zielorientierung, Flow, ausreichend Qualität…



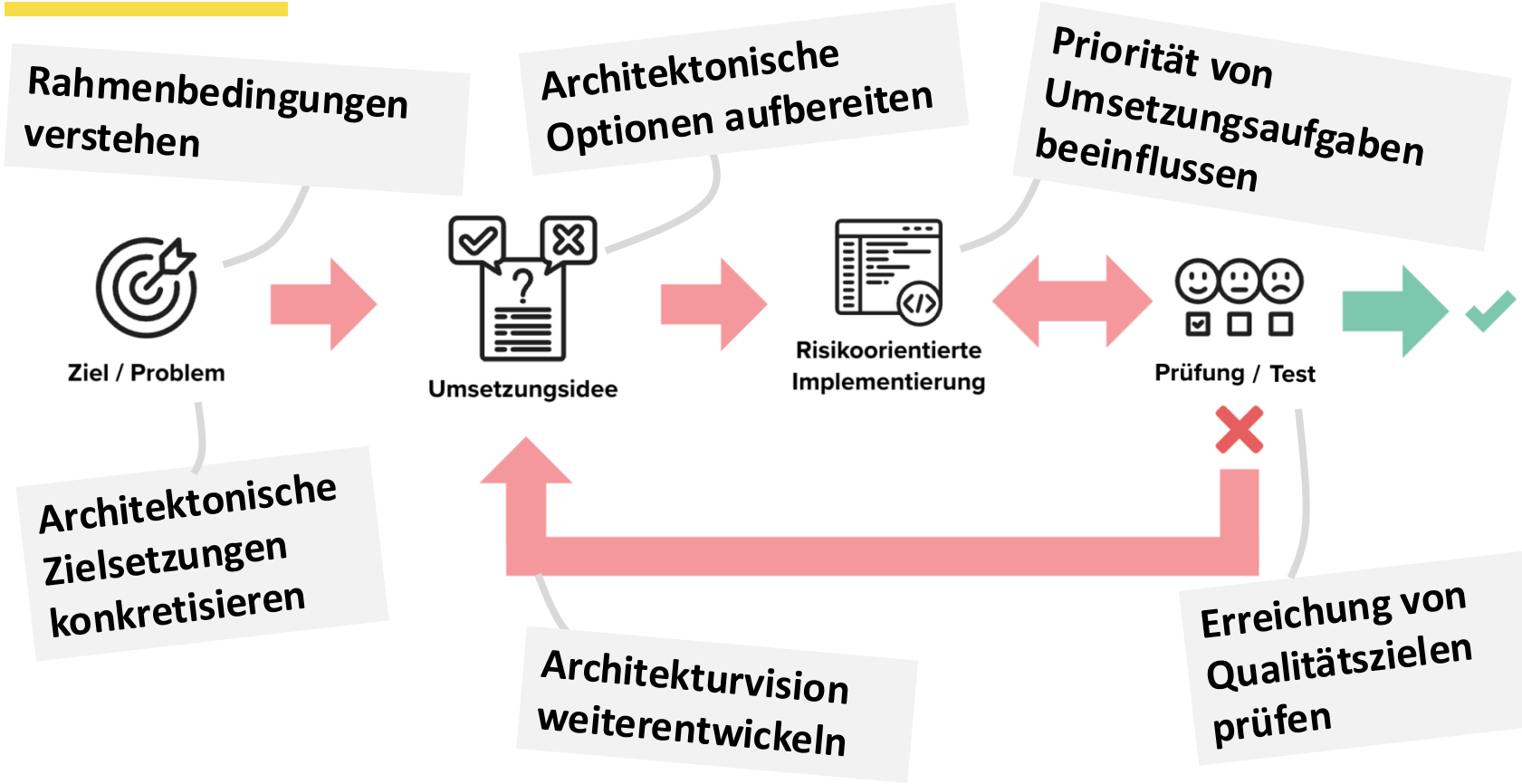
03.

Agile Software-Architektur

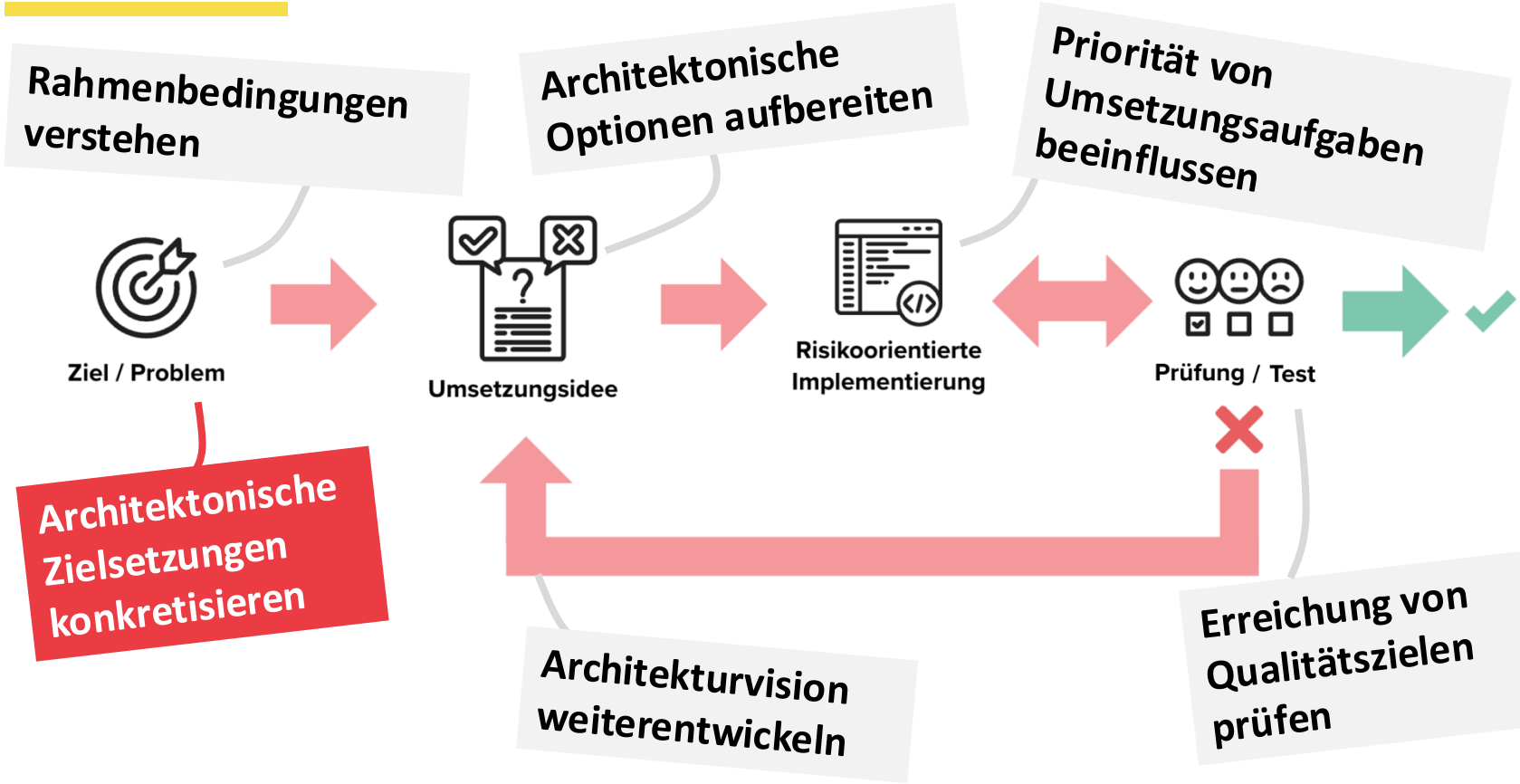
Das Manifest agiler Architekturarbeit

Kontinuierliche Verbesserung	vor	Initialer Perfektion
Zielorientierung	vor	Standardlösungen
Kollaboration	vor	Isolierter Spezialisierung
Breite Verantwortung	vor	Zentralisierung

Zielorientierung, Flow, ausreichend Qualität…



Zielorientierung, Flow, ausreichend Qualität…



Qualitätsmerkmale

Begriffe nach ISO 25010

Benutzbarkeit
(Usability)

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

Portabilität
(Portability)

Ist die Software leicht auf andere Zielumgebungen (z.B. anderes OS) übertragbar? …

Funktionale Eignung
(Functional Suitability)

Sind die berechneten Ergebnisse genau genug / exakt, ist die Funktionalität angemessen? …

Effizienz
(Performance)

Antwortet die Software schnell, hat sie einen hohen Durchsatz, einen geringen Ressourcenverbrauch? …

Kompatibilität
(Compatibility)

Ist die Software konform zu Standards, arbeitet sie gut mit anderen zusammen? …

Zuverlässigkeit
(Reliability)

Ist das System verfügbar, tolerant gegenüber Fehlern, nach Abstürzen schnell wieder hergestellt? …

Sicherheit
(Security)

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

Wartbarkeit
(Maintainability)

Ist die Software leicht zu ändern, erweitern, testen, verstehen? Lassen sich Teile wiederverwenden? …

Konkretisierung in Qualitätsszenarien

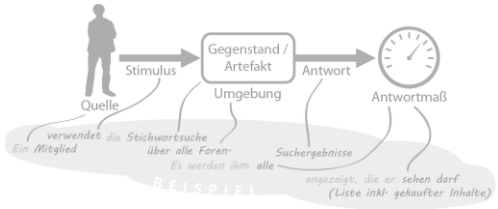
ID

z. B. W-0x

Zugeordnetes Qualitätsziel

z. B. Wartbarkeit

Szenariobeschreibung (2-3 Sätze)



Szenario-Art

- ☐ Verwendung/Use Case
- ☐ Änderung/Wachstum
- ☐ Fehler/Katastrophen/Stress

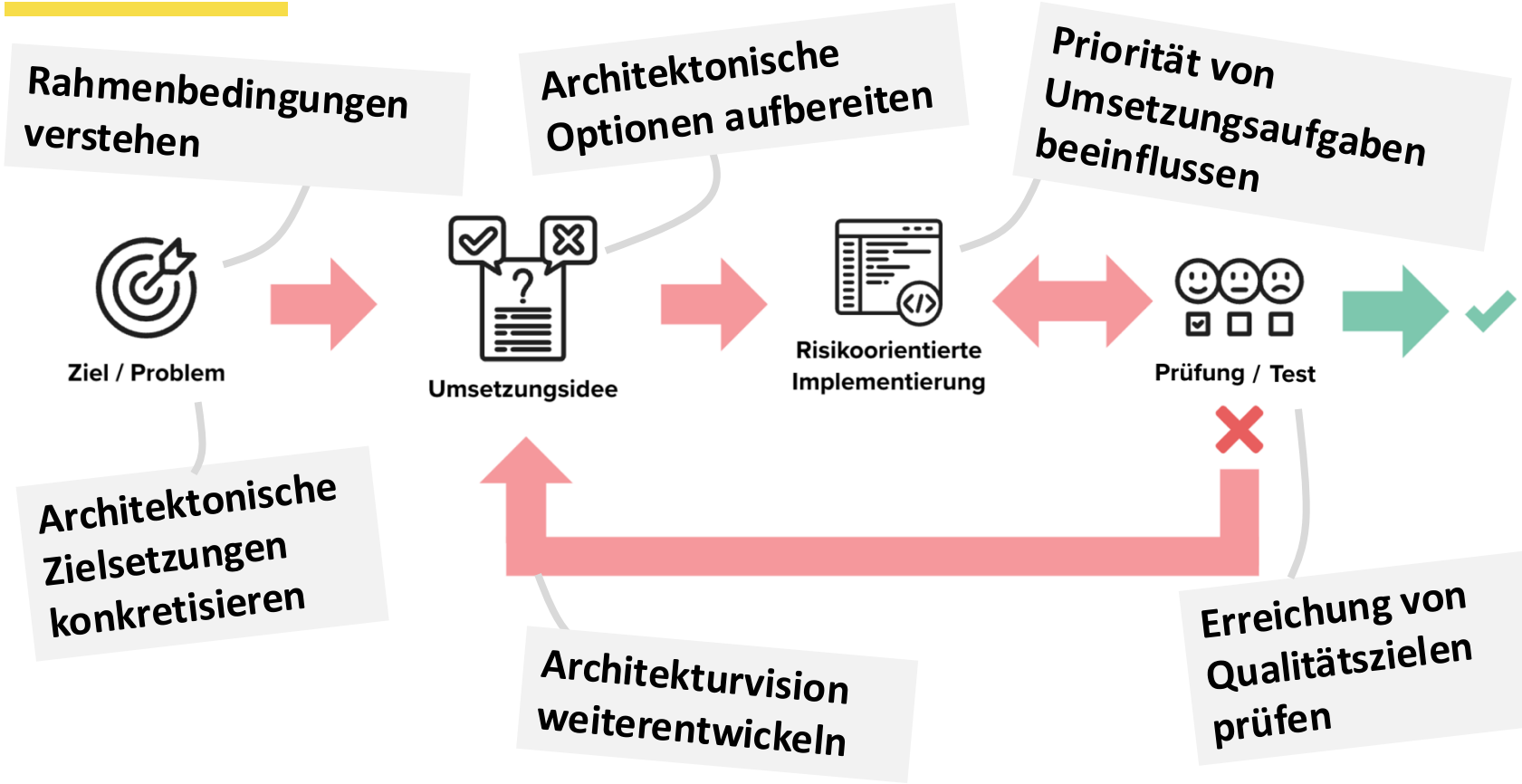
Szenario-Kategorie

- ☐ Akzeptanzkriterium
- ☐ Qualitätsgeschichte
- ☐ Prinzipienlücke

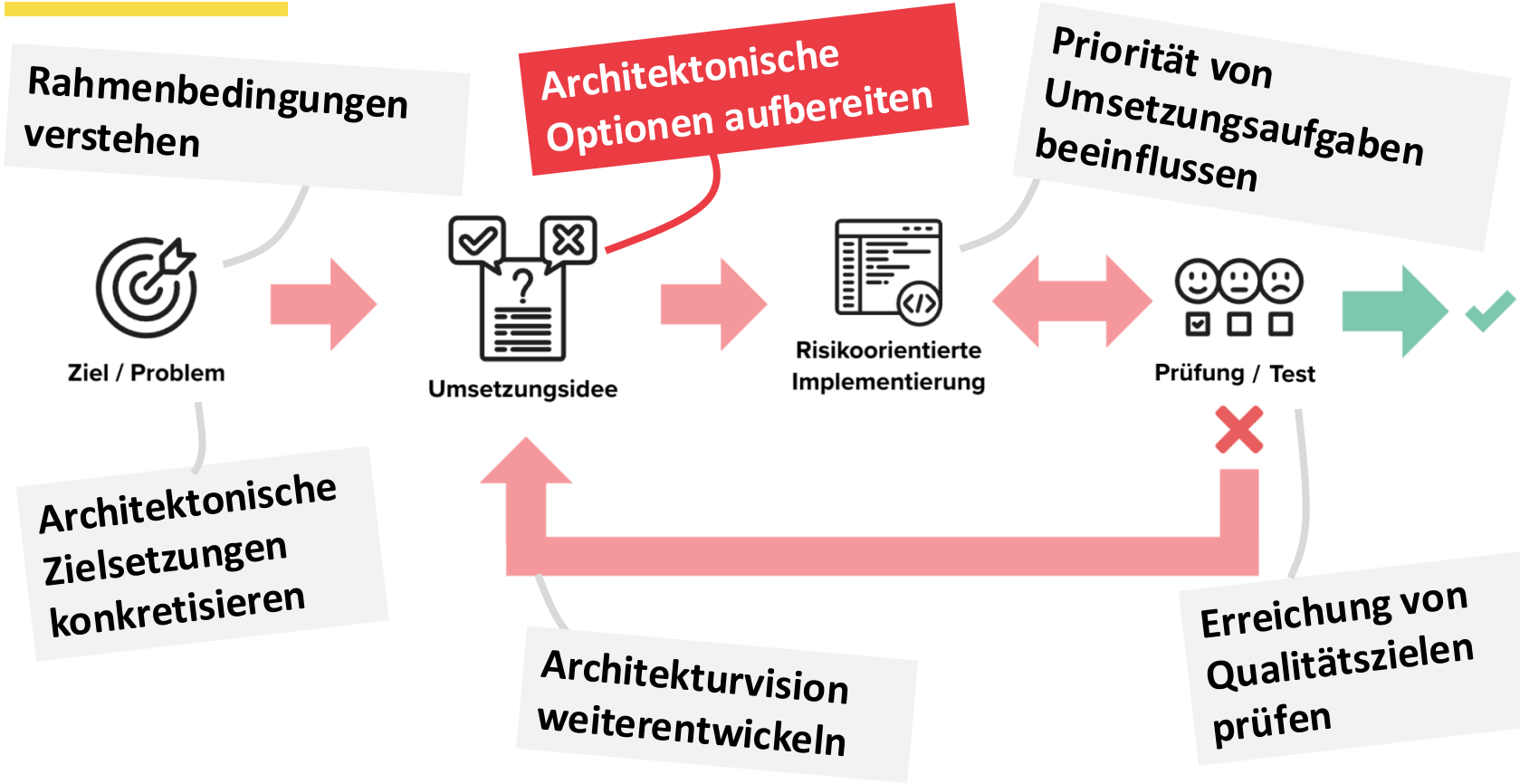
Aufwandsschätzung

☐ S ☐ M ☐ L ☐ XL

Zielorientierung, Flow, ausreichend Qualität…

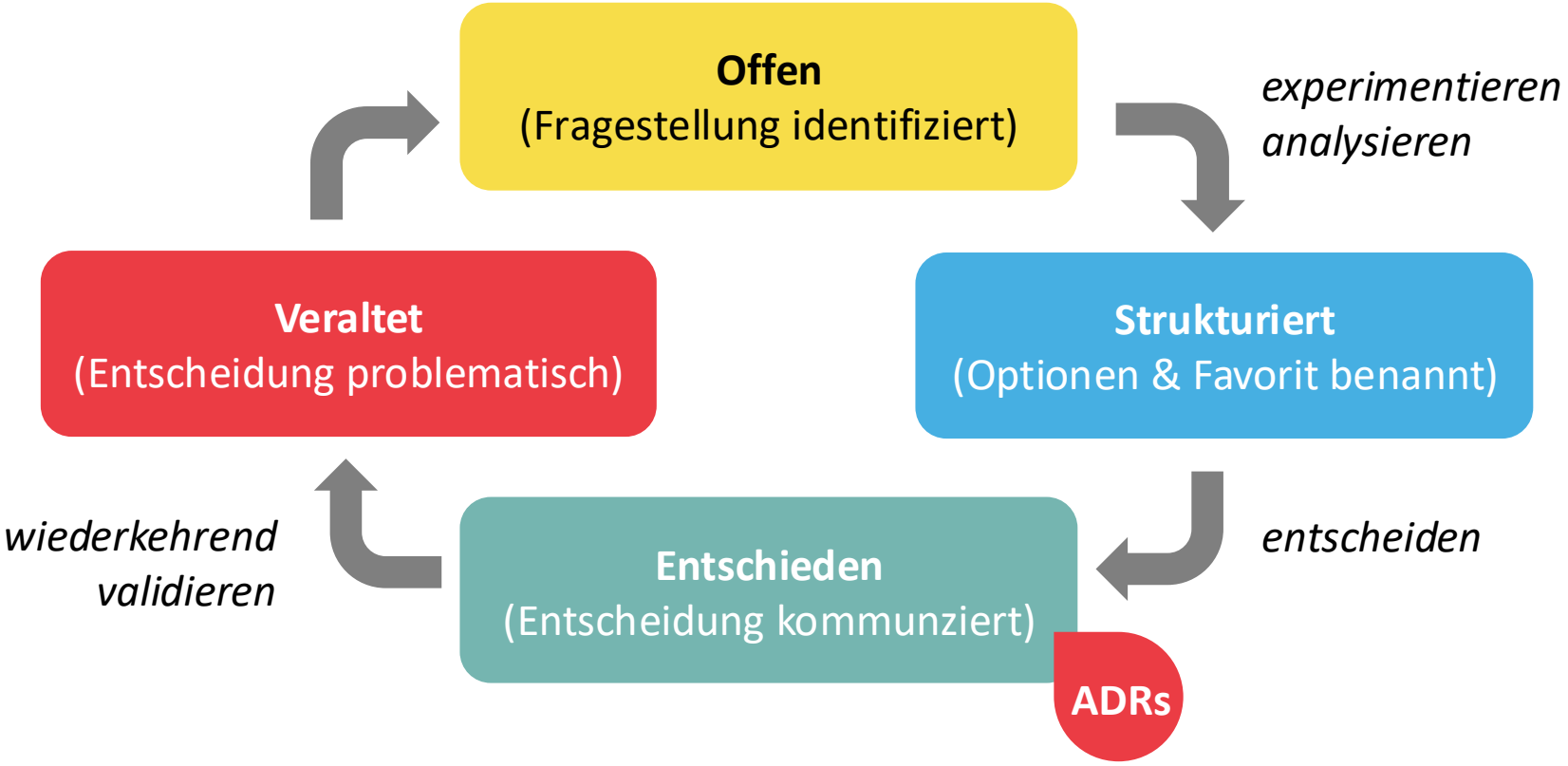


Zielorientierung, Flow, ausreichend Qualität…

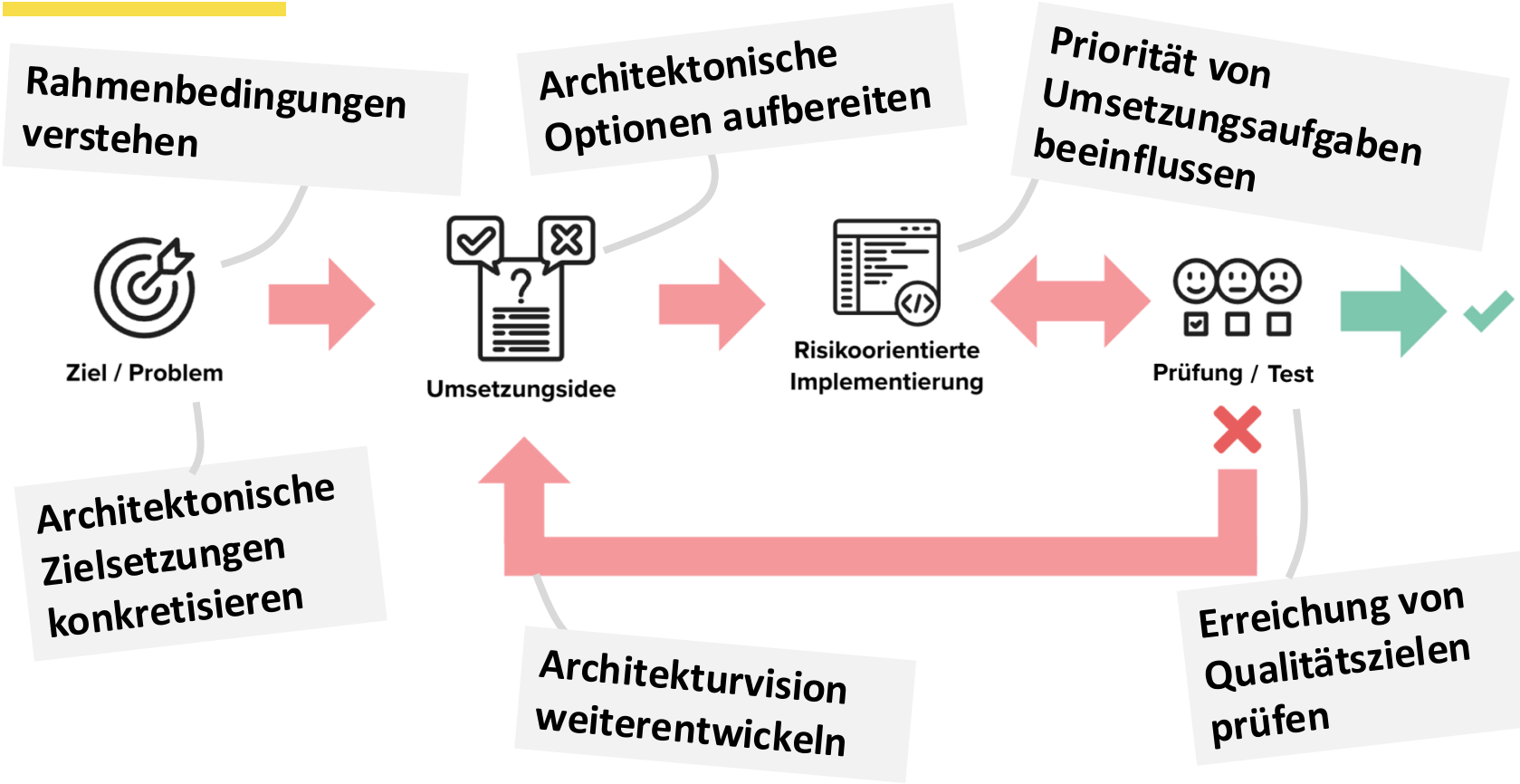


Architekturentscheidungen - ADRs

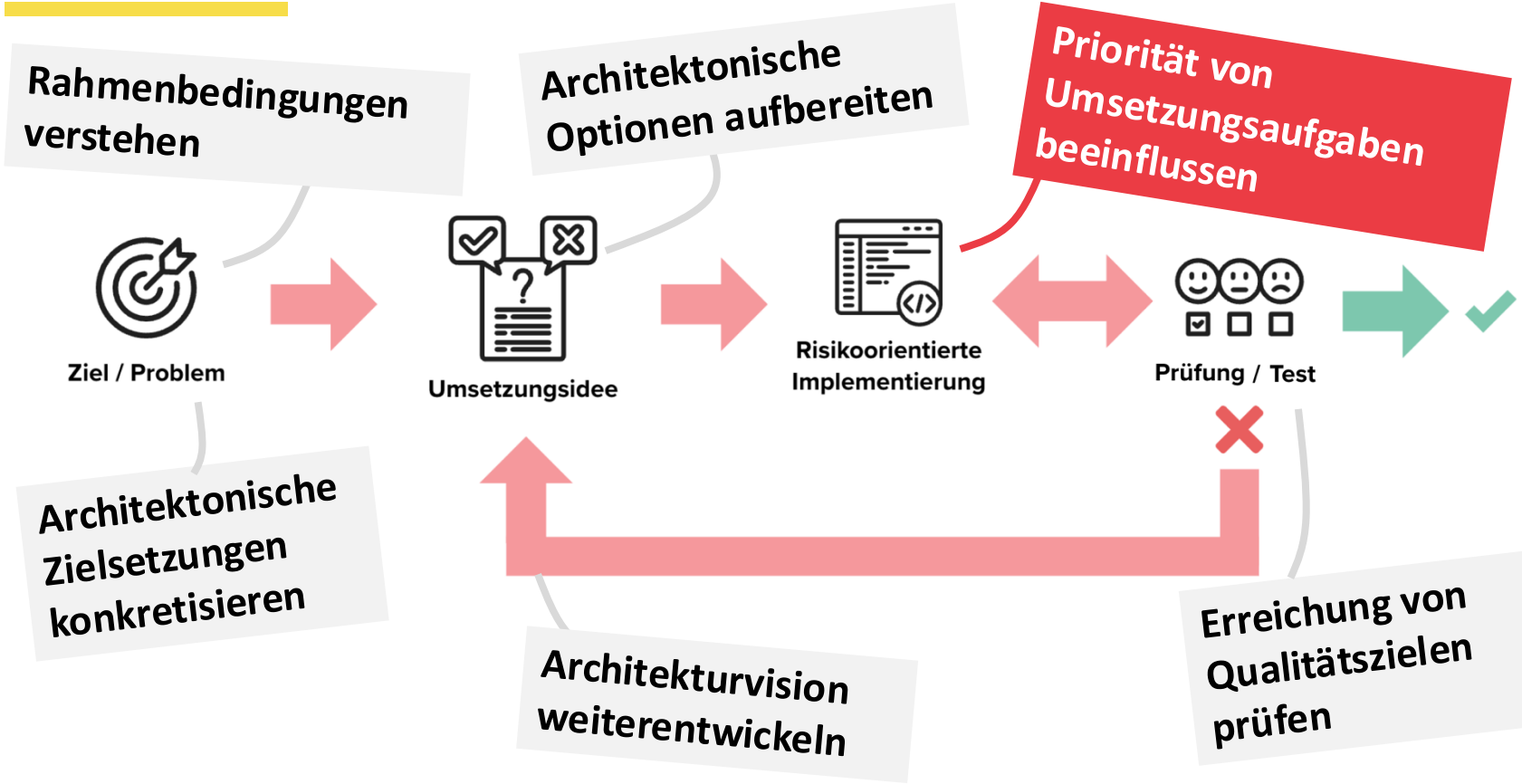
Architektonische Fragestellungen



Zielorientierung, Flow, ausreichend Qualität…



Zielorientierung, Flow, ausreichend Qualität…



Walking Skeleton

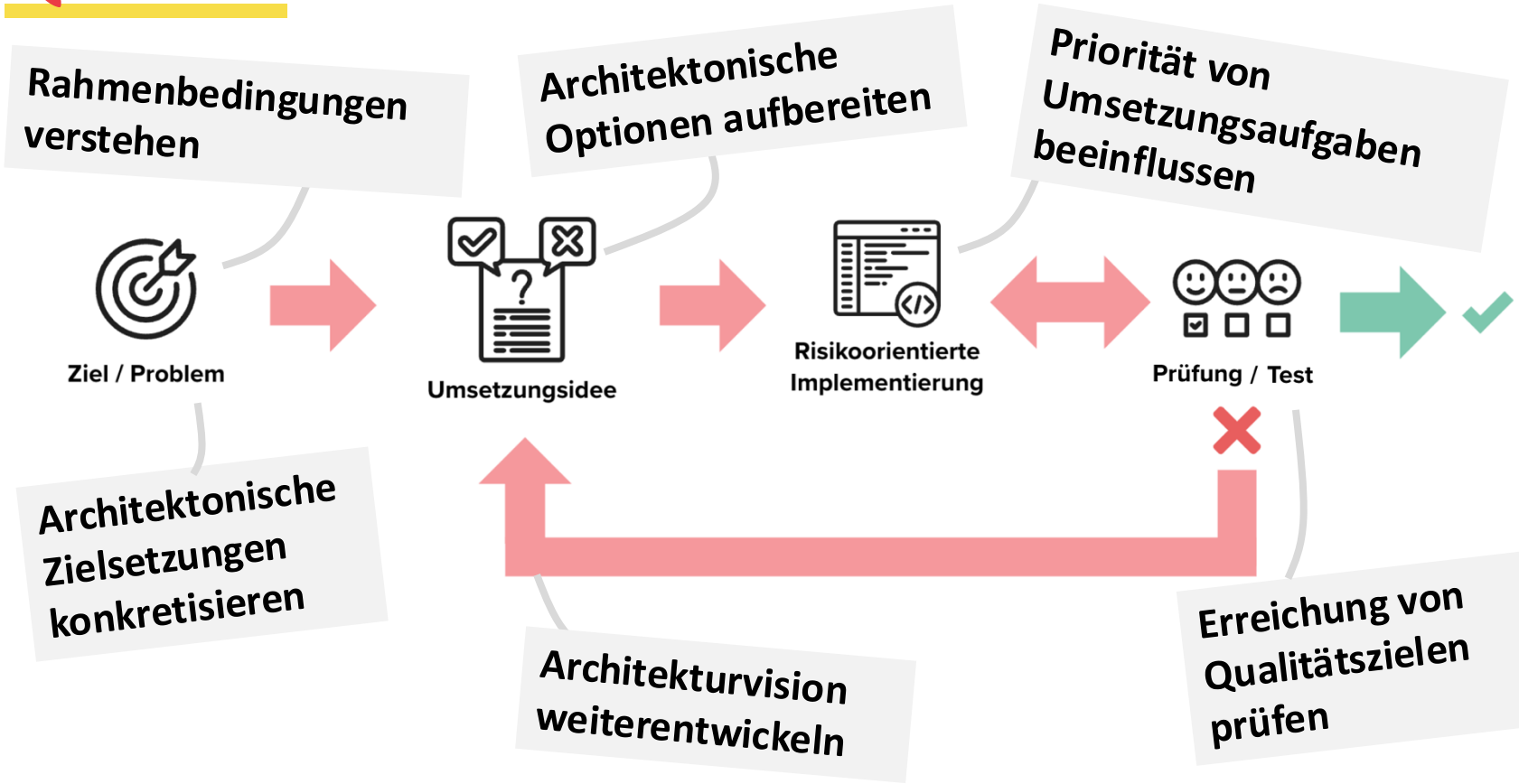
„**A Walking Skeleton** is a tiny implementation of the system that performs a small end-to-end function. It need **not** use the **final** architecture, but it should **link together** the **main architectural components**. The architecture and the functionality can then evolve in parallel.“

- Alistair Cockburn

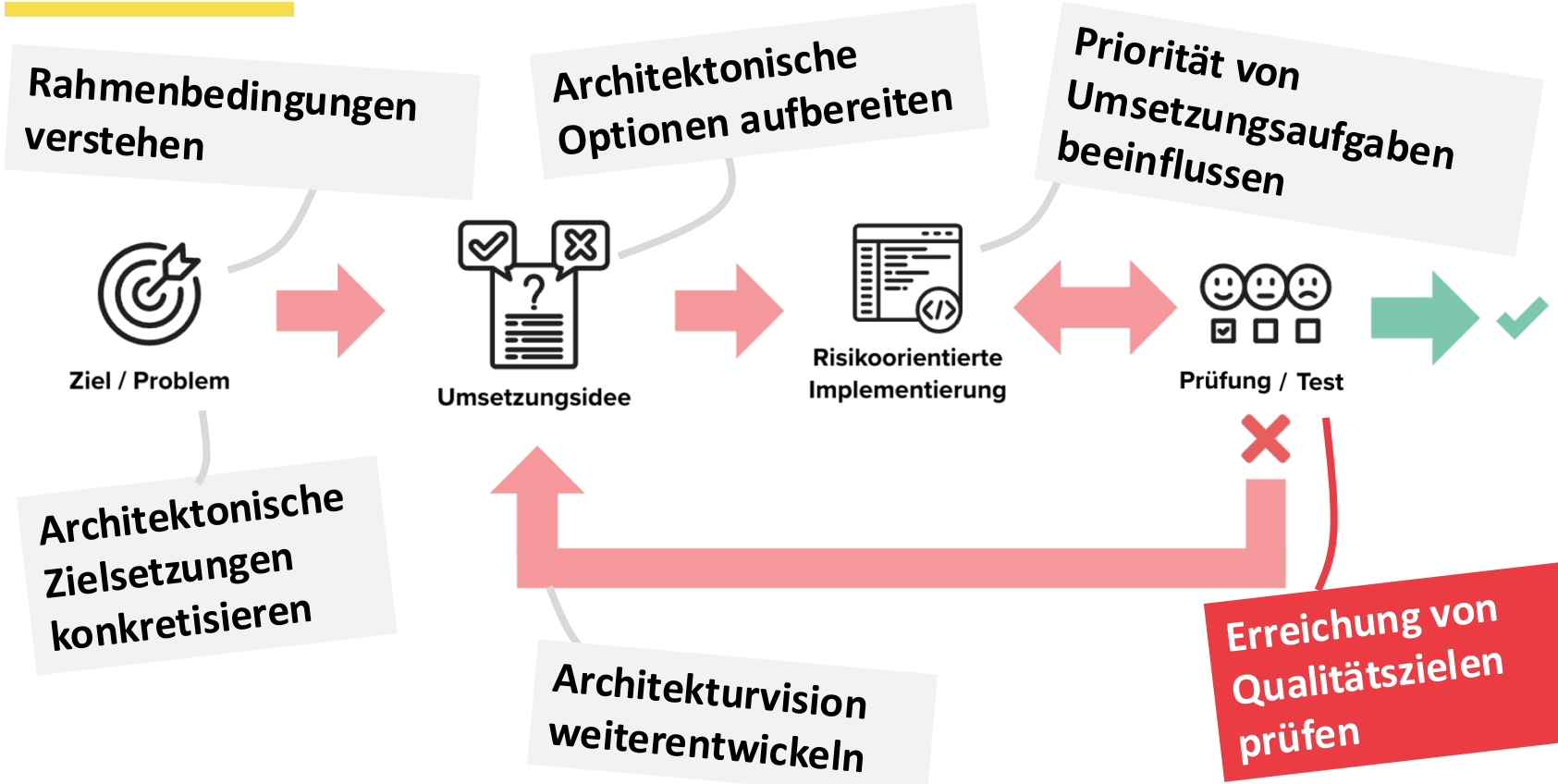
Offen
(Fragestellung identifiziert)

Strukturiert
(Optionen & Favorit benannt)

Zielorientierung, Flow, ausreichend Qualität…



Zielorientierung, Flow, ausreichend Qualität…



Elemente von Architekturarbeit

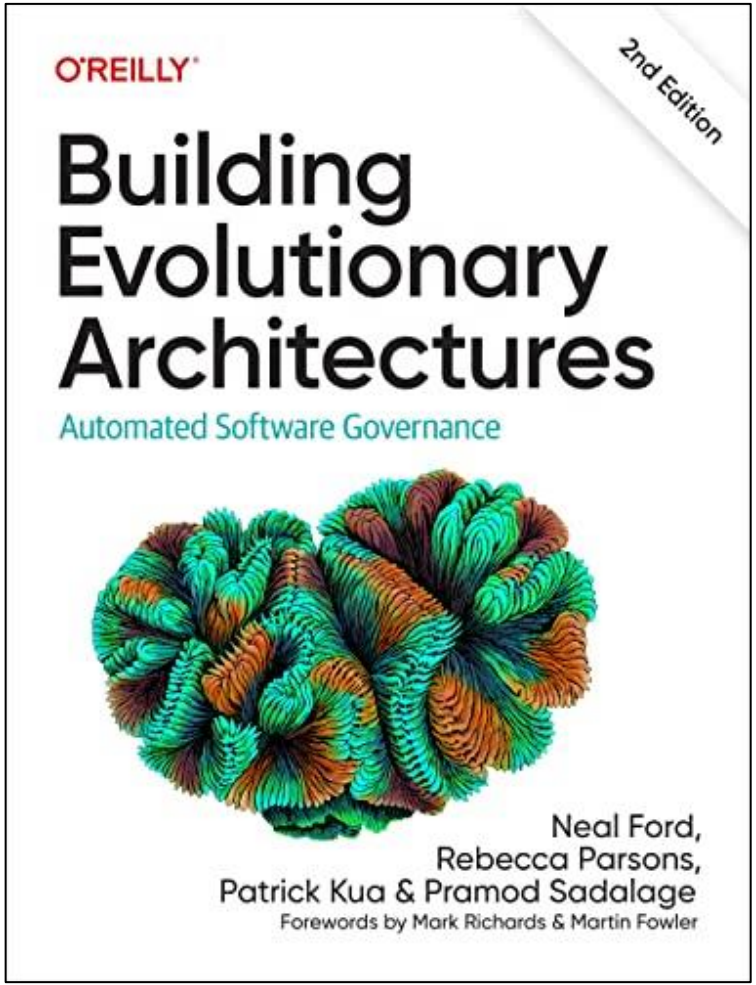
- Effizienz
- Sicherheit
- Zuverlässigkeit
- Skalierbarkeit
- Wartbarkeit
- Portierbarkeit
- …

„Qualitätsmerkmale“

- Strukturierung
- Muster, Stile
- Framework, Libs
- Protokolle
- Basistechnologien
- Umgebungen
- …

„Entscheidungen“

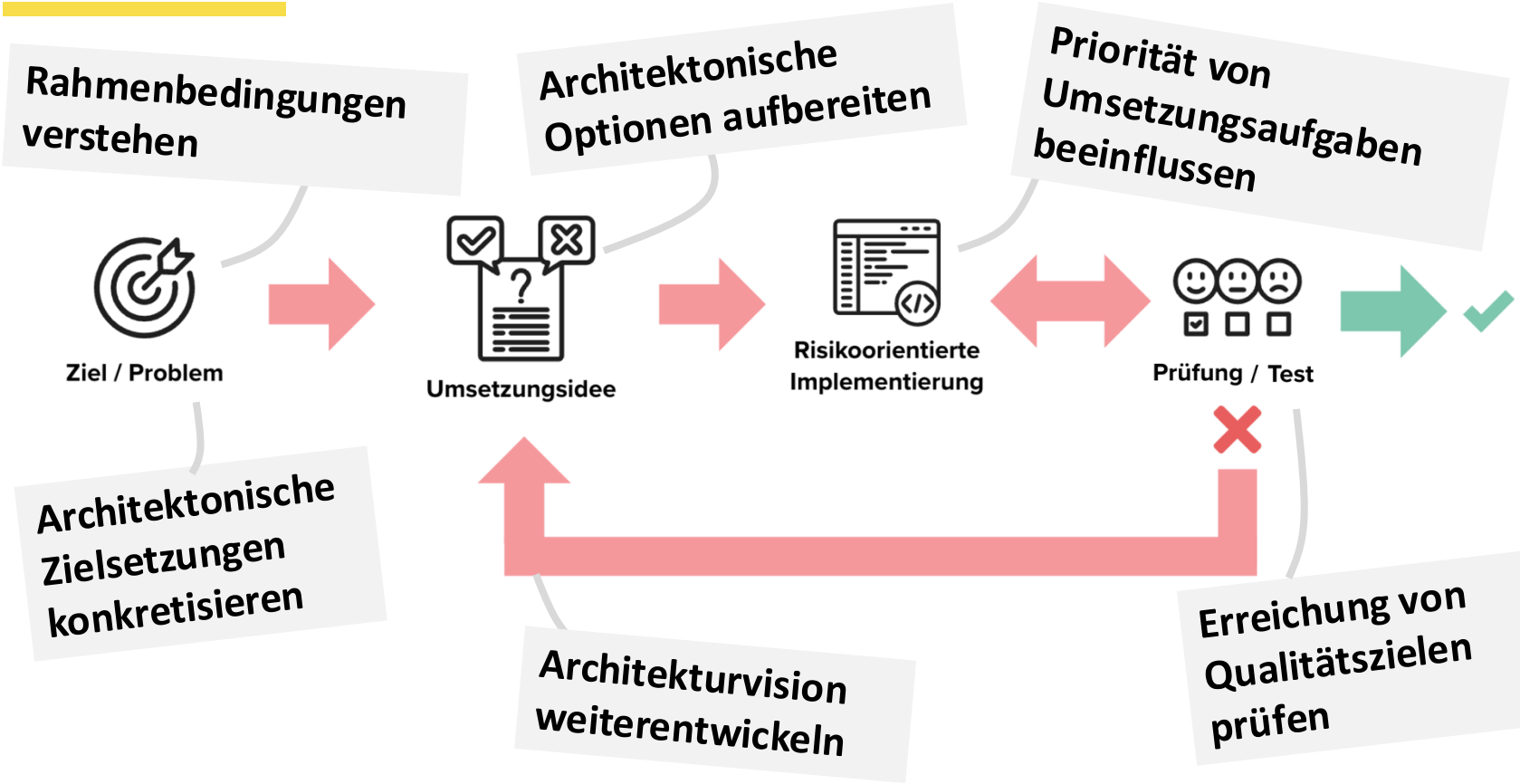
Feedback!
?



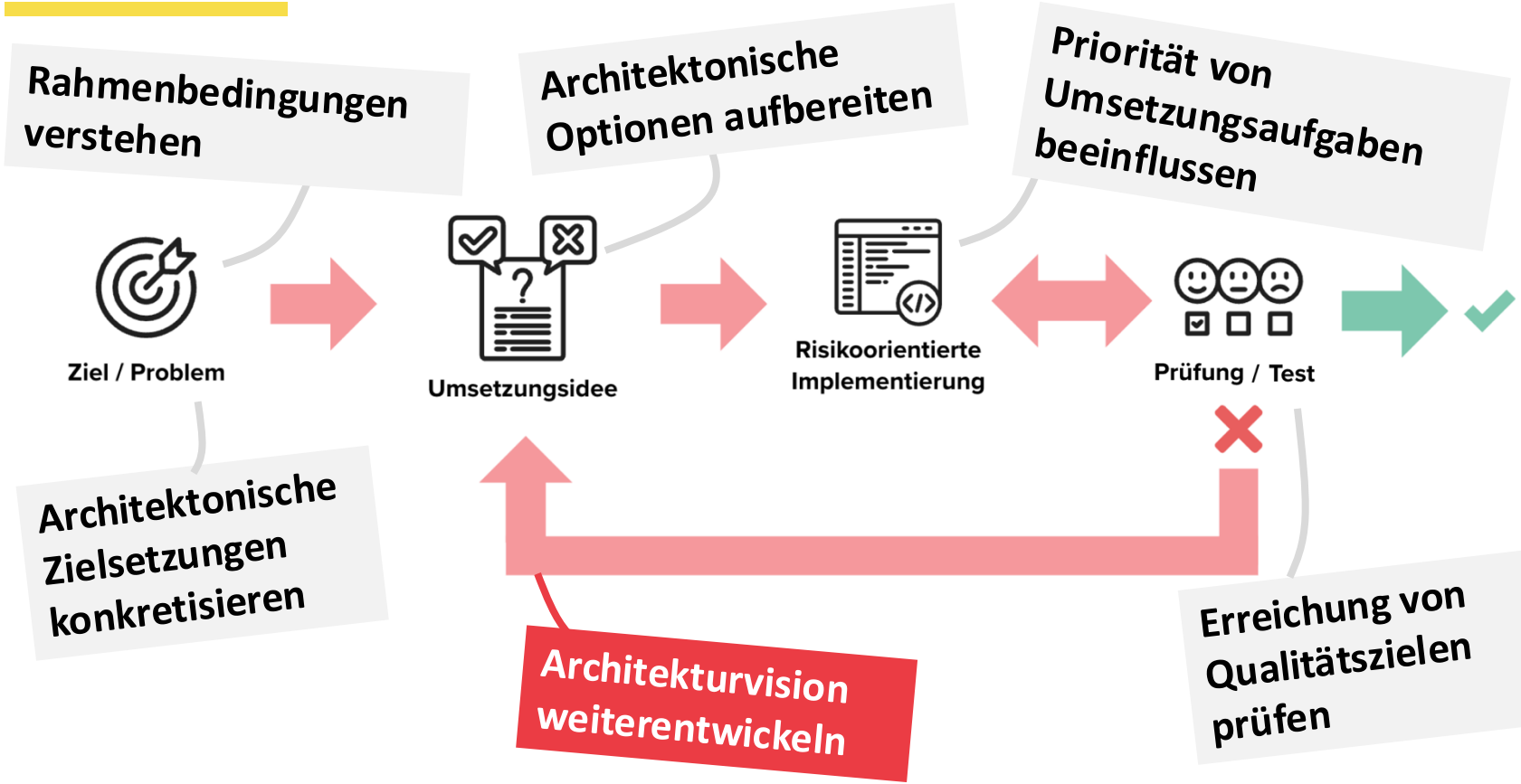
Eine **Fitness Function** misst objektiv, wie gut eine Lösung die an sie gesetzten Ziele erreicht.

- Building Evolutionary Architectures – Ford, Parsons, Kua, Sadalage, O'Reilly 2017

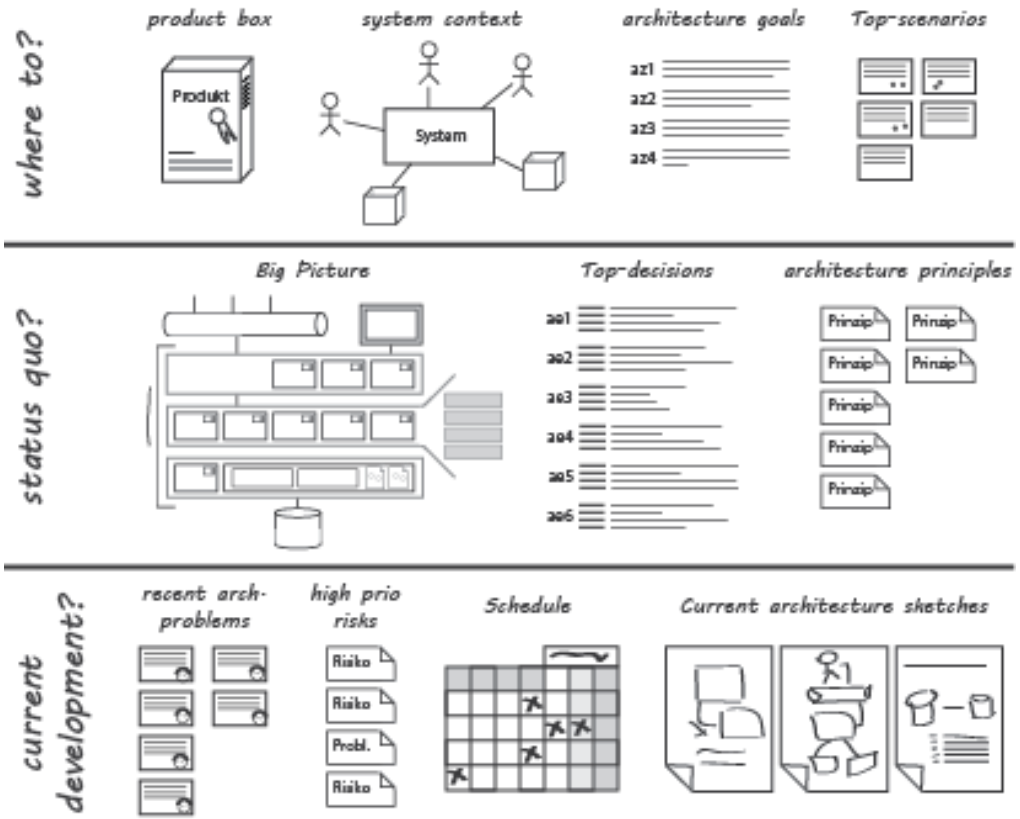
Zielorientierung, Flow, ausreichend Qualität…



Zielorientierung, Flow, ausreichend Qualität…



Vision und Zustand der Architektur zeigen



- Knapper Überblick
- Änderungshäufigkeit: unten mehr
- Physisch oder digital
- Erhöht Transparenz
- Ermöglicht Post-It Sessions

Vision and Goals

Quality Goals

1. Maintainability
2. Reliability
3. Functional Correctr
4. Performance
5. Interoperability
6. Usability
7. Security
8. Portability

System Con

Technical Principles

Process and Methodology

Architecture and the development process

Requirements Definition → Scheduling → Analysis / Detailing of Requirements → Implementation → Test → Deployment

nciples of Architectural practice

Early discovery, late decision

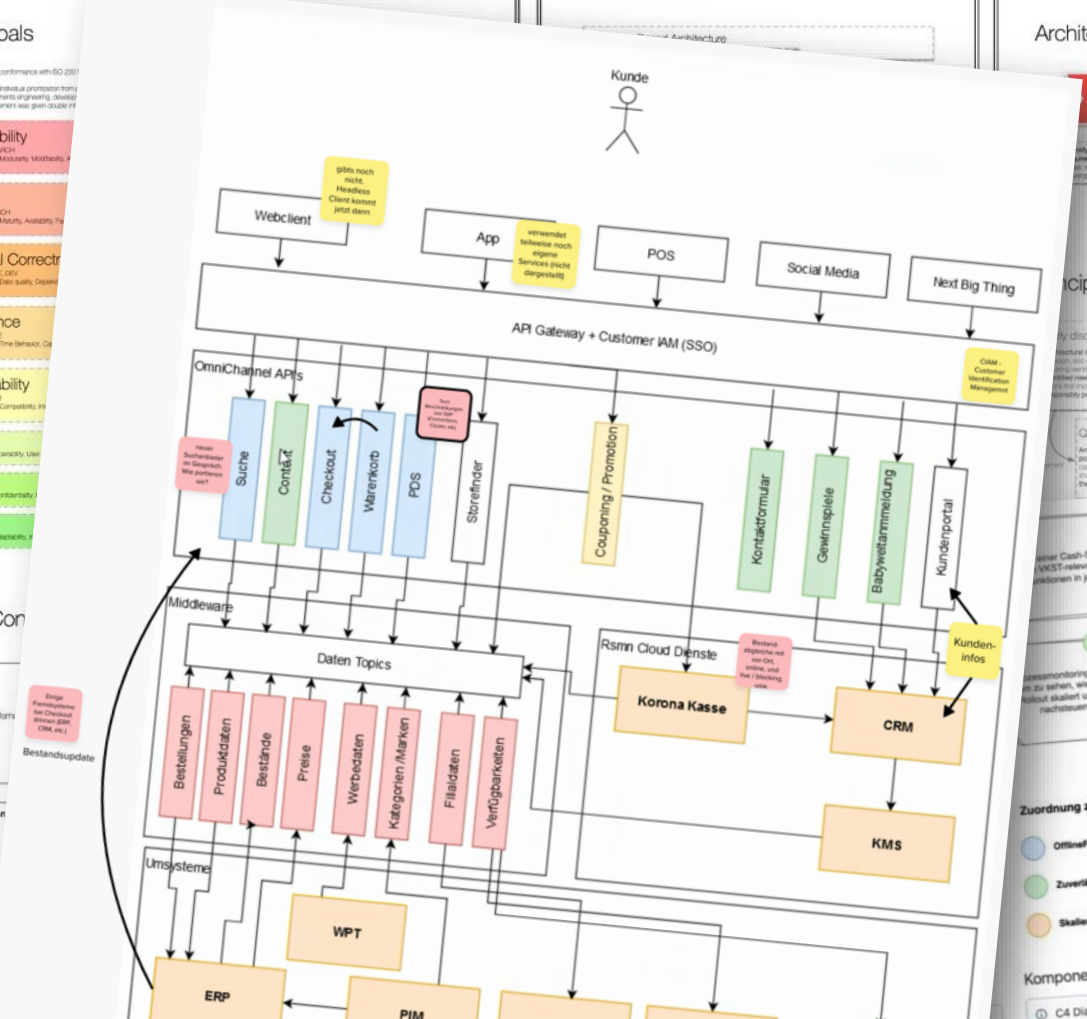
Quick feedback from slim implementation

Zuordnung zu Architekturzielen (Farbcode)

OfflineFähigkeit · Benutzbarkeit · Zuverlässigkeit · Wartbarkeit · Skalierbarkeit

Komponentendiagramm

C4 Diagramme zur Darstellung von:

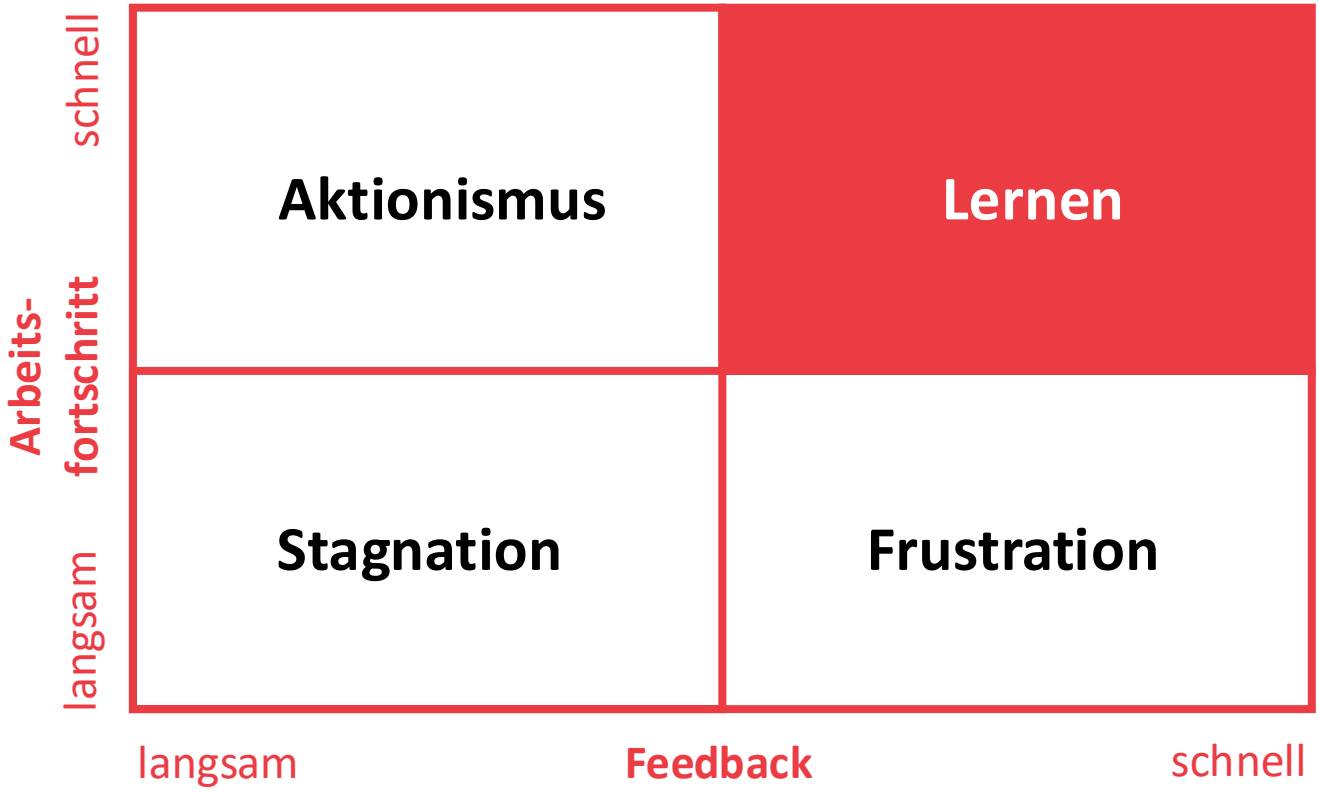


04.

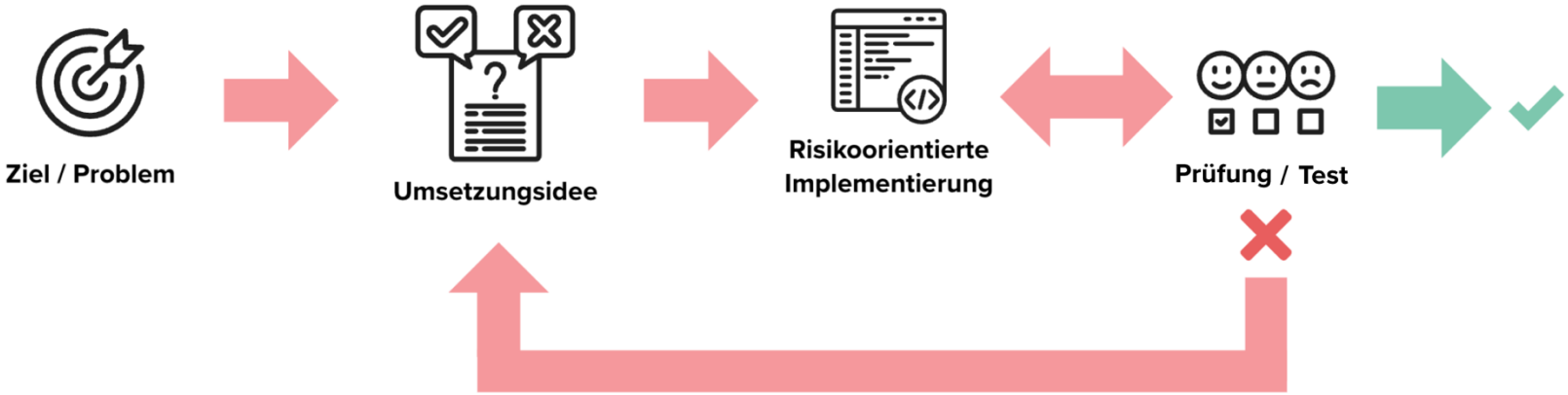
Und AI !?11!!!

AI! denkt irgendjemand auch mal an AI?

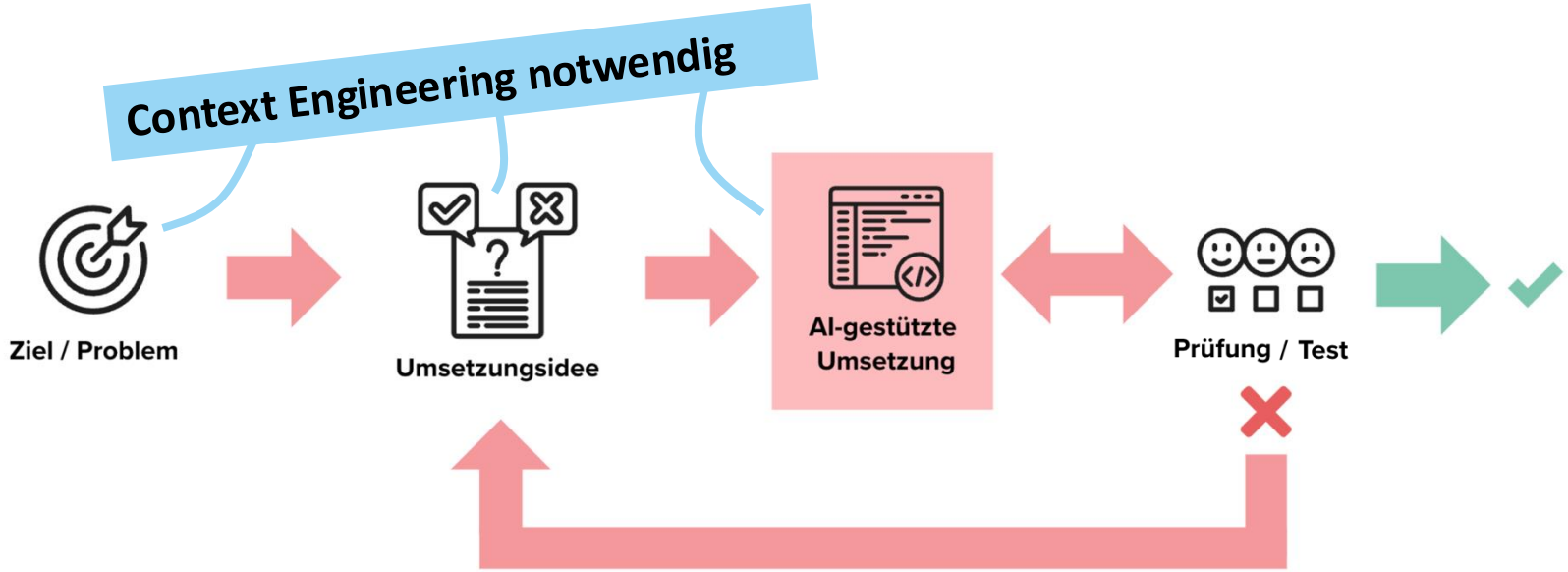
Feedback & Lernen



Was ändert die Verwendung von LLMs?



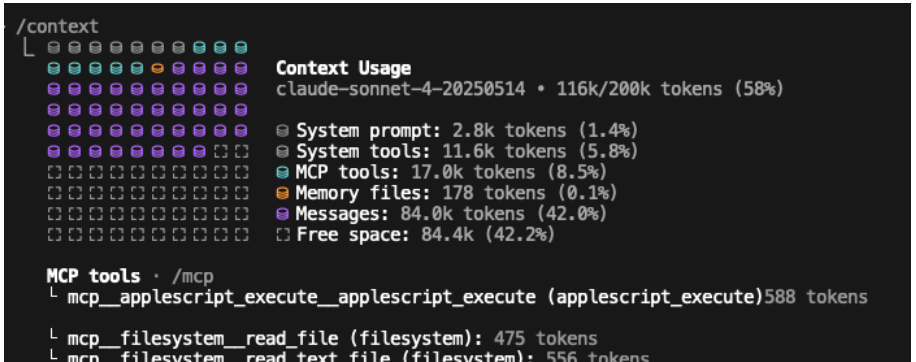
Generative KI in der Entwicklung



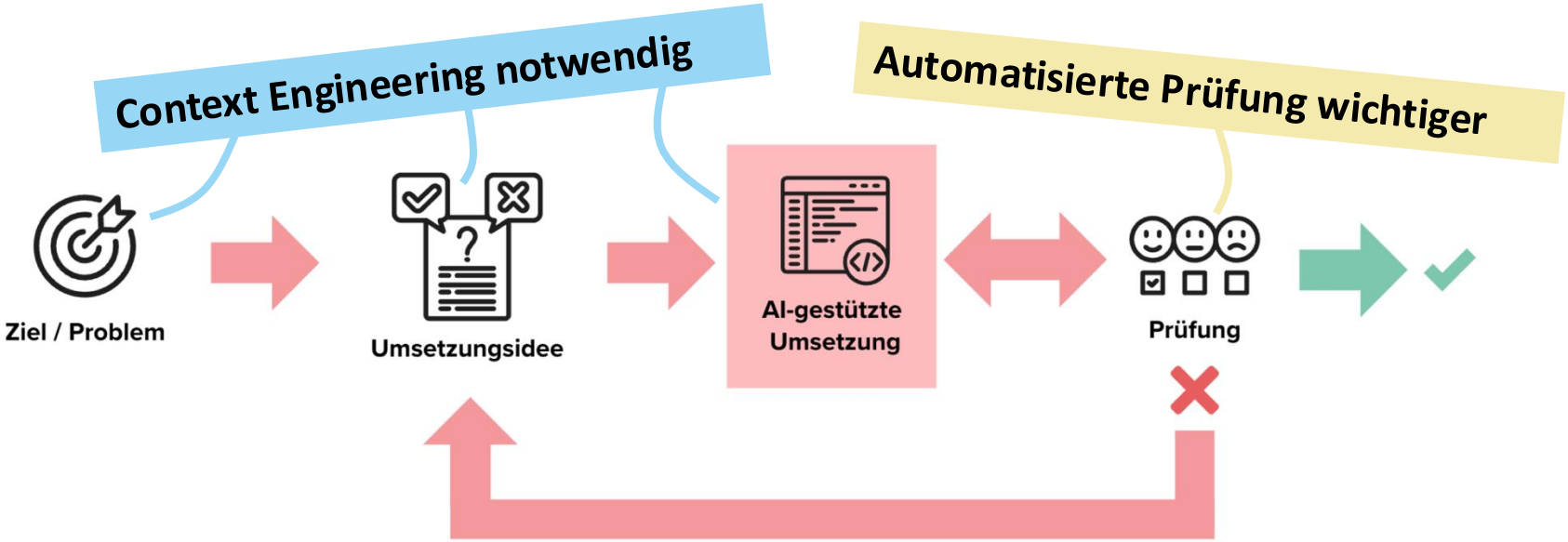
Context Engineering Optionen

- Claude.md / Agents.md
- Regeln (z.b. als .md)
- System Prompt
- Tools (Fähigkeiten)
- MCP-Server
- Sub-Agenten / Modes (eigener Kontext)
- Skills
- Hooks
- Templates
- …

```
your-project/
├── .claude/
│   ├── CLAUDE.md           # Main project instructions
│   └── rules/
│       ├── code-style.md   # Code style guidelines
│       ├── testing.md      # Testing conventions
│       └── security.md     # Security requirements
```

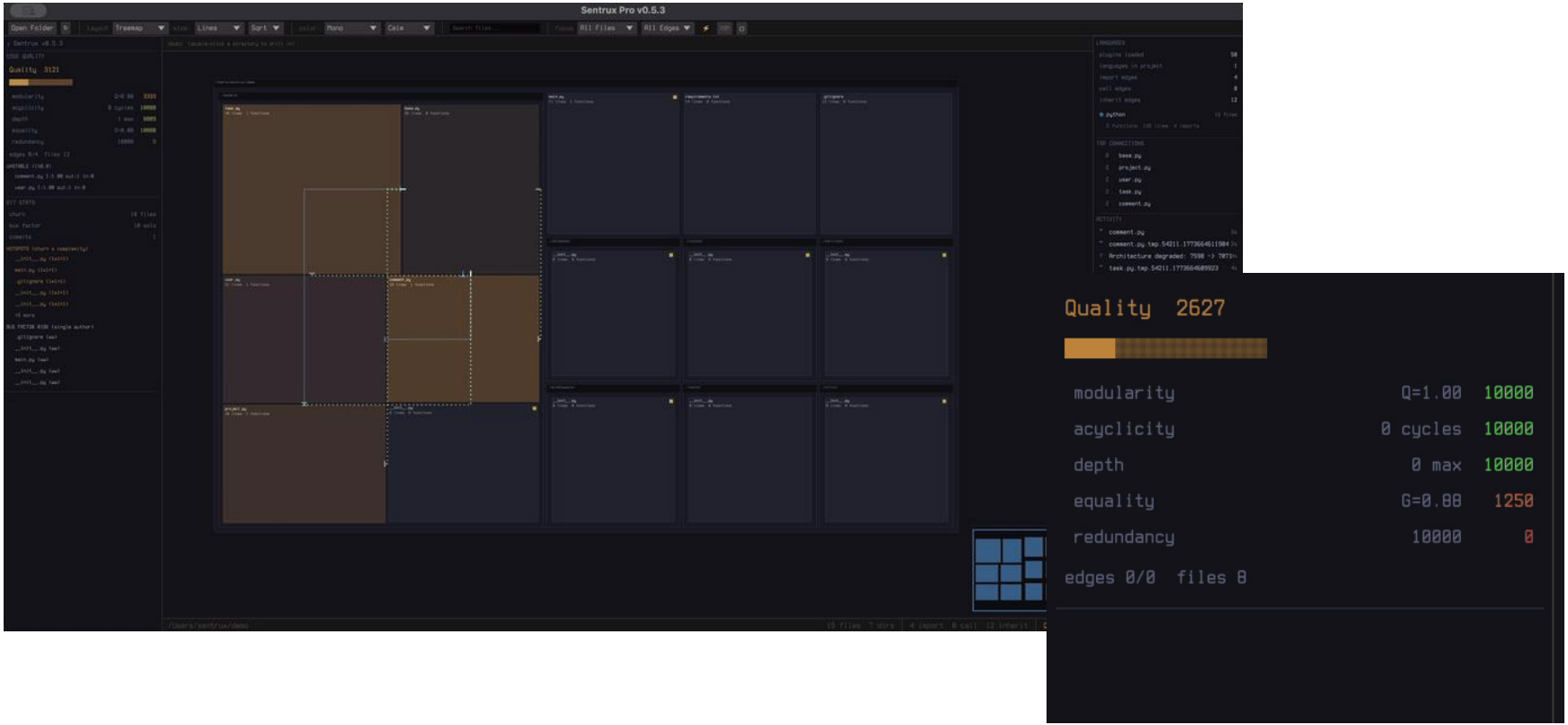


Generative KI in der Entwicklung



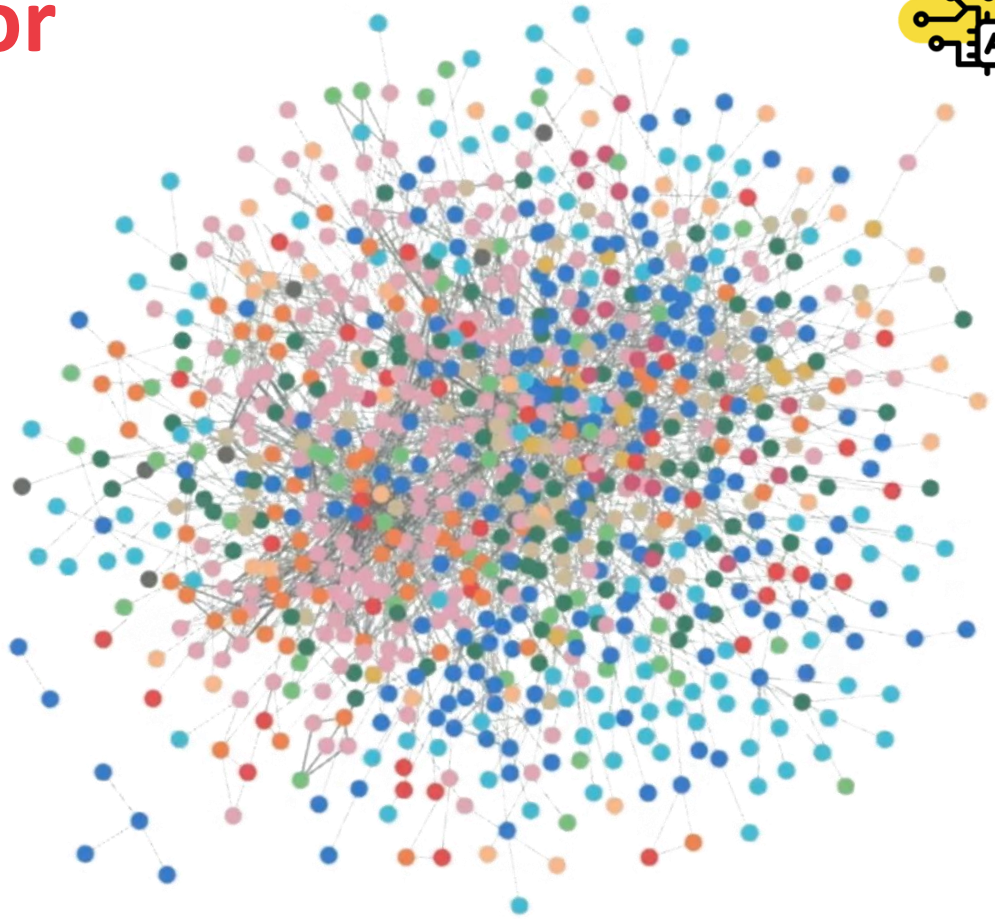
Design Feedback-Loop

sentrux

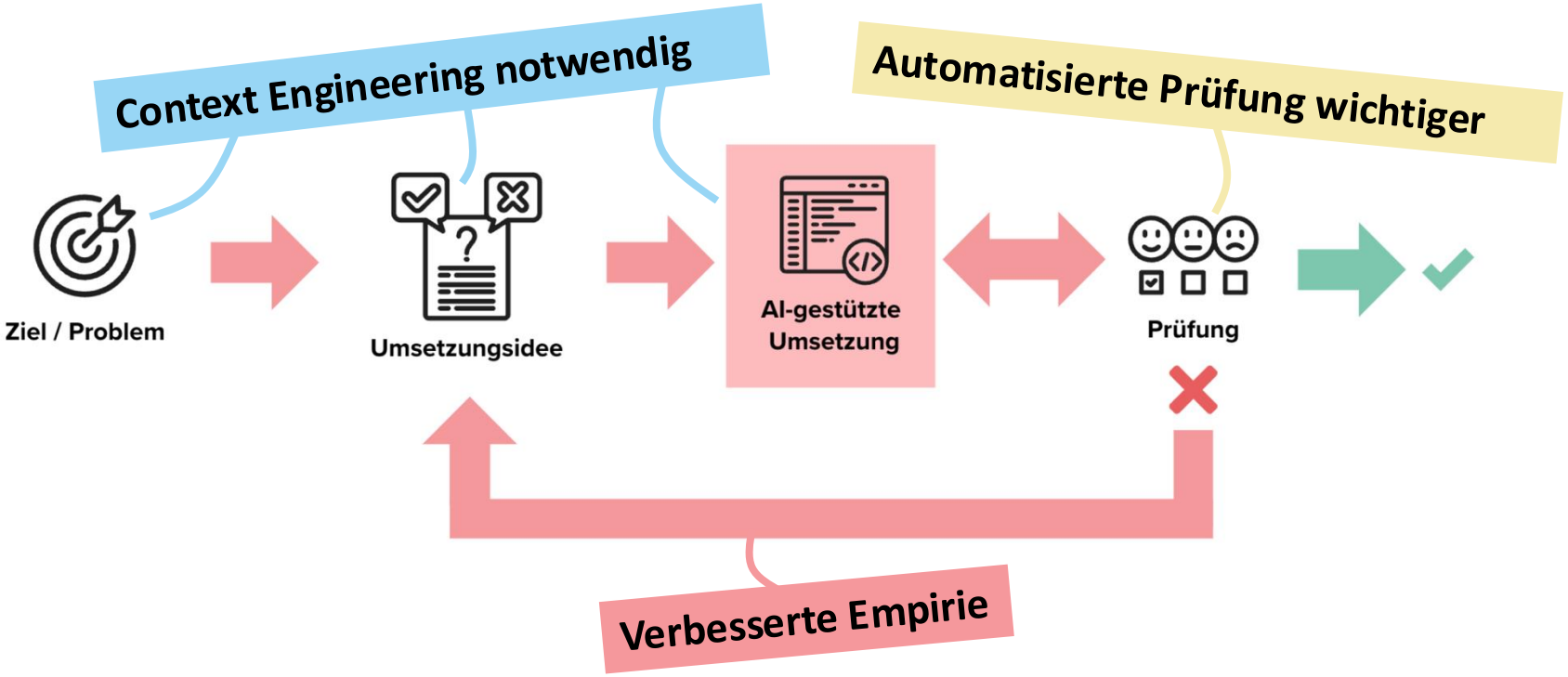


Graph Constructor

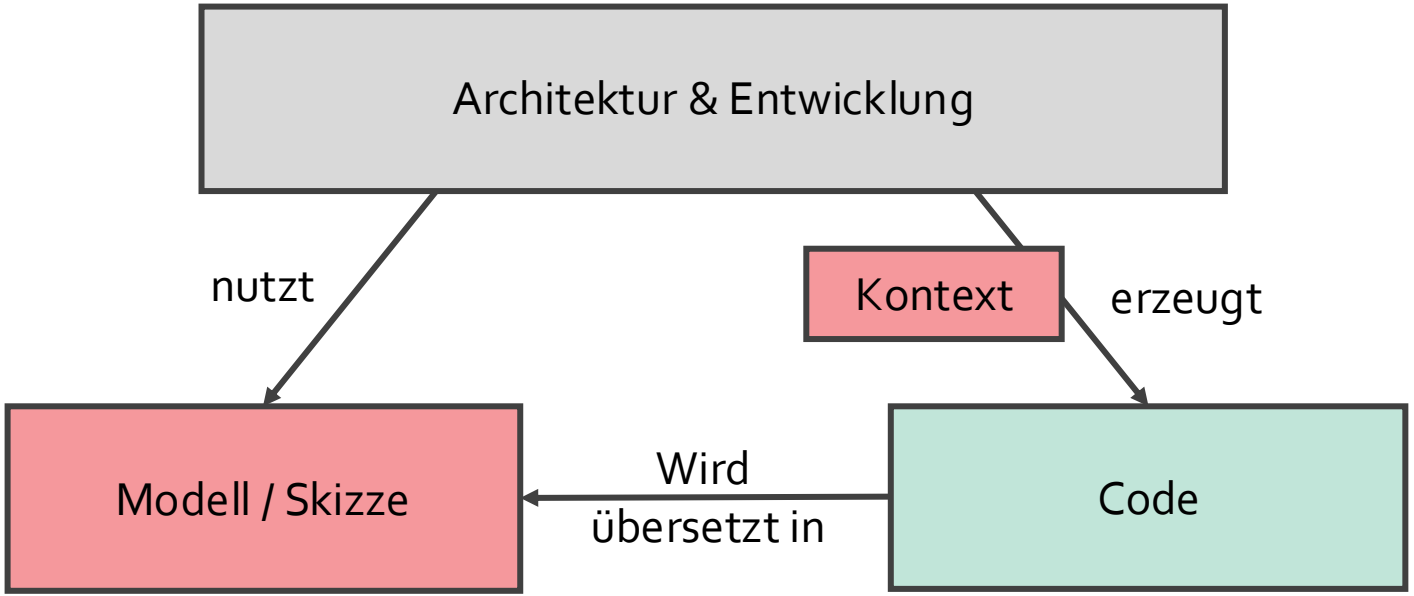
- Entitätsbeziehungen
- Methodenaufrufe
- Endpunkte
- Gemeinsame Änderung
- DB-Beziehungen
- Remote Methodenaufrufe
- Config-Dependencies
- Semantische Nähe
- …



Generative KI in der Entwicklung



Agentic: Architektur und Entwicklung



Aktuell als RC veröffentlicht: AGENTA

iSAQB Advanced Level Curriculum AGENTA

Status

CI - Releases and Main passing | last commit last friday | contributors 6 | issues 8 open | issues 36 closed

This is copyrighted work.

Content

This repository contains the iSAQB Advanced Level curriuclum AGENTA in AsciiDoc.

Released documents

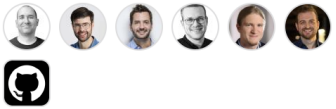
The latest released HTML and PDF documents (German and English) are published via GitHub Pages at public.isaqb.org/curriculum-agenta.

They are produced by CI on each release.

How to build locally

Prerequisite: Docker. Nothing else — no JDK, no Ruby.

Contributors 7



Languages

Batchfile 54.8% Shell 45.2%

Generated from isaqb-org/advanced-template

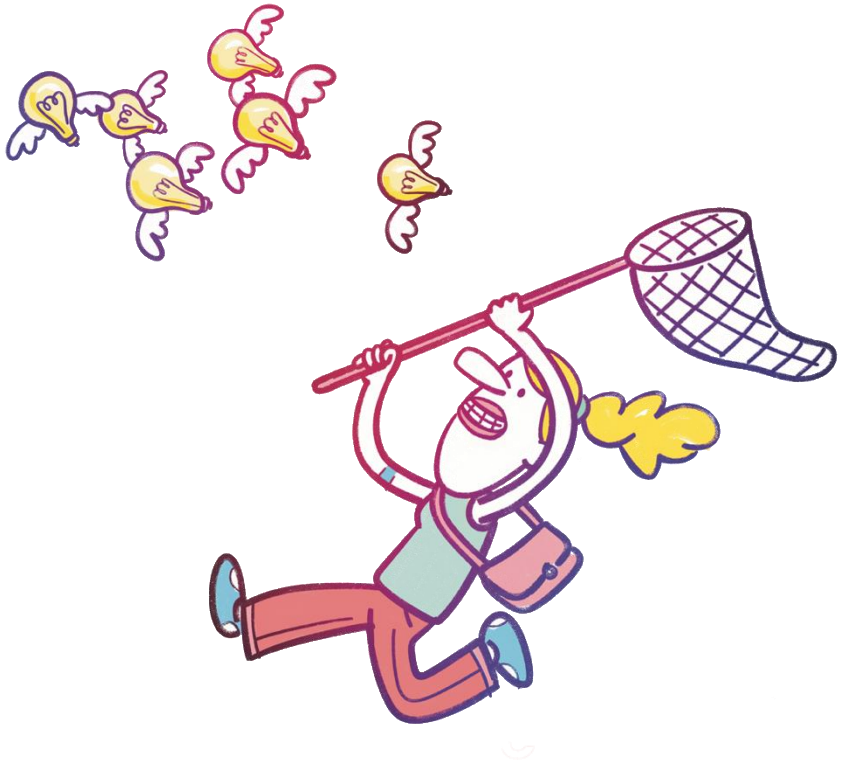
Architektur für Agentic Engineering Kontexte

https://github.com/isaqb-org/curriculum-agenta

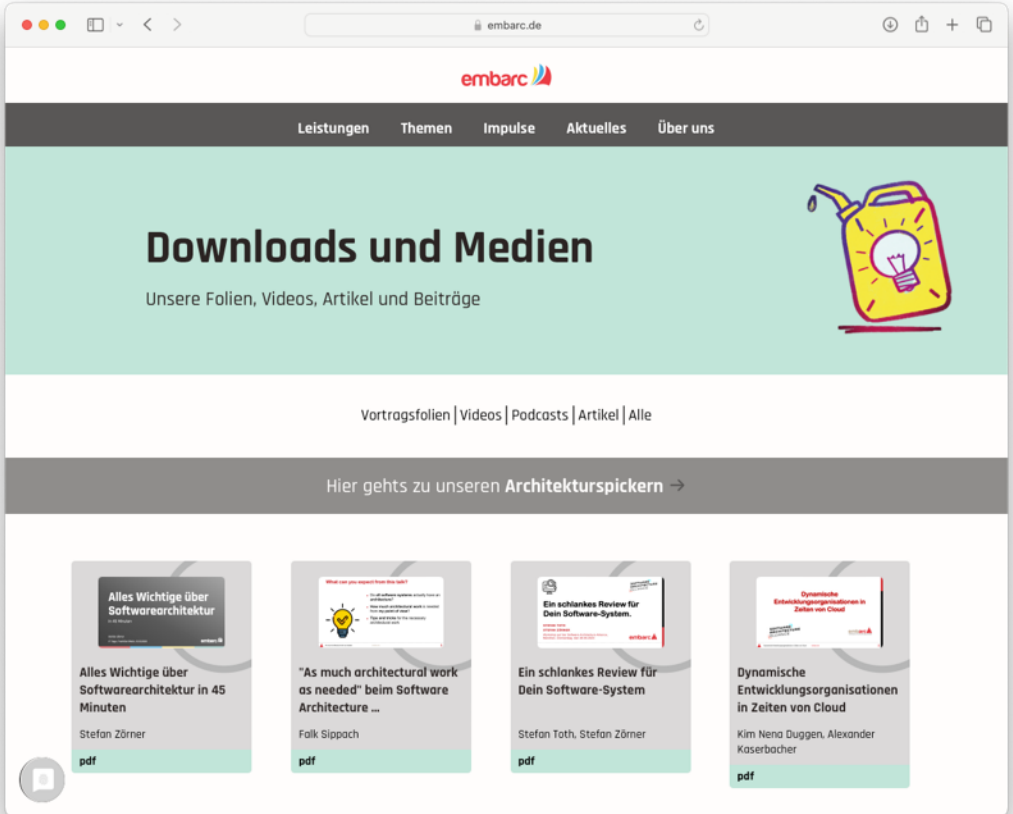
05.

Abschluss

Weitere Informationen...



Folien als PDF zum Download



→ embarc.de/download/

04/2025

Vorgehensmuster für Softwarearchitektur

Kombinierbare Praktiken in Zeiten von Agile und Lean

Stefan Toth

4. Auflage
Hanser Verlag

AGILA – Agile Softwarearchitektur

https://www.socreatory.com/de/trainings/agila

Feedback & Fragen?

Ich freue mich auf Fragen, Diskussionen, Pizza!