

# Architekturbewertung im Agilen Kontext

Stefan Toth | [st@embarc.de](mailto:st@embarc.de) | [st\\_toth \(twitter\)](https://twitter.com/st_toth)

Architekturpunsch 2022  
15. Dezember 2022

# Stefan Toth

✉ st@embarc.de

t @st\_toth

» xing.to/sto

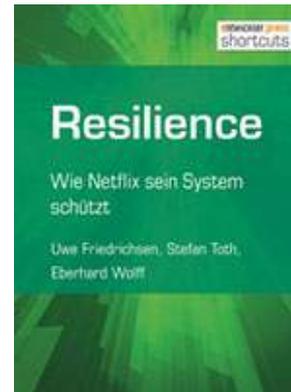


—  
[www.embarc.de](http://www.embarc.de)  
[www.swamuster.de](http://www.swamuster.de)



**embarc**   
Software Consulting GmbH





# Schnell-Einstieg

Architekturbewertung?



# Wir coden...

The image displays a multi-panel IDE interface with several windows open:

- src\_contexts.adb**: A file editor showing Ada code with comments in German, such as "Copyright (C) 2005-2019, AdaCore" and "This is free software; you can redistribute it and/or modify it...".
- Call Stack**: A window showing the current execution stack with 11 entries, including `gns.main.finish_setup` at the top.
- Code Hinting**: A window showing a code completion suggestion for `gns.main.finish_setup` at the current cursor position.
- Batch\_Script**: A terminal window showing the execution of a script with output like "Batch\_Script - all - common: string (1 = 0)".
- Debugger Console**: A window showing the execution of the script, listing thread IDs and process IDs.
- Services**: A window showing system services like `services`, `appjs`, and `adotts`.
- Code Editor**: A window showing a code snippet for a `loggerFunction` and a `dashboard` interface definition.

Multiple panels

Code hinting



# ...bauen Systeme

The screenshot displays a comprehensive flight support system interface. On the left, a map shows flight routes between EGLL and LRBS. The central panel lists flight details for LH 4444D, including aircraft type (A320neo), status (On-Time), and various alerts. The right panel provides a detailed view of flight data, including aircraft registration (DLM4444), departure time (20:26), and arrival time (08:00). Below the main interface, a series of small satellite images are visible, likely representing different views or data points related to the flight path.



# ...wir fragen uns ob das so passt



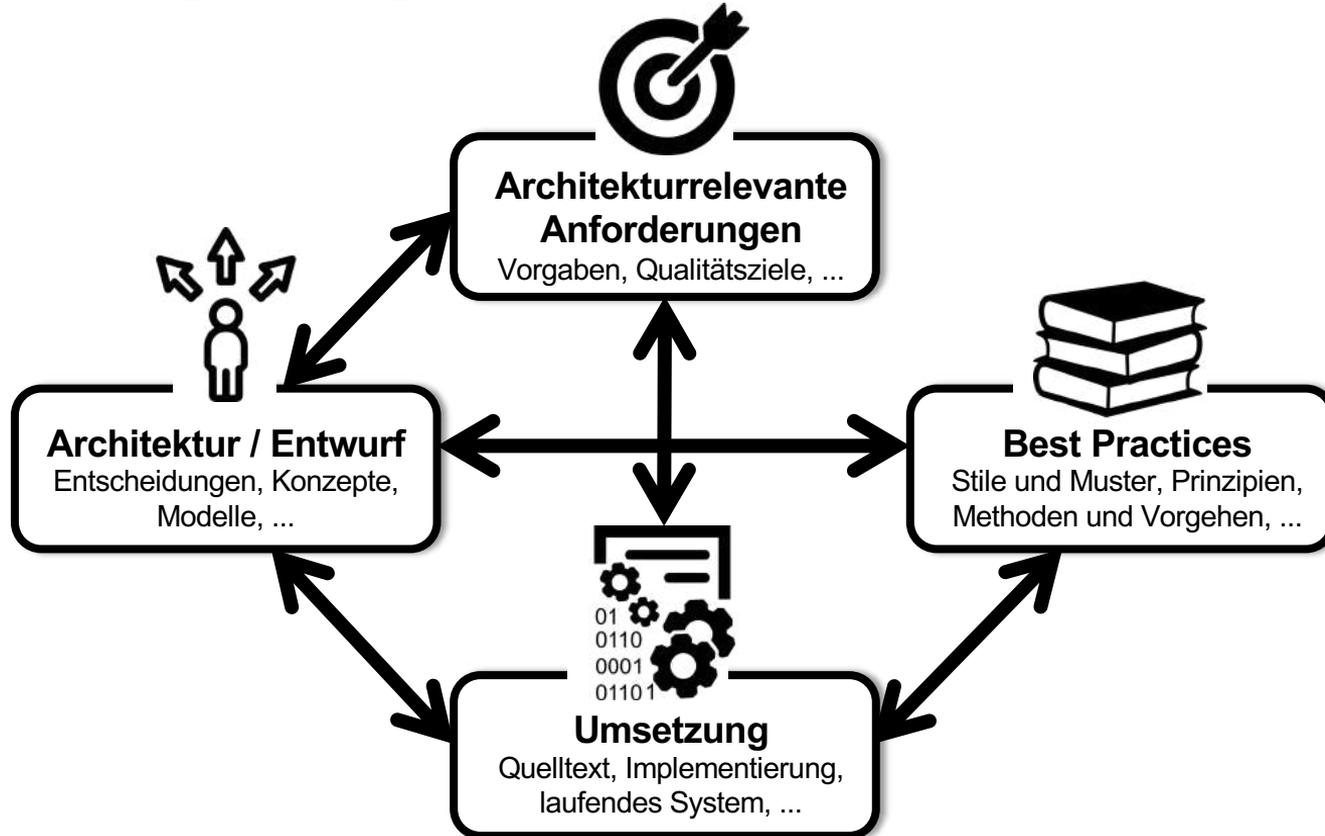


# Architekturbewertung

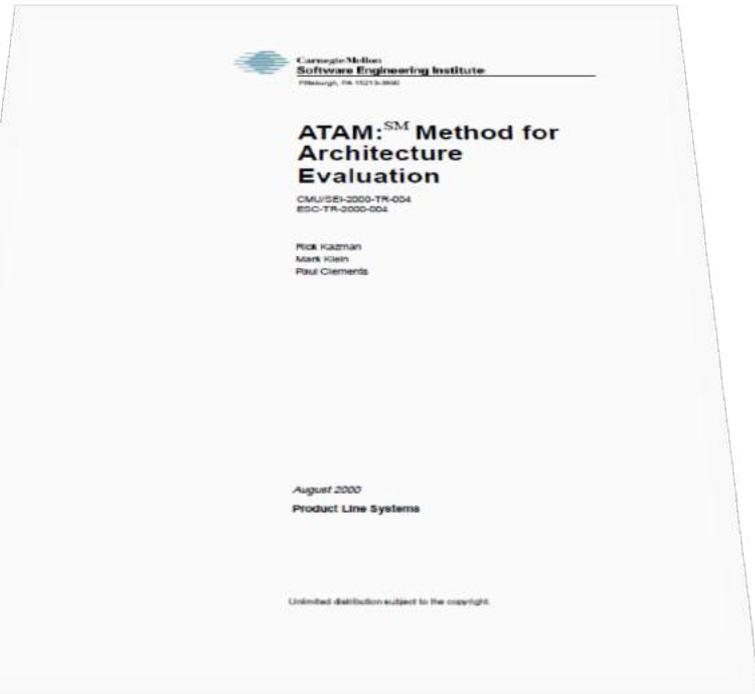
Wie reflektieren wir?



# Bewertungsmöglichkeiten



# Kompromisse, Tradeoffs, ATAM



## ATAM

### Architecture Tradeoff Analysis Method

- verbreitetste Methode zur Bewertung von Softwarearchitektur
- früh anwendbar
- qualitative Bewertung, szenarienbasiert



# ATAM - Architecture Tradeoff Analysis Method

| Step | Activity                                | Stakeholder Groups   |
|------|---|--|
| 1    | Present the ATAM                        | Evaluation team/Customer representatives/Architecture team |
| 2    | Present business drivers                | “  |
| 3    | Present architecture                    | “  |
| 4    | Identify architectural approaches       | “  |
| 5    | Generate quality attribute utility tree | “  |
| 6    | Analyze architectural approaches        | “  |
| 7    | Brainstorm and prioritize scenarios     | All stakeholders   |
| 8    | Analyze architectural approaches        | Evaluation team/Customer representatives/Architecture team |
| 9    | Present results                         | All stakeholders   |





# Reibungspunkte

Mit agilen Vorhaben

„High ceremony“



**Überprüfung**



# Personen die “es wissen“



**Nicht iterativ**



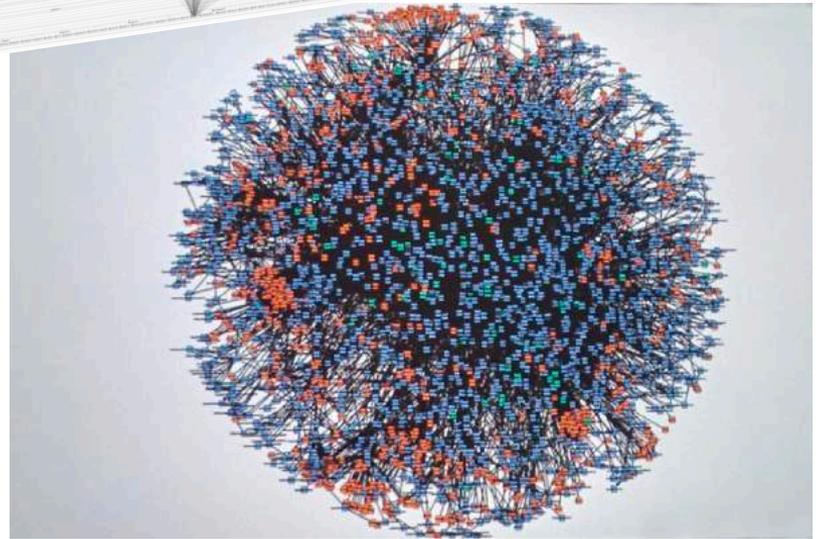
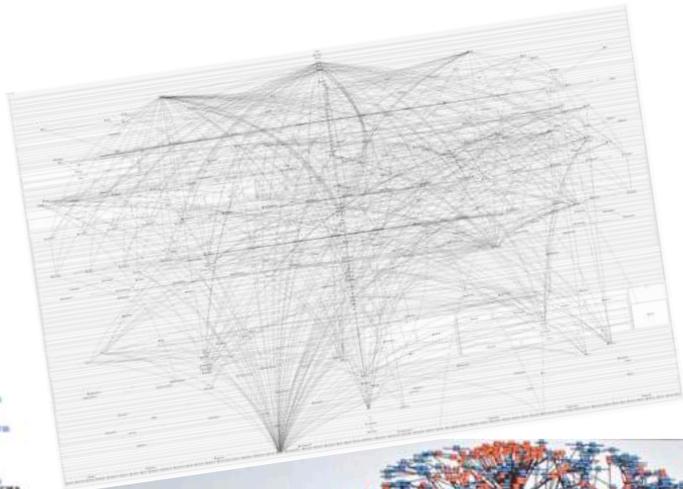


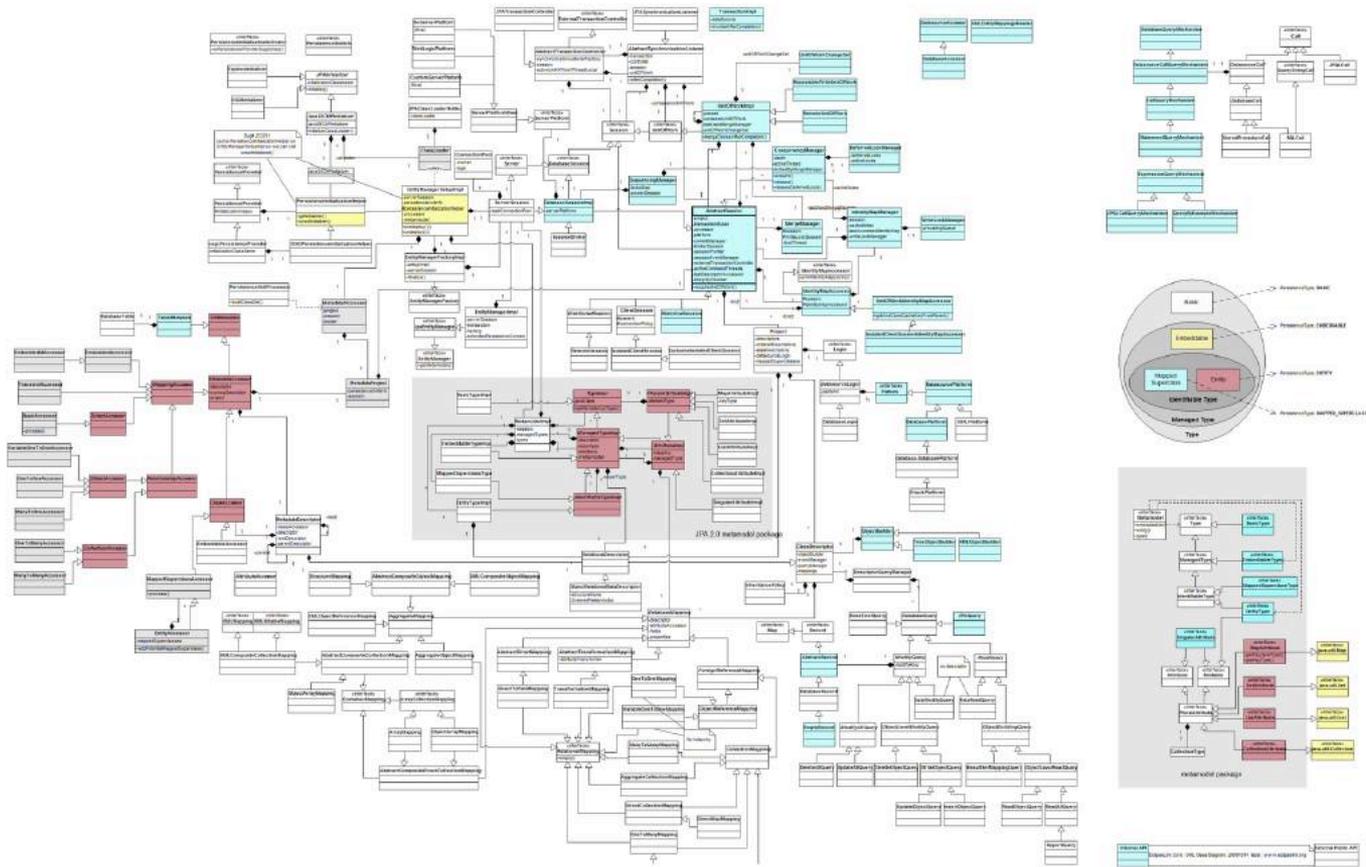
# Warum agile Vorhaben trotzdem reviewen

(sollten)



# Unsere Systeme...



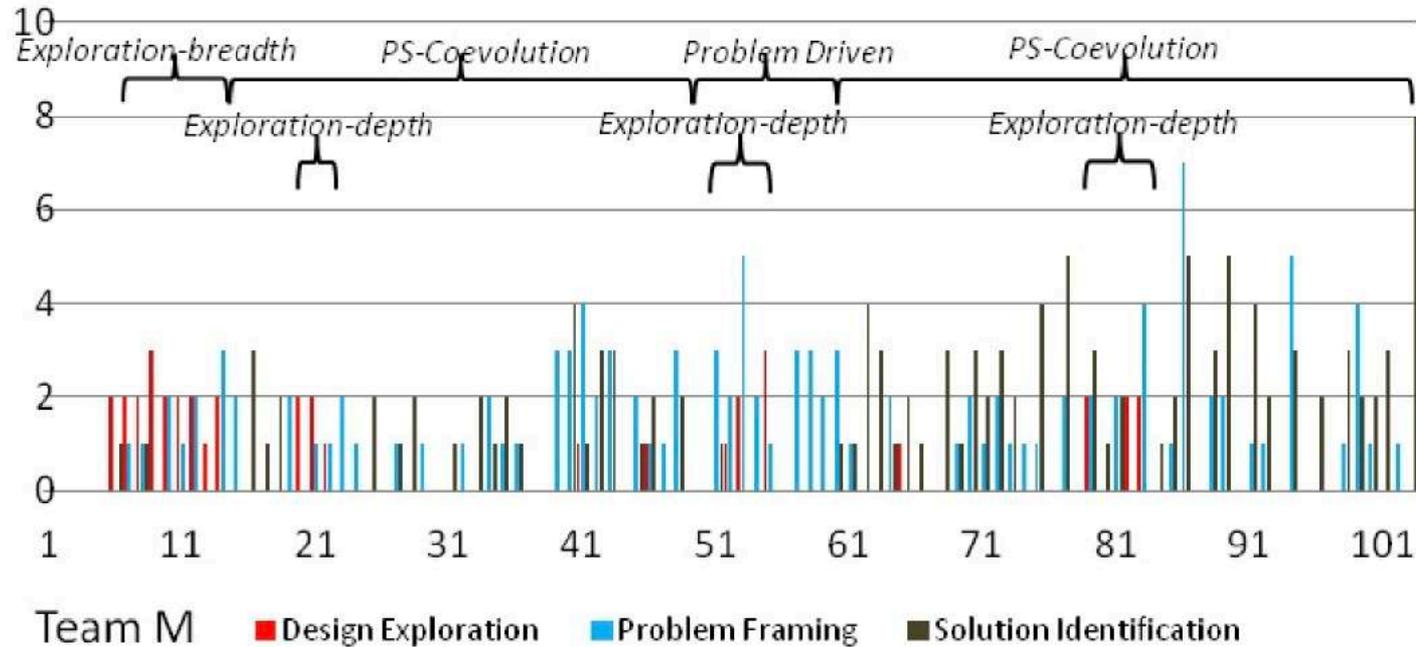


# Ist Softwareentwicklung kompliziert?

- + Dynamik in Umfeld & Lösungsraum...
- + (teilweise) Intransparenz...
- + Neuartigkeit...
- + Offenheit der Zielsituation...
- + Widersprüchlichkeiten...



# Wie wir Systeme designen



Aus: "Decision Making in Software Architecture", Hans van Vlieta, Antony Tang, 2015



# Wir handeln oft intuitiv

Anekdoten-  
getrieben

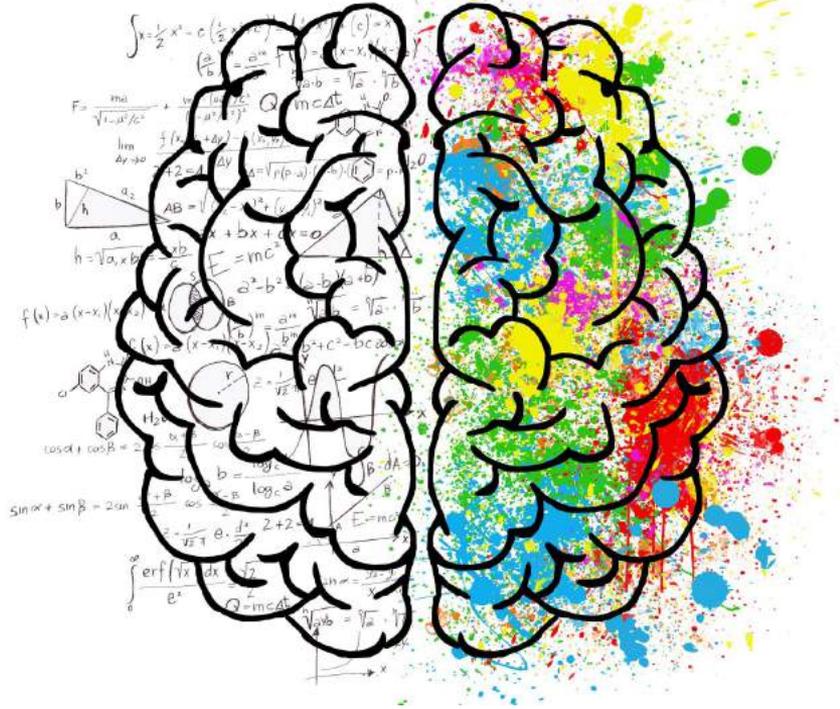
nach Bauchgefühl

Konsens-  
gesteuert

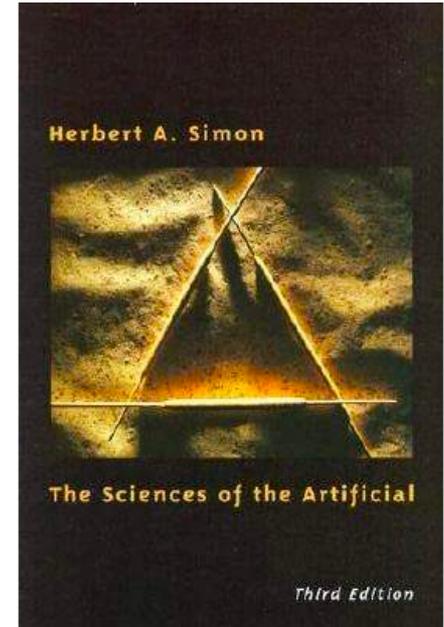
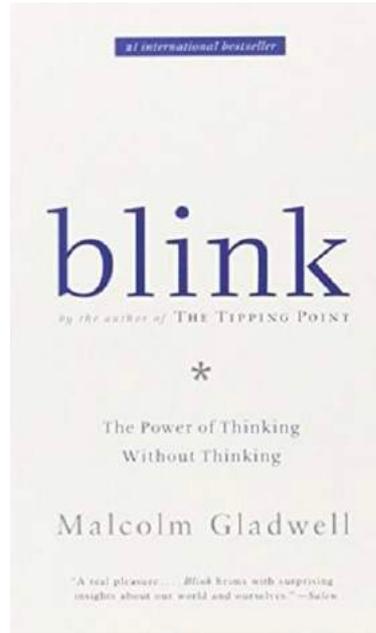
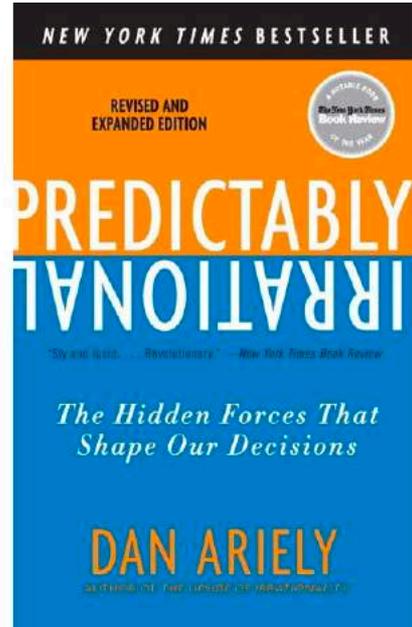
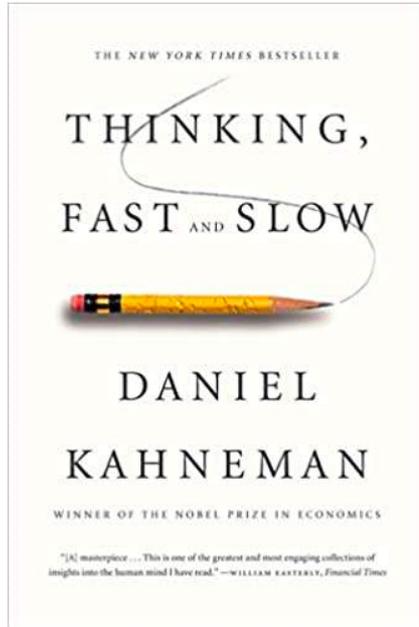
emotional

moralgetrieben

erfahrungsbasiert



# Wie Entscheidungen funktionieren...



**Wir agieren nicht  
isoliert...**

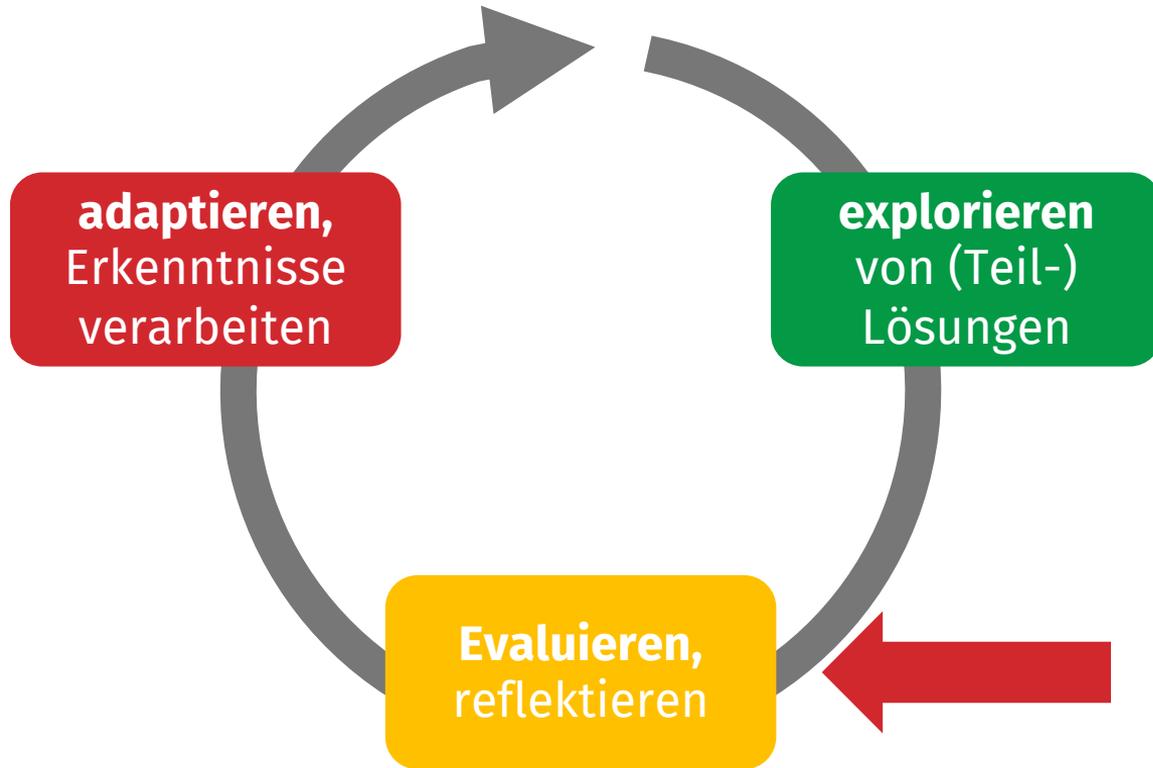


A large, industrial building under demolition or renovation. The structure is made of brick and steel, with many windows missing. The interior is filled with rubble and debris. A person is standing in the center of the debris field, providing a sense of scale. The lighting is dramatic, with strong shadows and highlights.

**Architekturarbeit ist unübersichtlich  
(und muss es sein)**

**Architekturbewertung ist kein „Gate“,  
sondern ein iteratives „WTF!?“**

# Arbeitsprozess für komplexe Probleme



# Bewertungsmethoden

Die in agilen Kontexten funktionieren



# Bewertungsmethoden

## ATAM

Architecture Tradeoff Analysis Method

## ARID

Architecture Review for Intermediate Designs

## CBAM

Cost-Benefit Analysis Method

## DCAR

Decision Centric Architecture Review

## PBAR

Pattern Based Architecture Review

## DASE

Decision and Scenario based architecture evaluation

## TARA

Tiny Architecture Review Approach

...



# Entscheidungs- Dokumentation

- Problem
- Detailbeschreibung
- Betrachtete Alternativen
- Forces für
- Forces gg. die Entscheidung

|                                     |  |   |  |  |
|-------------------------------------|--|---|--|--|
| Name                                | Redundancy of controllers  |   |  |  |
| Problem                             | The application should run even if the server fails  |   |  |  |
| Solution or description of decision | The system is deployed to two servers: one is active, the other one is inactive. The active server provides all system services, while the passive one is running in the background. When the active server fails, the inactive server becomes active. During the switch over, the active server tries to update the passive one to make sure that it has the same data and status. Both servers have an identical software configuration. This solution follows the <i>Redundant Functionality Pattern</i> .  |   |  |  |
| Considered alternative solutions    | Apply the <i>Redundancy Switch Pattern</i> : Both servers are active; external logic is used to decide which output is actually used in the control. In this case, cyclic data copying could be avoided. However, applying this solution would require major modifications to the system. Even though availability would be increased, it would also cause additional costs. The customers are not prepared for paying more for higher availability. Additionally, the external logic component could become a potential single point of failure. Therefore, this alternative was discarded. |   |  |  |
| Forces in favor of decision         | <ul style="list-style-type: none"> <li>• Easier to implement than the alternative solution</li> <li>• Scales easily to versions where redundancy is not used</li> <li>• No additional costs</li> </ul>   |   |  |  |
| Forces against the decision         | <ul style="list-style-type: none"> <li>• Slower switch over time than the alternative would have</li> <li>• Hard to offer higher availability than the current 99.99%</li> </ul>   |   |  |  |
| Outcome                             | Green  | Yellow  | Yellow   | Red  |
| Rationale for outcome               | Current solution seems to be ok.   | I am concerned about the slow switch over time. | Widely accepted solution. Availability might become a problem in the future. | We should really reconsider this decision, as the next release is likely to have higher availability requirements. |

7-10x



# Architecture Decision Records

**Titel (*title*):** Kurze Beschreibung der Entscheidung

**Status:** „proposed“, „accepted“, „discarded“, „deprecated“ oder „superseded“

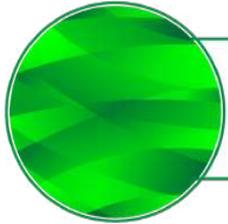
**Kontext (*context*):** Kräfte/Einflussfaktoren, zur Verfügung stehenden Optionen

**Entscheidung (*decision*):** Faktenbasierte Beschreibung der Entscheidung

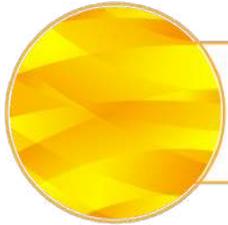
**Konsequenzen (*consequences*):** Auflistung aller positiven / negativen / neutralen Konsequenzen, ggf. Metriken der Architectural Fitness



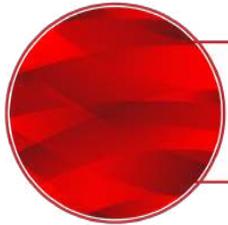
# Einschätzung und Argumente



Ich bin OK mit  
der Lösung



Ich sehe ausräumbare  
Risiken (akzeptabel)



Die Lösung gefährdet  
unser Produkt

- 15-20min pro Entscheidung
- Kräfte für und gegen die Entscheidung betrachten
- In der Diskussion werden Risiken und Probleme notiert
- Jeder Teilnehmer vergibt Ampelfarbe

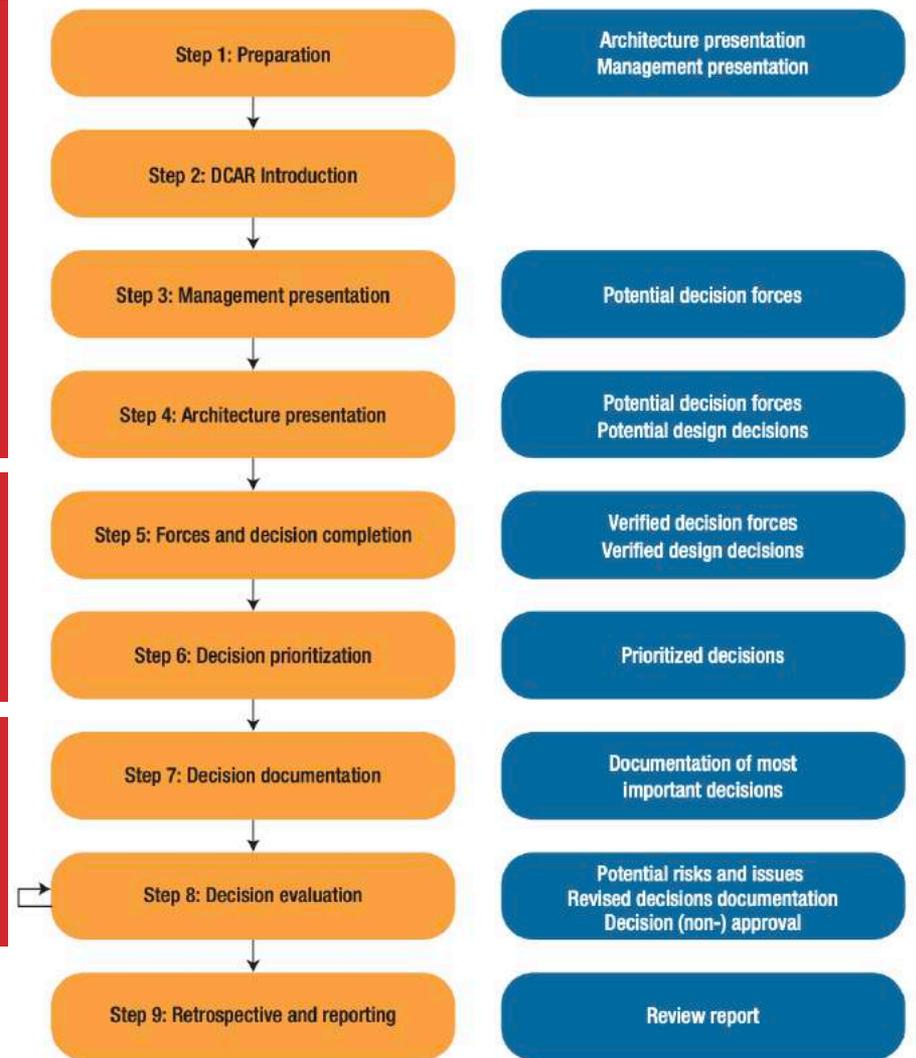


# DCAR Ablauf im Detail

Einstieg / Framing

Entscheidungs-Landkarte

Entscheidungsdoku & Review



# Ergebnisse eines DCARs

1

Wichtige  
**Einflüsse**  
("forces")

4

**Umstrittene Punkte** inkl.  
Ansprechpartner (Risiken)

2

**Ankerpunkte** der Architektur  
(sehr zentrale Entscheidungen)

5

**Schwächen** der  
Lösung (Probleme)

3

Gesicherte  
**Entscheidungen**

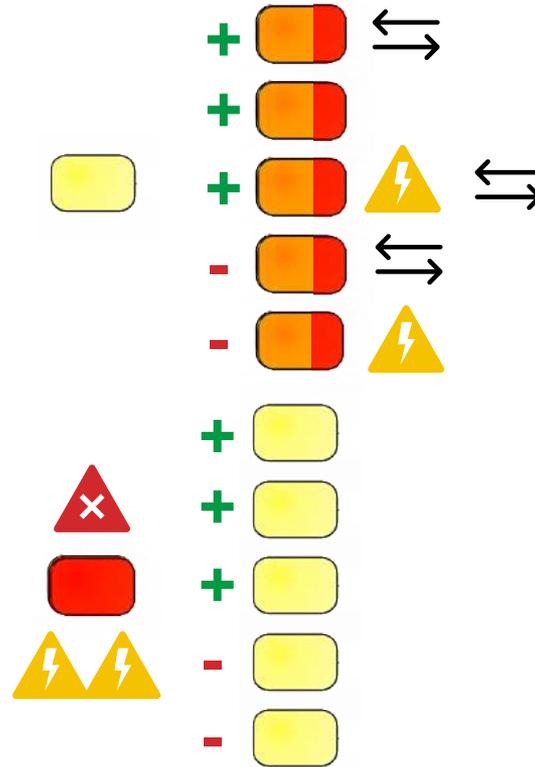
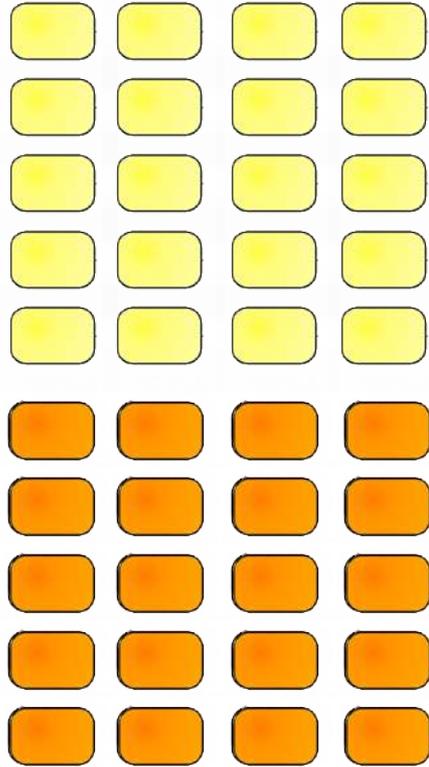
6

**Dokumentation**  
(ADRs)



# Wann steht das Ergebnis fest?

DCAR ATAM



-  Tradeoff
-  Risiko
-  Problem
-  „Force“
-  Entscheidung



# Bewertungsmethoden

## **ATAM**

Architecture Tradeoff Analysis Method

## **ARID**

Architecture Review for Intermediate Designs

## **CBAM**

Cost-Benefit Analysis Method

## **DCAR**

Decision Centric Architecture Review

## **PBAR**

Pattern Based Architecture Review

## **DASE**

Decision and Scenario based architecture evaluation

## **TARA**

Tiny Architecture Review Approach

## **STAR**

Scalable Technique for Architecture Reviews



# Qualitätsziele grob zusammenfassen



“Amazon is customer obsessed! If only one customer complains, we take the feedback and improve the system”

The Netflix logo, consisting of the word "NETFLIX" in a bold, red, uppercase, sans-serif font, centered within a light gray rectangular background.

NETFLIX

“Netflix-Members are able to watch tv series and films – as much as they want, any time, everywhere, on every internet-connected device out there.”



“Available everywhere, Great user experience, More convenient than piracy, Fast, reliable, always available, Scalable for many, many users.”





# Einschätzung: high / medium / low

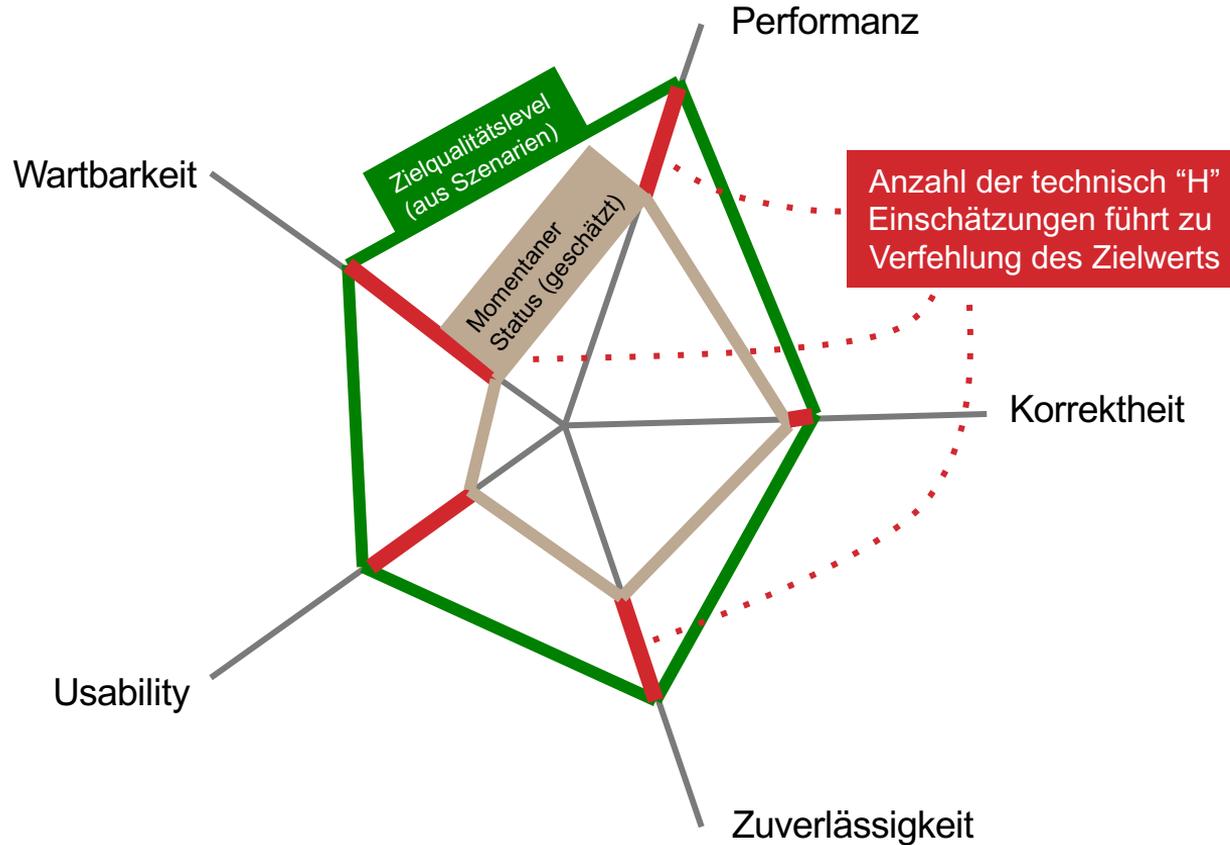
Szenarien mit hoch/mittel/niedrig bewerten in:

- **Fachliche Wichtigkeit**
- **Technische Schwierigkeit / Risiko**

H/M/L

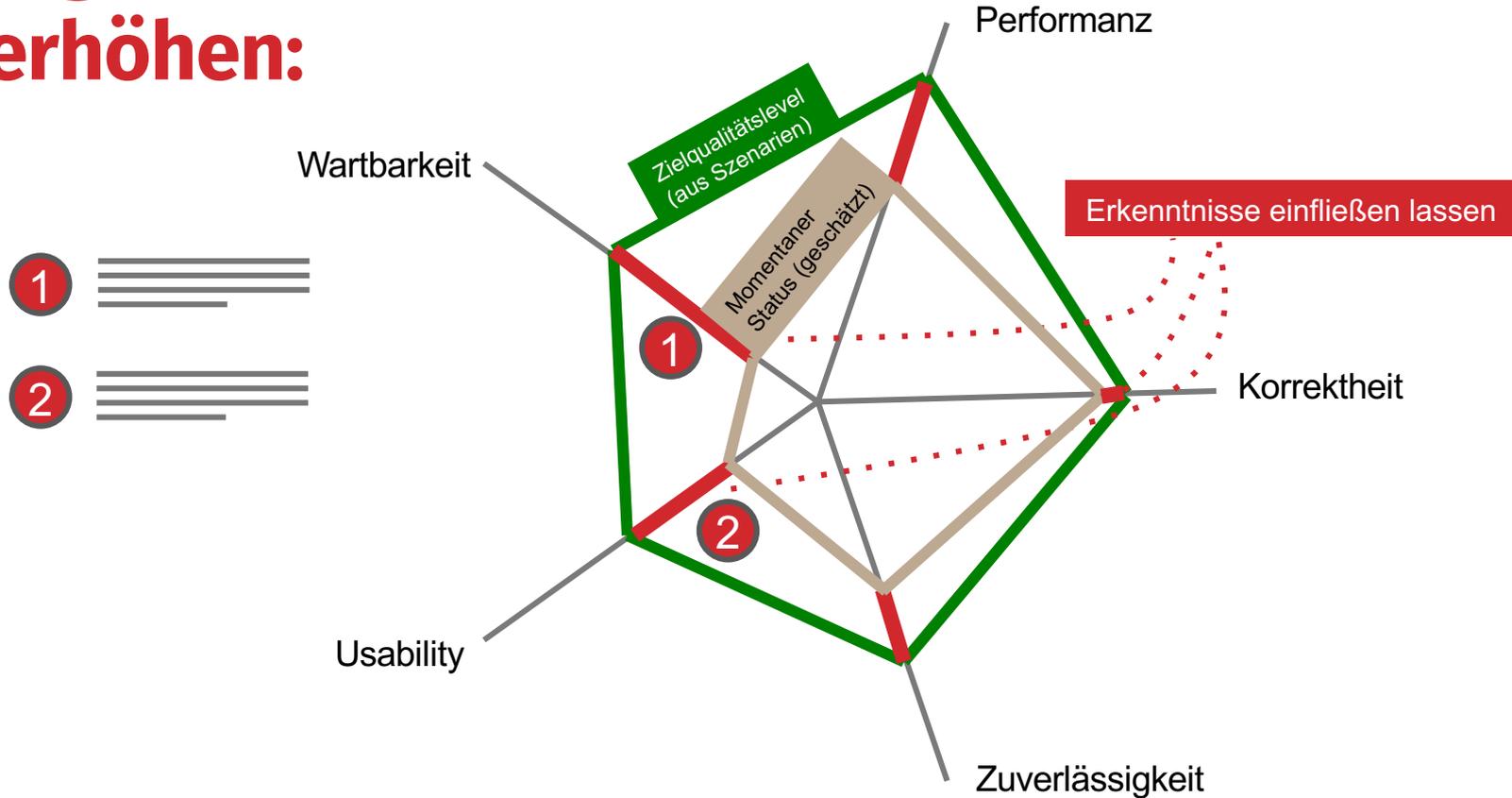


# Gesamtlücke schätzen:





# Ergebnis schärfen – Konfidenz und Tiefe erhöhen:



# Bewertungsmethoden

## ATAM

Architecture Tradeoff Analysis Method

## ARID

Architecture Review for Intermediate Designs

## CBAM

Cost-Benefit Analysis Method

## DCAR

Decision Centric Architecture Review

## PBAR

Pattern Based Architecture Review

## DASE

Decision and Scenario based architecture evaluation

## TARA

Tiny Architecture Review Approach

## Pre-Mortem

Risiko-Brainstorming and Mitigation



# Das Pre-Mortem



Hier ruht das so erfolgreich  
begonnene Projekt X...



# Das Grundprinzip



# Pre Mortem Ablauf

1



10 min

## Pre-Mortem einführen

Methode kurz vorstellen, Modus erklären

2



20 min

## Gründe des Scheiterns

Brainstorming: Was ist passiert?  
Clustern

3

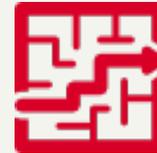


30 min

## Risiken priorisieren

%, € und: Was ist technisch beeinflussbar?

4



60 min

## Mitigierungsstrategien

Konkrete technische/methodische Verbesserungen



# Pre Mortem – Der Schnelleinstieg

1



10 min

## Pre-Mortem einführen

Methode kurz vorstellen, Modus erklären

2



20 min

## Gründe des Scheiterns

Brainstorming: Was ist passiert?  
Clustern

3

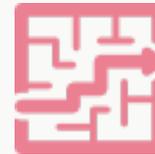


30 min

## Risiken priorisieren

%, € und: Was ist technisch beeinflussbar?

4



60 min

## Mitigierungsstrategien

Konkrete technische/methodische Verbesserungen





# Pre Mortem – Gründe des Scheiterns

1



10 min

**Pre-Mortem einführen**  
Methode kurz vorstellen, Modus erklären

2



20 min

**Gründe des Scheiterns**  
Brainstorming: Was ist passiert?  
Clustern

3



30 min

**Risiken priorisieren**  
%, € und: Was ist technisch beeinflussbar?

4



60 min

**Mitigierungsstrategien**  
Konkrete technische/methodische Verbesserungen



# Pre Mortem – Risiken priorisieren

1



10 min

## Pre-Mortem einführen

Methode kurz vorstellen, Modus erklären

2



20 min

## Gründe des Scheiterns

Brainstorming: Was ist passiert?  
Clustern

3

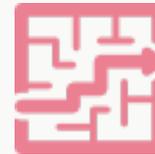


30 min

## Risiken priorisieren

%, € und: Was ist technisch beeinflussbar?

4



60 min

## Mitigierungsstrategien

Konkrete technische/methodische Verbesserungen



# realistisch!

Was ist wahrscheinlich ein Problem?

# schmerzhaft!

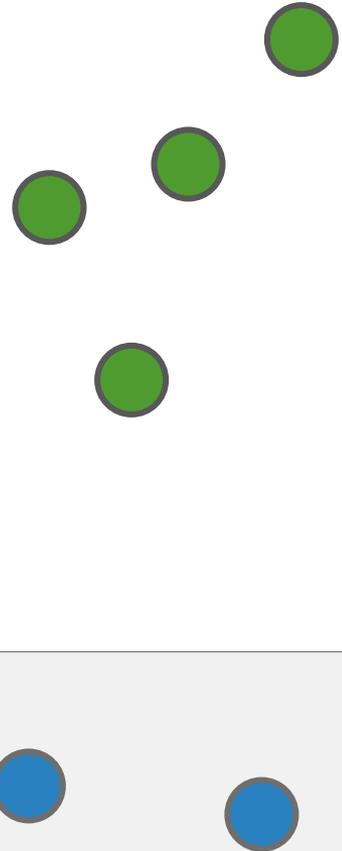
Was hat große Auswirkungen?

# behandelbar!

Was hat zumindest einen kleinen Hebel?

# kreativ!

Was gewinnt den Preis für bester Einfall?



# Pre Mortem – Mitigierungsstrategien

1



10 min

**Pre-Mortem einführen**  
Methode kurz vorstellen, Modus erklären

2



20 min

**Gründe des Scheiterns**  
Brainstorming: Was ist passiert?  
Clustern

3



30 min

**Risiken priorisieren**  
%, € und: Was ist technisch beeinflussbar?

4



60 min

**Mitigierungsstrategien**  
Konkrete technische/methodische Verbesserungen



# Was?

Was können wir konkret tun?

# Wer?

Wer kümmert sich darum?

# Wann?

Wann setzen wir diese Aktionen?

Wann betrachten wir erneut?

4  60 min

**Mitigierungsstrategien**  
Konkrete technische/methodische  
Verbesserungen



# Im Überblick...

Was funktioniert in agilen Vorhaben?



# Angesprochene Möglichkeiten

Entscheidungsorientiert

## DCAR

Decision Centric Architecture Review

**Initiale Dauer:** 4-8h

**Iterative Dauer:** 1-2h

**Empfohlene Häufigkeit:** Jede (2.) Iteration

Qualitätsorientiert

## STAR

Scalable Technique for Architecture Reviews

**Initiale Dauer:** 4h

**Iterative Dauer:** 2-3h

**Empfohlene Häufigkeit:** Jede 2.-4. Iteration

Risikoorientiert

## Pre-Mortem

Risiko-Brainstorming and Mitigation

**Initiale Dauer:** 2h

**Iterative Dauer:** 2h

**Empfohlene Häufigkeit:** Jede (2.) Iteration





# Weiterführendes

Wo geht's weiter?

# Training iSAQB® CPSA®-Advanced **ARCEVAL** — 2 Tage

## Trainer: Stefan Toth



SCAN ME

### Teilnehmerstimmen:

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



Nächster Termin:  
16.-17. Feb 2023  
[socreatory.com](https://socreatory.com)

**Lernen von den Besten.**



# Danke.

Jegliche Fragen sind willkommen!

 stefan.toth@embarc.de

 @st\_toth

 xing.to/sto

  
embarc   
Software Consulting GmbH

