

Stefan Zörner



Der Werkzeugkoffer für Architektur-Reviews - gestern, heute und morgen

Vortrag auf den IT-Tagen 2025
IT-Konferenz für Software-Entwicklung, -Architektur, DevOps, KI, ...
Frankfurt am Main, den 10. Dezember 2025



0



embarc.de

Werkzeugkoffer für Architektur-Reviews

1

Abstract

Der Werkzeugkoffer für Architektur-Reviews - gestern, heute und morgen

Architekturbewertungen decken Risiken in Softwarelösungen auf und sichern Technologie- und Entwurfsentscheidungen ab. Das etablierte Instrumentarium an Methoden und Werkzeugen reicht von Workshop-basierten Durchsprachen à la ATAM bis hin zu Tool-gestützten Analysen des Quelltextes oder des laufenden Systems mit Metriken. Die Methoden sind teilweise seit 20+ Jahren im Praxiseinsatz, d.h. wir können auf einen reichen Erfahrungsschatz zurückgreifen. Gleichzeitig dreht die Welt sich weiter.

In diesem Vortrag bekommen die Teilnehmer einen historischen Abriss und einen Überblick über das ganze Arsenal. Ich diskutiere die jeweiligen Stärken und Schwächen der Ansätze bei unterschiedlichen Anlässen und in unterschiedlichen Kontexten. Die Teilnehmer sehen Beispielergebnisse aus Reviews und lernen, in welchen Konstellationen und Zeitpunkten im Software-Lebenszyklus sich welche Herangehensweise heute anbietet. Sie erfahren, was bereits AI-gestützt in diesem Themenfeld möglich ist, wo die Reise voraussichtlich hingeht, und wo ich die Grenzen dabei sehe.

Nach dem Vortrag können die Teilnehmer informiert darüber entscheiden, wie sie ein Softwaresystem bzw. dessen Lösungsarchitektur fundiert, angemessen und zeitgemäß beleuchten.

1



embarc.de

Werkzeugkoffer für Architektur-Reviews

2

Stefan Zörner

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

Schwerpunkte

- Methodische Softwarearchitektur (Entwurf, Bewertung, Dokumentation)
- Architektur-Reviews



2

DO.

Agenda

Der Werkzeugkoffer für Architektur-Reviews

01. Warum Softwarearchitektur reviewen?
02. Wie funktionieren Architektur-Reviews?
03. Was ist drin im Werkzeugkoffer?
04. Wie entscheide ich, welche(s) Werkzeug(e) ich verwende?
05. Ausblick und weitere Informationen




3

01.

Warum Softwarearchitektur reviewen?

- 01. Warum Softwarearchitektur reviewen?**
- 02. Wie funktionieren Architektur-Reviews?
- 03. Was ist drin im Werkzeugkoffer?
- 04. Wie entscheide ich, welche(s) Werkzeug(e) ich verwende?
- 05. Ausblick und weitere Informationen



4


embarc.de

Was ist eigentlich ein Review?

Unter einem Review (dt. Kritik, Rezension) verstehen wir ganz allgemein die Bewertung eines Gegenstandes.



Werkzeugkoffer für Architektur-Reviews
5

5



embarc.de

Werkzeugkoffer für Architektur-Reviews

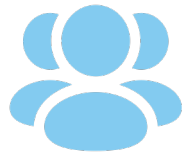
6

Spannend für jedes Review

Wer bewertet? Reviewer und deren Hintergrund

Was wird bewertet? Konkreter Betrachtungsgegenstand

Wann wird bewertet? Zeitpunkt des Reviews



6

Ein konkretes Beispiel: Die Planung einer Hochzeit ...



7



8



9



Warum Softwarearchitektur reviewen?

Die Motivation bei einem Architektur-Review ist vor allem

- **Risiken aufzudecken**

Risiko == Problem, das im weiteren Verlauf des Softwarevorhabens auftreten kann, aber nicht muss ...

- **Getroffene Entscheidungen abzusichern**

Sind wir auf dem richtigen Weg? Waren das gute Entscheidungen? Passen sie?



Softwarearchitektur



Umfasst die wichtigsten Entscheidungen für ein Softwaresystem.
Betrifft viele. Schwer zu ändern.

Themen für Architekturentscheidungen

- **Zerlegung**

Wie zerfällt unser System in kleine Teile?
Architekturstil, Modularisierung,
Abhängigkeiten ...

- **Technologie-Stack**

Was setzen wir ein?
Programmiersprachen, Frameworks,
Middleware, ...

- **Zielumgebung**

Wo läuft unsere Software (oder Teile davon)?
Client, On Premises, Cloud,
Verteilungsaspekte, Betrieb ...

- **Vorgehen**

Wie arbeiten wir?
Team-Schnitt, CI/CD, Ausliefern,
Testen, Updates, ...



Softwarearchitektur

Umfasst die wichtigsten Entscheidungen für ein Softwaresystem. Betrifft viele. Schwer zu ändern.



Themen für Architekturentscheidungen

- **Zerlegung**
Wie zerfällt unser System in kleine Teile?
Architekturstil, Modularisierung, Abhängigkeiten ...
- **Zielumgebung**
Wo läuft unsere Software (oder Teile davon)?
Client, On Premises, Cloud, Verteilungsaspekte, Betrieb ...
- **Technologie-Stack**
Was setzen wir ein?
Programmiersprachen, Frameworks, Middleware, ...
- **Vorgehen**
Wie arbeiten wir?
Team-Schnitt, CI/CD, Ausliefern, Testen, Updates, ...



Rahmenbedingungen

Vorgaben, die ein bestimmtes Softwaresystem einhalten muss.



Themen für Rahmenbedingungen

- **Organisatorische Aspekte**
z.B. Skills im Team, Budget, Termine, Regulatorik ...
- **Technische Aspekte**
z.B. Betriebssystem, Persistenztechnologien ...

schränken in der Regel ein

Softwarearchitektur

Umfasst die wichtigsten Entscheidungen für ein Softwaresystem. Betrifft viele. Schwer zu ändern.



Themen für Architekturentscheidungen

- **Zerlegung**
Wie zerfällt unser System in kleine Teile?
Architekturstil, Modularisierung, Abhängigkeiten ...
- **Zielumgebung**
Wo läuft unsere Software (oder Teile davon)?
Client, On Premises, Cloud, Verteilungsaspekte, Betrieb ...
- **Technologie-Stack**
Was setzen wir ein?
Programmiersprachen, Frameworks, Middleware, ...
- **Vorgehen**
Wie arbeiten wir?
Team-Schnitt, CI/CD, Ausliefern, Testen, Updates, ...



Qualitätsmerkmale (Begriffe nach ISO 25010:2023)

Funktionale Eignung (Functional Suitability)

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

Effizienz (Performance Efficiency)

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? ...

Benutzbarkeit (Interaction Capability)

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

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? ...

Übertragbarkeit (Flexibility)

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

Betriebssicherheit (Safety)

Sind Personen, Tiere, Sachen und Umwelt vor Schäden durch die Software geschützt? ...



Qualitätsmerkmale

Eigenschaften, die Softwaresysteme zu einem gewissen Grad haben können oder vermissen lassen.



★ Oberbegriffe für Softwarequalität

- Benutzbarkeit
- Betriebssicherheit
- Effizienz
- Funktionale Eignung
- Kompatibilität
- Sicherheit
- Wartbarkeit
- Zuverlässigkeit
- Übertragbarkeit

(nach ISO 25010:2023)

Rahmenbedingungen

Vorgaben, die ein bestimmtes Softwaresystem einhalten muss.



⚠ Themen für Rahmenbedingungen

- Organisatorische Aspekte
z.B. Skills im Team, Budget, Termine, Regulatorik ...
- Technische Aspekte
z.B. Betriebssystem, Persistenztechnologien ...

schränken in der Regel ein

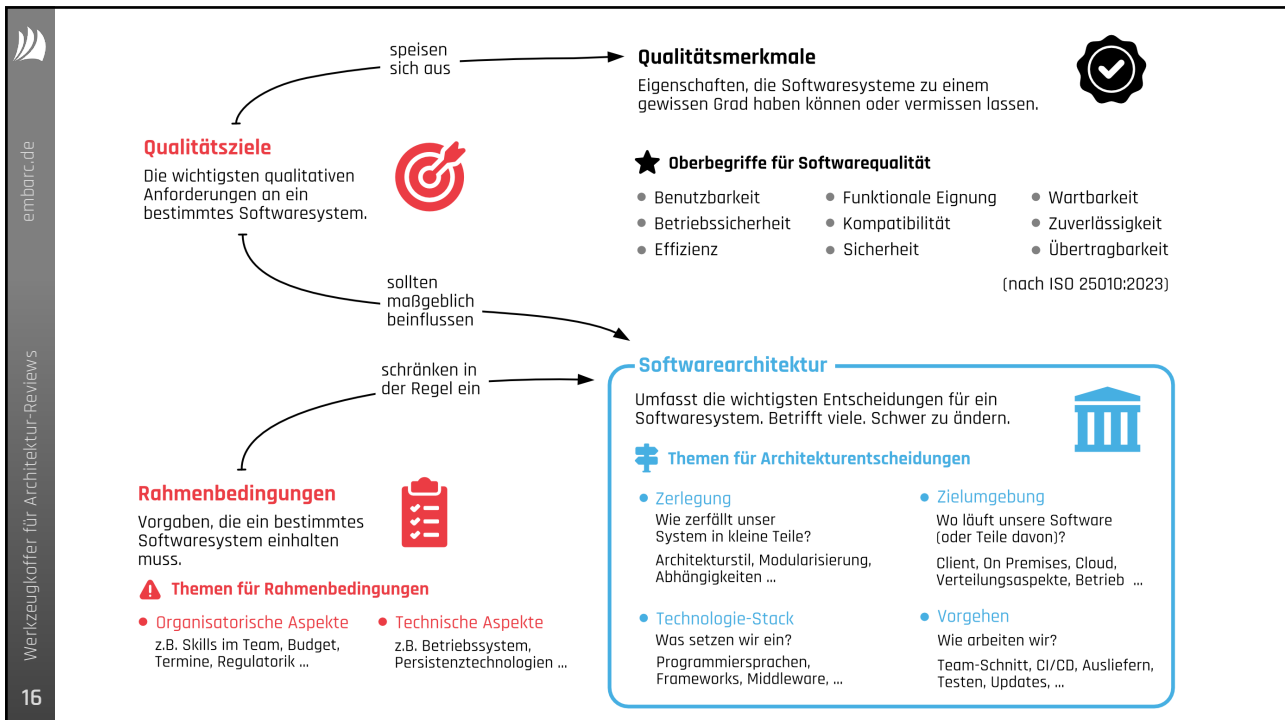
Softwarearchitektur

Umfasst die wichtigsten Entscheidungen für ein Softwaresystem. Betrifft viele. Schwer zu ändern.



⚙ Themen für Architekturentscheidungen

- Zerlegung
Wie zerfällt unser System in kleine Teile?
Architekturstil, Modularisierung, Abhängigkeiten ...
- Zielumgebung
Wo läuft unsere Software (oder Teile davon)?
Client, On Premises, Cloud, Verteilungsaspekte, Betrieb ...
- Technologie-Stack
Was setzen wir ein?
Programmiersprachen, Frameworks, Middleware, ...
- Vorgehen
Wie arbeiten wir?
Team-Schnitt, CI/CD, Ausliefern, Testen, Updates, ...




16

02.

Wie funktionieren Architektur-Reviews?

01. Warum Softwarearchitektur reviewen?
02. Wie funktionieren Architektur-Reviews?
03. Was ist drin im Werkzeugkoffer?
04. Wie entscheide ich, welche(s) Werkzeug(e) ich verwende?
05. Ausblick und weitere Informationen



17



Abgleichen ...



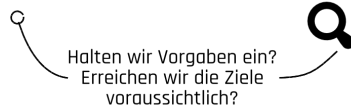
Architekturrelevante Anforderungen

Rahmenbedingungen,
Qualitätsziele

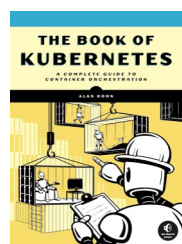
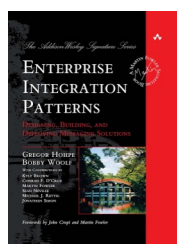
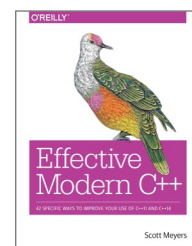
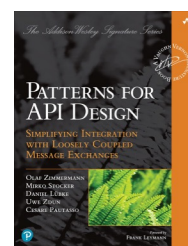
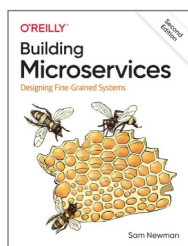


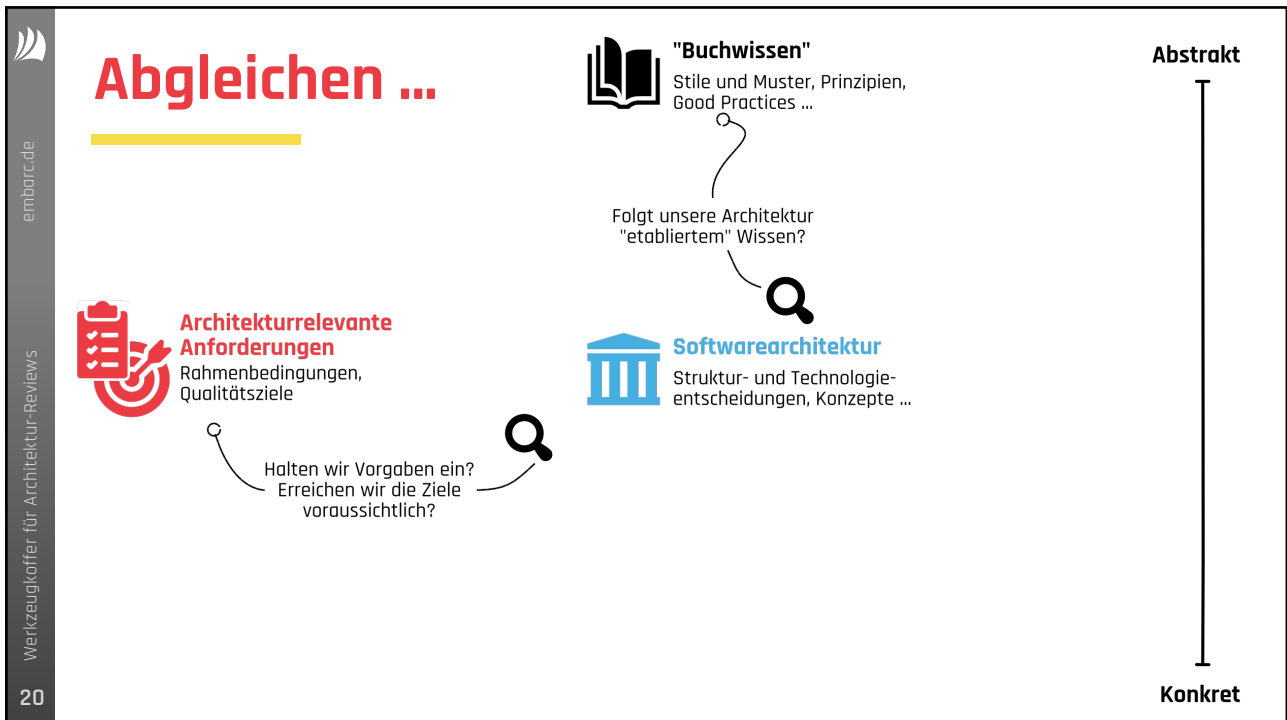
Softwarearchitektur

Struktur- und Technologie-
entscheidungen, Konzepte ...



„Buchwissen“





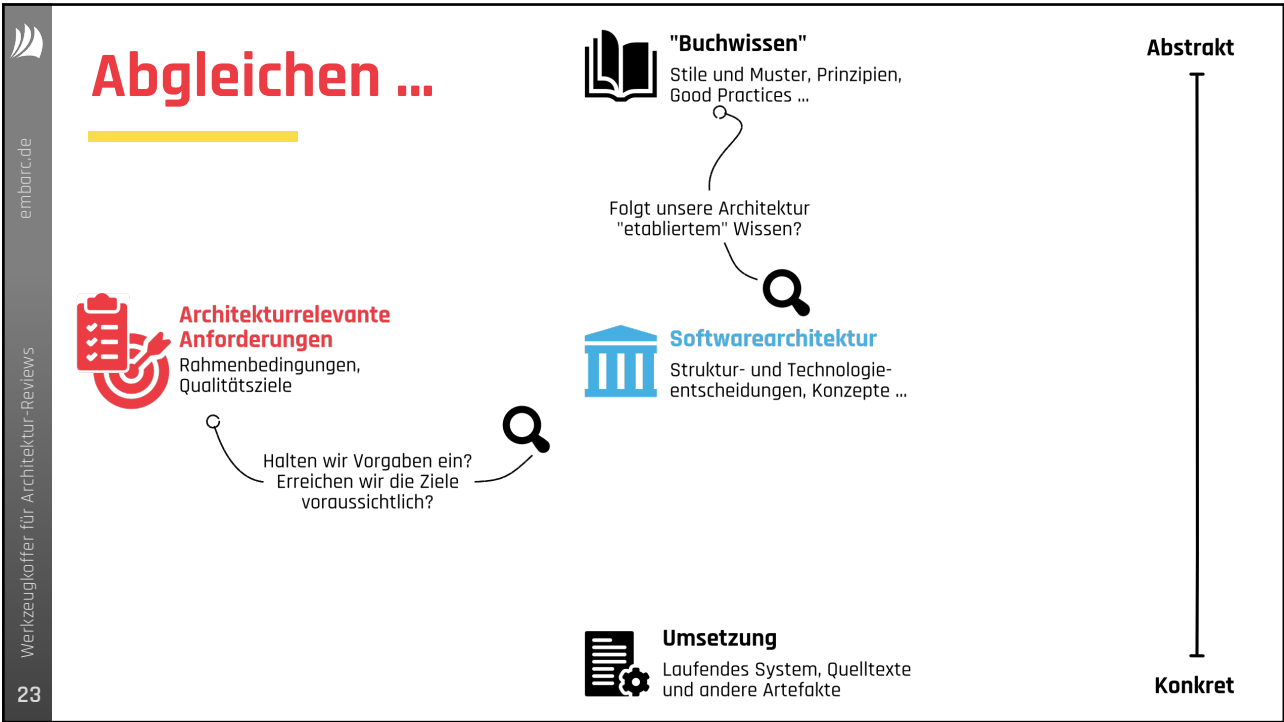
20



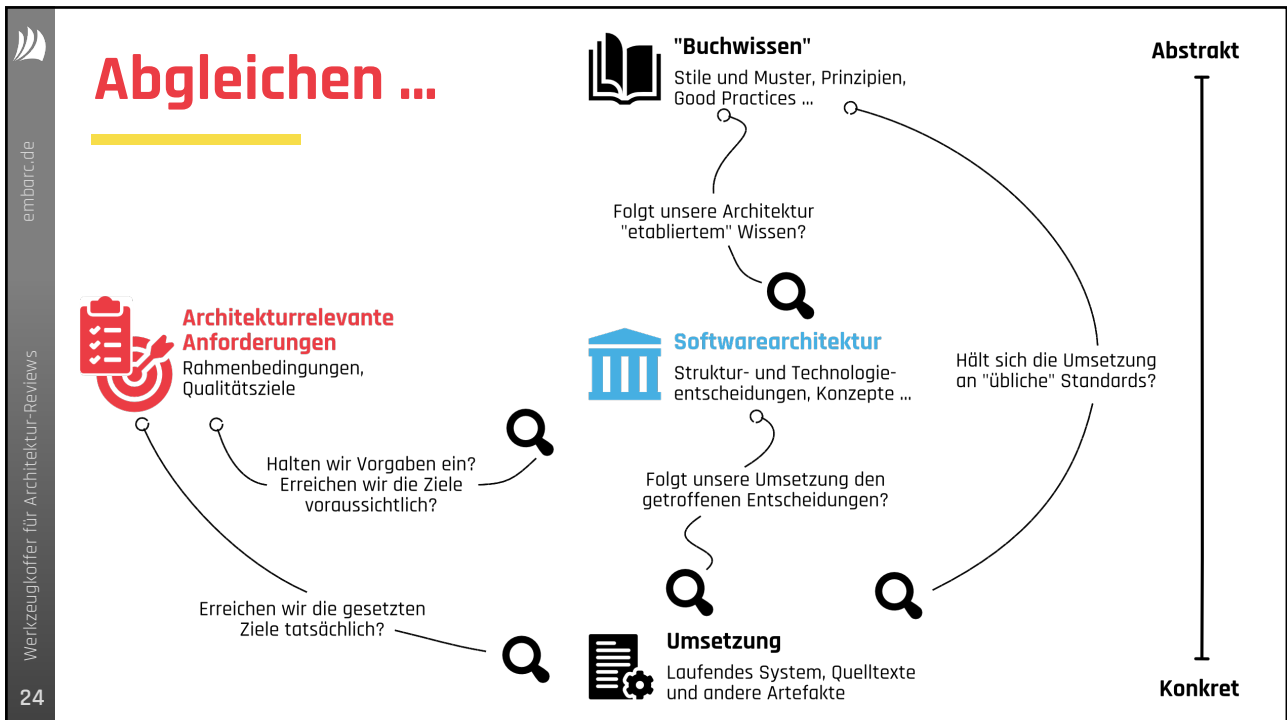
21



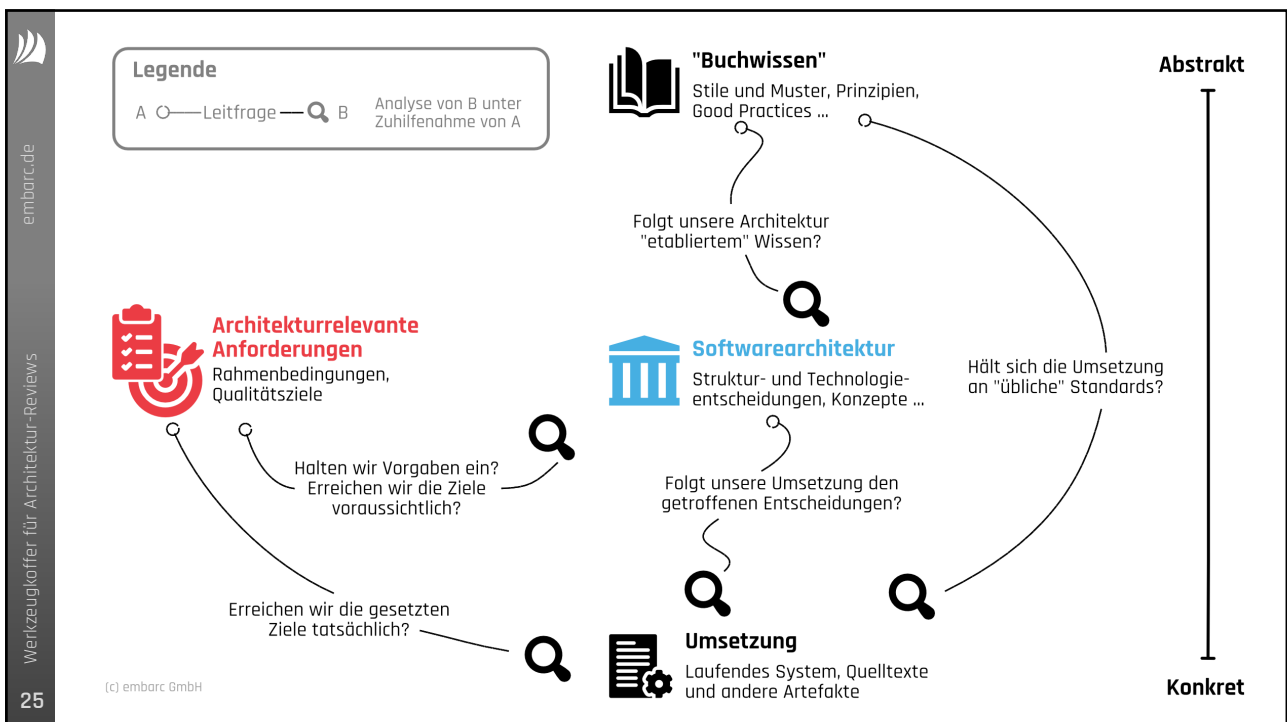
22



23



24



25



Qualitativ vs. quantitativ



Qualitative Analyse

Setzt auf Diskussion, Austausch und Durchsprachen. Oft **Workshop-basiert.**



Quantitative Analyse

Setzt auf Messungen und Metriken. In der Regel **Tool-basiert.**

Qualitative Analyse setzt auf Diskussion, Austausch und Durchsprachen. Oft Workshop-basiert.





Qualitative Analyse



Übliche Inputs

- Architekturmodelle und -dokumentation
- Architekturrelevante Anforderungen
- Wissen und Erfahrung der Beteiligten



Typische Werkzeuge und Hilfsmittel

- Interviews, Fragetechniken
- Qualitätsszenarien
- Priorisierung und Fokussierung
- Templates zur Moderation und Ergebnissicherung

Quantitative Analyse setzt auf Messungen und Metriken. In der Regel Tool-basiert.





Quantitative Analyse



Übliche Inputs

- Im Betrieb erhobene Daten (z.B. Log-Files)
- Quelltexte, Versionsverwaltung
- Laufendes System (z.B. Antwortzeiten)



Typische Werkzeuge und Hilfsmittel

- Dynamische Analysen (z.B. Last- und Performance-Tests)
- Statische Code-Analyse (z.B. Abhängigkeiten, Komplexität)
- Logging und Tracing, Monitoring



Vor- und Nachteile



Qualitative Analyse

Stärken

- + beteiligte Stakeholder gut einbindbar
- + vergleichsweise früh anwendbar
- + passt auf alle Qualitätsmerkmale

Schwächen

- ▼ Je nach Anzahl Beteiligter hoher Aufwand
- ▼ Moderation der Workshops herausfordernd
- ▼ Mitunter bleibt ein Restrisiko ("Bauchgefühl")



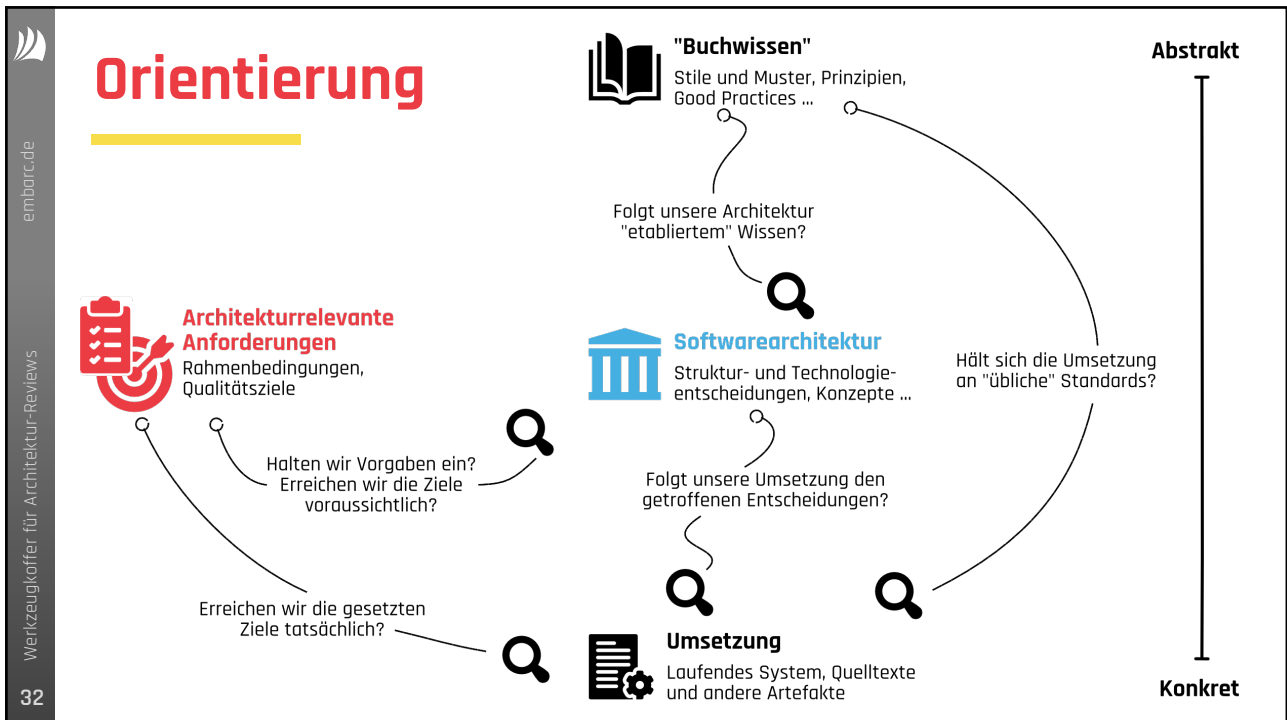
Quantitative Analyse

Stärken

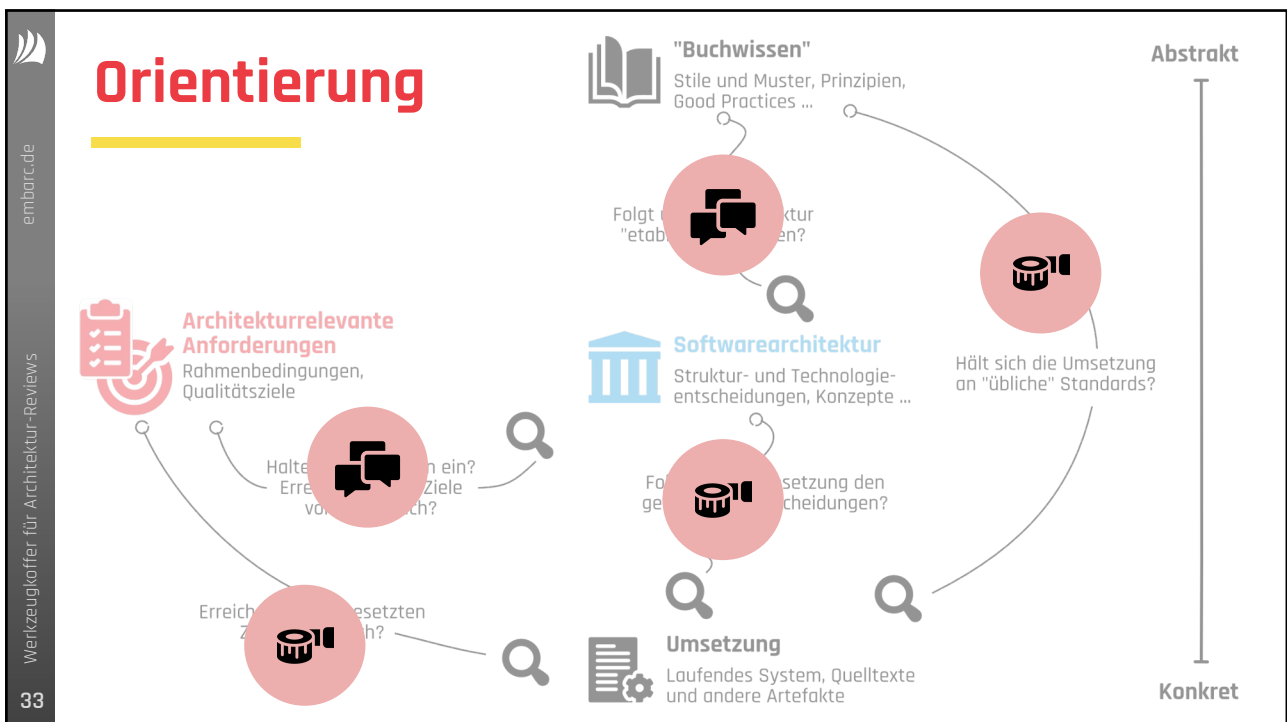
- + gut automatisierbar und leicht wiederholbar
- + wenig Bauchgefühl, Zahlen sind gute Argumente

Schwächen

- ▼ Messungen können nicht alle Qualitätsmerkmale gut erfassen
- ▼ Gefahr der Missdeutung und Fehlleitung



32




33

03.

Was ist drin im Werkzeugkoffer?

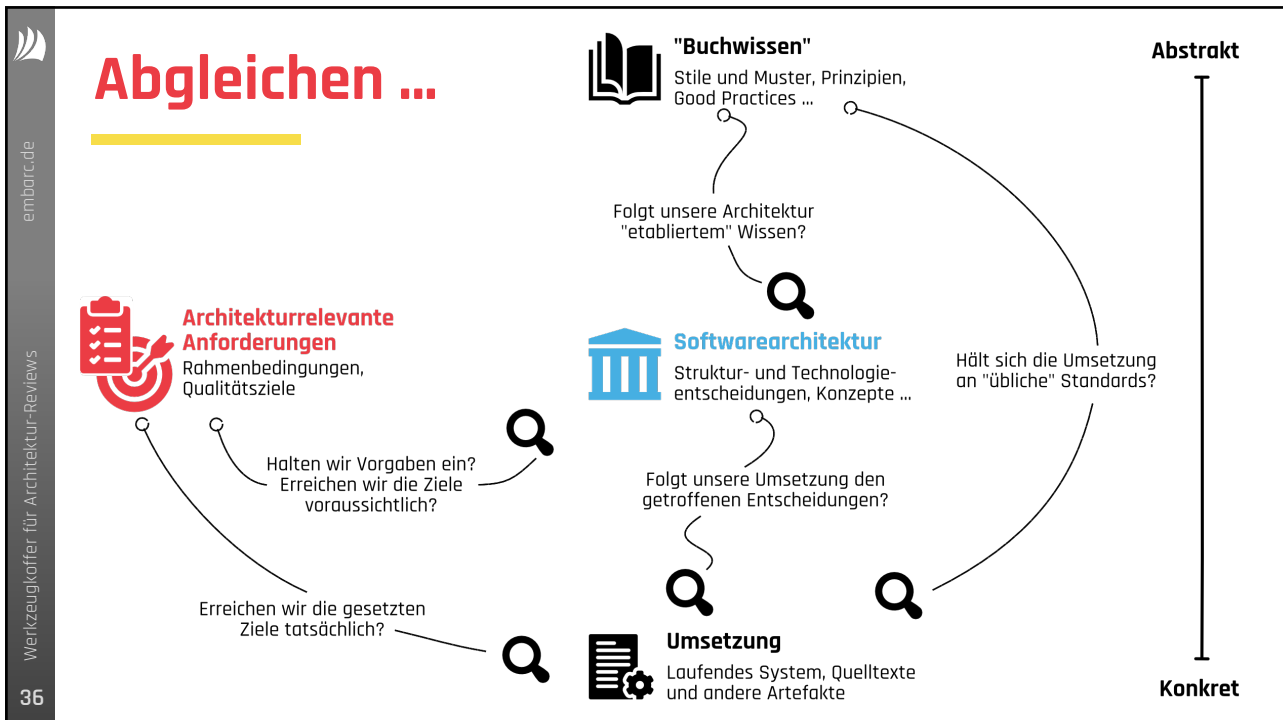
01. Warum Softwarearchitektur reviewen?
02. Wie funktionieren Architektur-Reviews?
- 03. Was ist drin im Werkzeugkoffer?**
04. Wie entscheide ich, welche(s) Werkzeug(e) ich verwende?
05. Ausblick und weitere Informationen



34



35



36

Review-Tätigkeiten

Tätigkeiten, welche die Abgleiche konkretisieren, vor- und nachbereiten. Und für die es auch passendes Handwerkszeug gibt.

- Review-Ziel **fixieren** und **absichern**
- Maßgebliche Stakeholder **identifizieren**
- Review strukturieren, **planen** und **durchführen**
- Architekturrelevante Anforderungen **sichten** und **präzisieren**
- Lösungsstrategie und -ansätze **kommunizieren** (ggf. **identifizieren**)
- Stärken und Schwächen **herausarbeiten**
- Kompromisse **sichtbar machen**
- Risiken **aufdecken** und **bewerten**
- Erkenntnisse **aufbereiten** und **verdichten**
- Ergebnisse **kommunizieren**

37



38

Themenbereiche für Werkzeuge

Review-Methoden

Veröffentlichte Vorgehensweisen für Reviews mit Rollen, Phasen ...

Unterstützungsmaterialien

Hilfestellung beim Erarbeiten von Ergebnissen, z.B. in Form von Vorlagen, Check-Listen, Fragenkatalogen ...

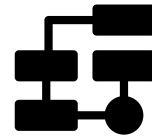
Software-Tools

Für Analyse-Tätigkeiten, vor allem im quantitativen Bereich.

39



Review-Methoden



Veröffentlichte Vorgehensweisen für Reviews mit Rollen, Phasen ...

Allgemeine Review-Methoden

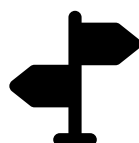
- Offen für alle Qualitätsmerkmale, System-Arten und Technologien
- Unterscheiden sich in Review-Ansatz und Aufwand



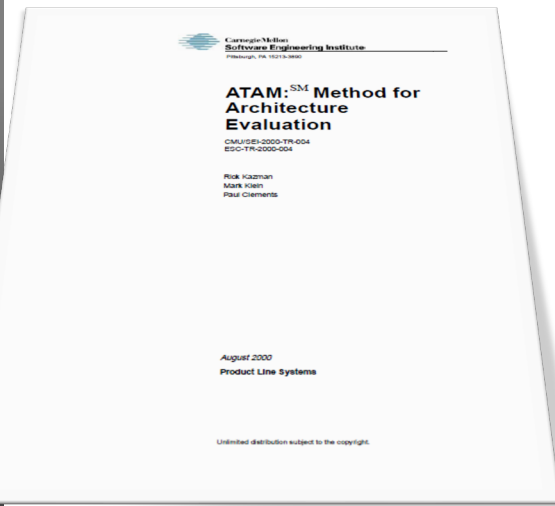
Allgemeine Review-Methoden

Beispiele für veröffentlichte + frei zugängliche Methoden

- **ATAM** - Architecture Tradeoff Analysis Method
- **DCAR** - Decision Centric Architecture Review
- **TARA** - Tiny Architecture Review Approach
- **LASR** - Lightweight Approach for Software Reviews
- ...



ATAM

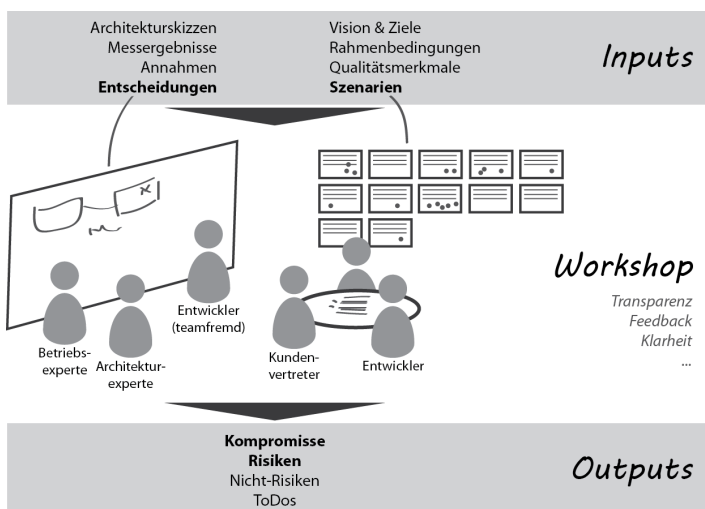


Architecture Tradeoff Analysis Method

- **Akademischer Ursprung:** Carnegie Mellon University, 2000
- Bekannteste Methode zur Bewertung von Softwarearchitektur
- **Qualitativer** Ansatz, Szenarien- und **Workshop**-basiert
- Fröh anwendbar, geht von den **Zielen** aus

42

Schematischer Review-Workshops

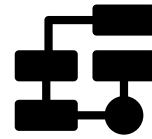


Quelle der Abbildung: Stefan Toth, "Vorgehensmuster für Softwarearchitektur", Hanser

43



Review-Methoden



Veröffentlichte Vorgehensweisen für Reviews mit Rollen, Phasen ...

Allgemeine Review-Methoden

- Offen für alle Qualitätsmerkmale, System-Arten und Technologien
- Unterscheiden sich in Review-Ansatz und Aufwand

Spezifische Review-Methoden

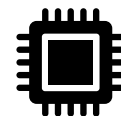
- Zugesnitten auf einen Aspekt (z.B. Kosten, bestimmtes Qualitätsmerkmal, Technologie)



Spezifische Review-Methoden

Beispiele für veröffentlichte + frei zugängliche Methoden

- **CBAM** - Cost-Benefit Analysis Method
- **OWASP** Software Assurance Maturity Model (Security)
- **AWS** Well Architected Framework (Cloud)
- ...



embarc.de
Werkzeugkoffer für Architektur-Reviews

Review-Methoden

Veröffentlichte Vorgehensweisen für Reviews mit Rollen, Phasen ...


Allgemeine Review-Methoden
Offen für alle Qualitätsmerkmale, System-Arten und Technologien.

ATAM Architecture Tradeoff Analysis Method
DCAR Decision Centric Architecture Review
TARA Tiny Architecture Review Approach
LASR Lightweight Approach for Software Reviews
...

Unterscheiden sich in Review-Ansatz und Aufwand.

Unterstützungsmaterialien

Hilfestellung beim Erarbeiten von Ergebnissen, z.B. in Form von Vorlagen, Check-Listen, Fragenkatalogen ...



Ablaufpläne für Reviews
Schritt für Schritt, jeweils mit Beteiligten, Inputs, (Zwischen-)ergebnissen ...

- Phasen, Schritte ... (z.B. aus den Methoden links im Koffer)

Vorlagen für Review-Erkenntnisse
Zum strukturierten Festhalten von Roh- oder Zwischenergebnissen, etwa für ...

- Szenariendurchsprachen (z.B. aus ATAM)
- Entscheidungen (z.B. aus DCAR)
- Vertiefungen (z.B. aus LASR)
- ...

Gliederungsvorschläge für Architekturbeschreibungen
Zur Aufbereitung des Betrachtungsgegenstandes oder Teilen davon.

- Architekturdokumentation (z.B. arc42)
- Prägnante Überblicke (z.B. Canvases)
- Architekturentscheidungen (ADRs)
- ...

Material für Software-Tool- und AI-Einsatz
Vorlagen, Setups, Inhalte, etwa für den Umgang mit LLMs oder für Agenten, z.B.

- Konfigurationen für Analyse-Tools
- System-Prompts und Prompt-Vorlagen
- RAG-Aufbereitungen
- ...

Sammlungen und Check-Listen
Ideegeber in Form von Listen, Tabellen, Baumstrukturen ... z.B.

- Qualitätsmerkmale (z.B. ISO 25010, Q42)
- Rahmenbedingungen (z.B. aus arc42)
- Risiken (z.B. aus LASR oder OWASP)
- Entwurfsprinzipien (allgemein oder Technologie-spezifisch, z.B. für Cloud)
- ...

Vorlagen für Review-Resultate
Zum Verdichten, zur Kommunikation und Präsentation der Review-Ergebnisse

- Ergebnisbericht (z.B. aus ATAM)
- Visualisierungen (z.B. aus LASR)
- Präsentationsvorlagen
- ...

46

embarc.de
Werkzeugkoffer für Architektur-Reviews

Unterstützungsmaterialien

Hilfestellung beim Erarbeiten von Ergebnissen, z.B. in Form von Vorlagen, Check-Listen, Fragenkatalogen ...

Ablaufpläne für Reviews
Schritt für Schritt, jeweils mit Beteiligten, Inputs, (Zwischen-)ergebnissen ...

- Phasen, Schritte ... (z.B. aus den eben genannten Methoden)

Gliederungsvorschläge für Architekturbeschreibungen
Zur Aufbereitung des Betrachtungsgegenstandes oder Teilen davon, z.B.:

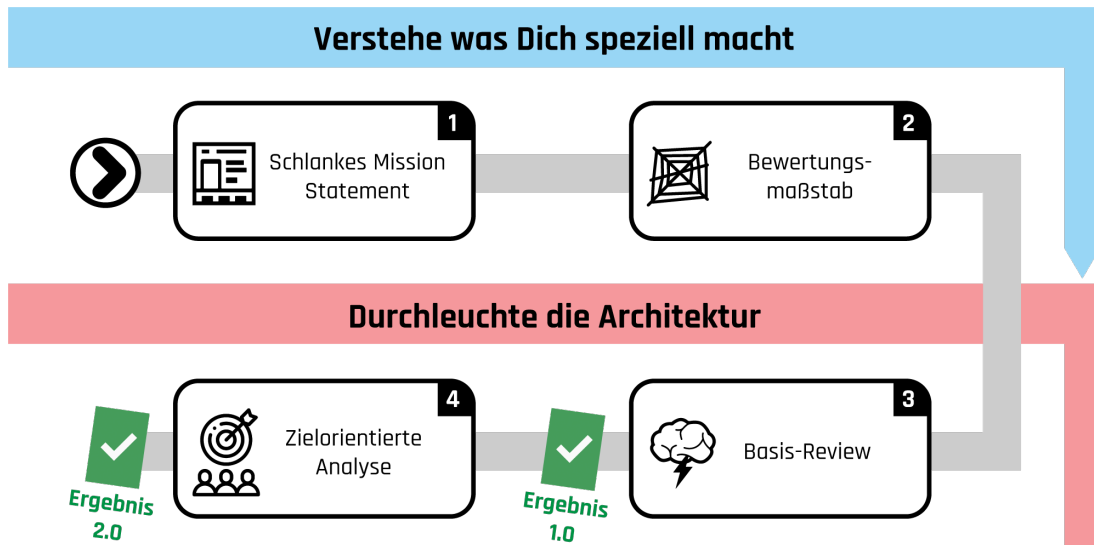
- Architekturdokumentation (z.B. arc42)
- Prägnante Überblicke (z.B. Canvases)
- Architekturentscheidungen (ADRs)



47



Beispiel für einen Ablauf: LASR



48



Unterstützungsmaterialien (2/3)



Sammlungen und Check-Listen

Ideengeber in Form von Listen, Tabellen, Baumstrukturen ... z.B.

- Qualitätsmerkmale (z.B. ISO 25010, Q42)
- Rahmenbedingungen (z.B. aus arc42)
- Risiken (z.B. aus LASR oder OWASP)



Vorlagen für Review-Erkenntnisse

Zum strukturierten Festhalten von Roh- oder Zwischenergebnissen, etwa für ...

- Szenarien-Durchsprachen (z.B. aus ATAM)
- Entscheidungen (z.B. aus DCAR)
- Vertiefungen (z.B. aus LASR)

49

Beispiel: Standard-Risiken aus LASR (Lightweight Approach für Software-Reviews) als Karten-Set.



50

Beispiel für eine ADR-Gliederung (DCAR)

Name der Entscheidung	Dezentrale Speicherung von Gruppen.		
Problemstellung	App-user sollen für ihren Austausch selbst Gruppen anlegen und pflegen können. Die hohe Sicherheit der direkten 1:1 Kommunikation in Threads soll dabei idealerweise erhalten bleiben.		
Lösung oder getroffene Entscheidung	Gruppen verwalten wir dezentral auf den Endgeräten. Gruppennachrichten werden vom Client in einzelne Direktnachrichten an die Gruppenmitglieder gesplittet und wie gewohnt Ende zu Ende verschlüsselt. Die Gruppengröße begrenzen wir, aus Lastgründen, z.B. auf 30.		
Betrachtete Alternative(n)	Gruppenverwaltung auf dem Server: Gruppennachrichten werden an den Server geschickt und von dort verteilt. (Neues Kommunikations-Pattern mit anderer Verschlüsselungsidee).		
Pro	<ul style="list-style-type: none">• Aufrechterhaltung der Ende-zu-Ende-Verschlüsselung auch in Gruppen.• Schutz der Privat-Sphäre durch dezentrale Speicherung der Gruppen.• geringer Konzeptions- und Umsetzungsaufwand		
Contra	<ul style="list-style-type: none">• Höherer Rechenaufwand auf Backend und Client (Vor-/Entschlüsseln) durch Versand vieler einzelner Nachrichten.• Nicht abgeholte Nachrichten gehen für "Nachzügler" in der Gruppe verloren.• Begrenzung der Gruppengröße schmälert Funktionsumfang		
Ergebnis der Diskussion	Anita: Grün... Aus meiner Sicht sehr gute Lösung, die Vertraulichkeit als unser höchstes Gut berücksichtigt. Laura: Gelb... Mir ist noch unklar, wie wir die Gruppen verlässlich synchronisieren. Auch, wie ein Mitglied eine Gruppe verlassen kann.	Peter: Gelb... Mir gefällt die Limitierung der Gruppengröße, ist das wirklich erforderlich? Welche Anzahl wäre dann noch gangbar? Michael: Grün... Passt.	Mrs. Rot... Für den Versand von Medien (z.B. Fotos) brauchen wir eine andere Lösung, da sie den Client zu sehr belastet.

Decision-Centric Architecture Review

- Ursprung: Universitäten von Groningen (NL), Tampere (FI), 2011
- Geht in der Analyse von den (Architektur-)Entscheidungen aus (zum Vergleich: ATAM steigt über die qualitativen Anforderungen ein)
- Produziert im Rahmen der Analyse dokumentierte Entscheidungen
- Hat dafür eine Vorlage parat

51



Unterstützungsmaterialien (3/3)



Material für Software-Tool- und AI-Einsatz

Vorlagen, Setups, Inhalte, etwa für den Umgang mit LLMs oder für Agenten, z.B.

- Konfigurationen für Analyse-Tools
- System-Prompts und Prompt-Vorlagen
- RAG-Aufbereitungen

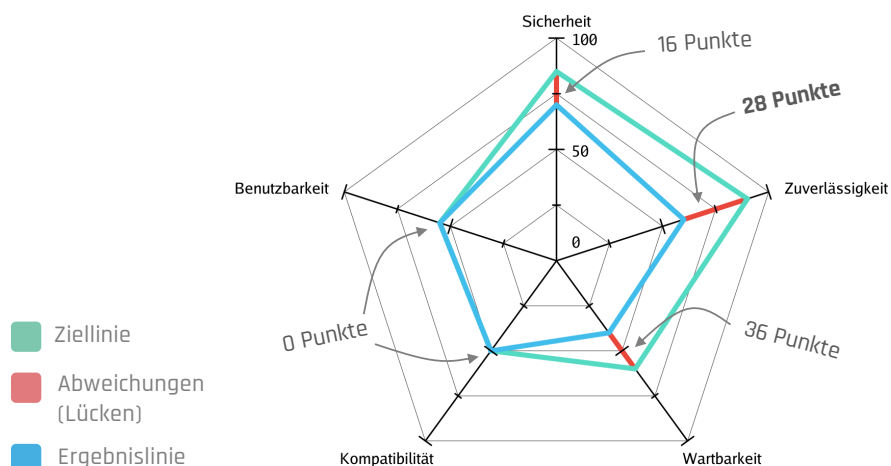
Vorlagen für Review-Resultate


Zum Verdichten, zur Kommunikation und Präsentation der Review-Ergebnisse, z.B.


- Ergebnisbericht (z.B. aus ATAM)
- Visualisierungen (z.B. aus LASR)
- Präsentationsvorlagen



Beispiel: LASR-Ergebnis-Diagramm




embarc.de



Review-Methoden

Veröffentlichte Vorgehensweisen für Reviews mit Rollen, Phasen ...

Allgemeine Review-Methoden

Offen für alle Qualitätsmerkmale, System-Arten und Technologien.

ATAM Architecture Tradeoff Analysis Method






DCAR Decision Centric Architecture Review

TARA Tiny Architecture Review Approach

LASR Lightweight Approach for Software Reviews

...

Unterscheiden sich in Review-Ansatz und Aufwand.

Spezifische Review-Methoden





Zugeschnitten auf einen Aspekt (z.B. Kosten, bestimmtes Qualitätsmerkmal, Technologie). Beispiele für solche Methoden bzw. Ressourcen, die Methodik beinhalten:

CBAM Cost-Benefit Analysis Method

OWASP Software Assurance Maturity Model (Security)

AWS Well Architected Framework (Cloud)

...

Unterstützungsmaterialien

Hilfestellung beim Erarbeiten von Ergebnissen, z.B. in Form von Vorlagen, Check-Listen, Fragenkatalogen ...

Ablaufpläne für Reviews

Schritt für Schritt, jeweils mit Beteiligten, Inputs, (Zwischen-)ergebnissen ...

- Phasen, Schritte ... (z.B. aus den Methoden links im Koffer)

Vorlagen für Review-Erkenntnisse

Zum strukturierten Festhalten von Roh- oder Zwischenergebnissen, etwa für ...

- Szenariendurchsprachen (z.B. aus ATAM)
- Entscheidungen (z.B. aus DCAR)
- Vertiefungen (z.B. aus LASR)

...

Gliederungsvorschläge für Architekturbeschreibungen

Zur Aufbereitung des Betrachtungsgegenstandes oder Teilen davon.

- Architekturdokumentation (z.B. arc42)
- Prägnante Überblicke (z.B. Canvases)
- Architekturentscheidungen (ADRs)

...

Material für Software-Tool- und AI-Einsatz

Vorlagen, Setups, Inhalte, etwa für den Umgang mit LLMs oder für Agenten, z.B.

- Konfigurationen für Analyse-Tools
- System-Prompts und Prompt-Vorlagen
- RAG-Aufbereitungen

...

Sammlungen und Check-Listen

Ideengeber in Form von Listen, Tabellen, Baumstrukturen ... z.B.

- Qualitätsmerkmale (z.B. ISO 25010, Q42)
- Rahmenbedingungen (z.B. aus arc42)
- Risiken (z.B. aus LASR oder OWASP)
- Entwurfsprinzipien (allgemein oder Technologie-spezifisch, z.B. für Cloud)

...

Vorlagen für Review-Resultate


Zum Verdichten, zur Kommunikation und Präsentation der Review-Ergebnisse


- Ergebnisbericht (z.B. aus ATAM)
- Visualisierungen (z.B. aus LASR)
- Präsentationsvorlagen

...

54


54


embarc.de




Software-Tools

Für Analyse-Tätigkeiten, vor allem im quantitativen Bereich.




In Software gegossene Checklisten und Fragenkataloge

z.B. aus OWASP, AWS Well Architected Framework ...



Werkzeuge zur Exploration der Implementierung

z.B. zur Generierung von Dokumentationsartefakten



Werkzeuge zur statischen Code-Analyse

z.B. für Abhängigkeiten bzgl. des Umfangs (LOCs), Clone Coverage, Code Smells ...

55

55

28



Software-Tools (2/2)



Auf Review-Aufgaben spezialisierte KI-Agenten

z.B. unterfüttert mit Methodik, Check-Listen etc.



Analyse-Werkzeuge für Ereignisse im laufenden System

z.B. für Logs, Tracing, Monitoring



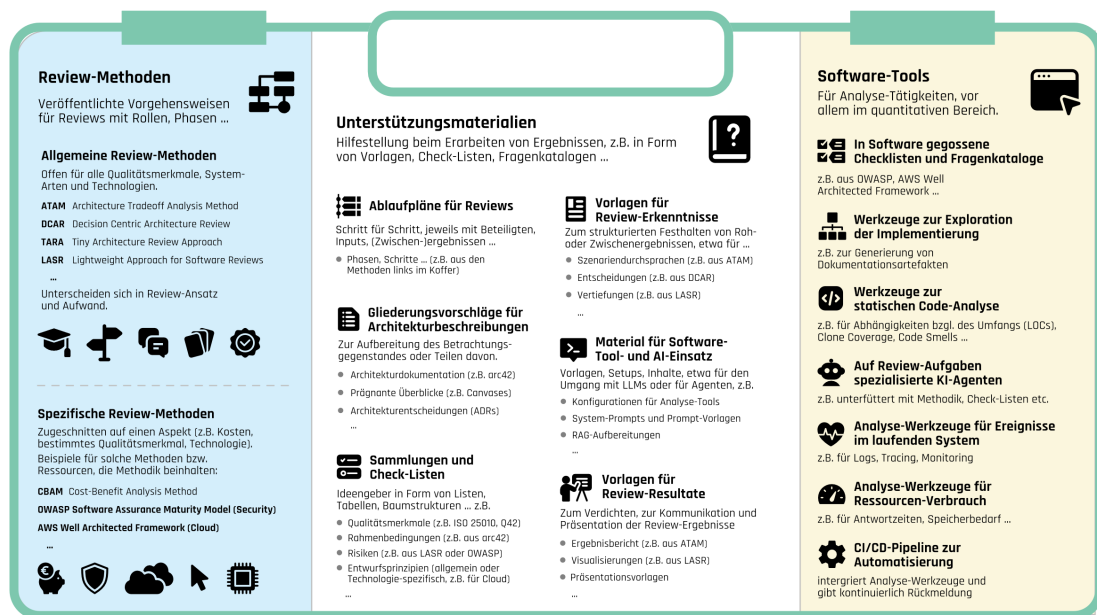
Analyse-Werkzeuge für Ressourcen-Verbrauch

z.B. für Antwortzeiten, Speicherbedarf ...



CI/CD-Pipeline zur Automatisierung

integriert Analyse-Werkzeuge und gibt kontinuierlich Rückmeldung



04.

Welche(s)
Werkzeug(e)
verwende ich?

01. Warum Softwarearchitektur reviewen?
02. Wie funktionieren Architektur-Reviews?
03. Was ist drin im Werkzeugkoffer?
- 04. Wie entscheide ich, welche(s) Werkzeug(e) ich verwende?**
05. Ausblick und weitere Informationen





Fragen zur Methoden- und Werkzeugfindung



Wo kommt der Impuls für das Architektur-Review her?

- Eigener Antrieb im Entwicklungsteam, Management, andere Organisationseinheit, Kunden oder Auftraggeber ...



Was ist die Zielsetzung des Reviews?

- Absichern einer Architekturvision, Aufdecken von Risiken, Erarbeiten von Handlungsfeldern für eine Überarbeitung ...



Was (genau) ist der Betrachtungsgegenstand?

- Eine einzelne Anwendung, ein Teil einer Anwendung, eine Anwendungslandschaft, einzelne Entscheidungen, eine Plattform ...



Wer ist am Ergebnis des Reviews interessiert?

- Wir als Entwicklungsteam, unser Management, ein externer Partner ...



Fragen zur Methoden- und Werkzeugfindung



Was soll der Fokus des Reviews sein? Lassen sich umgekehrt Themen in der Analyse vernachlässigen?

- Einzelne Technologien, Systemteile (z.B. das Frontend), Qualitätsmerkmale (z.B. Sicherheit), ...



Wie reif ist der Betrachtungsgegenstand?

- Konzept oder Idee, prototypisch oder teilweise implementiert, bereits im Markt oder in Produktion ...



Wie viel Wissen um die Qualität des Betrachtungsgegenstandes ist da?

- Keins, einzelnen Risiken oder technologische Schulden benennbar, eklatante Schwächen bereits bekannt ...



Wie nachvollziehbar ist die Architektur des Betrachtungsgegenstandes?

- Historisch gewachsen, von ausgewählten Leuten erklärbar, nachvollziehbar dokumentiert ...



Ein guter Daumenwert zum Start

Meine Erfahrung nach

- Allgemeine, qualitative Methode bieten guten Rahmen für ein Review
- Vertiefungen gezielt einsetzen, wenn aufgedeckte Risiken oder Unsicherheiten das nahelegen
- Eine schlanke Herangehensweise (z.B. LASR) bietet sich für ein frühes, erstes Review ohne große Beteiligung Team-Externer an.
- Bei Risiko-reichen Bewertungen eher auf fundiertere Methoden zurückgreifen.

05.

Ausblick und weitere Informationen

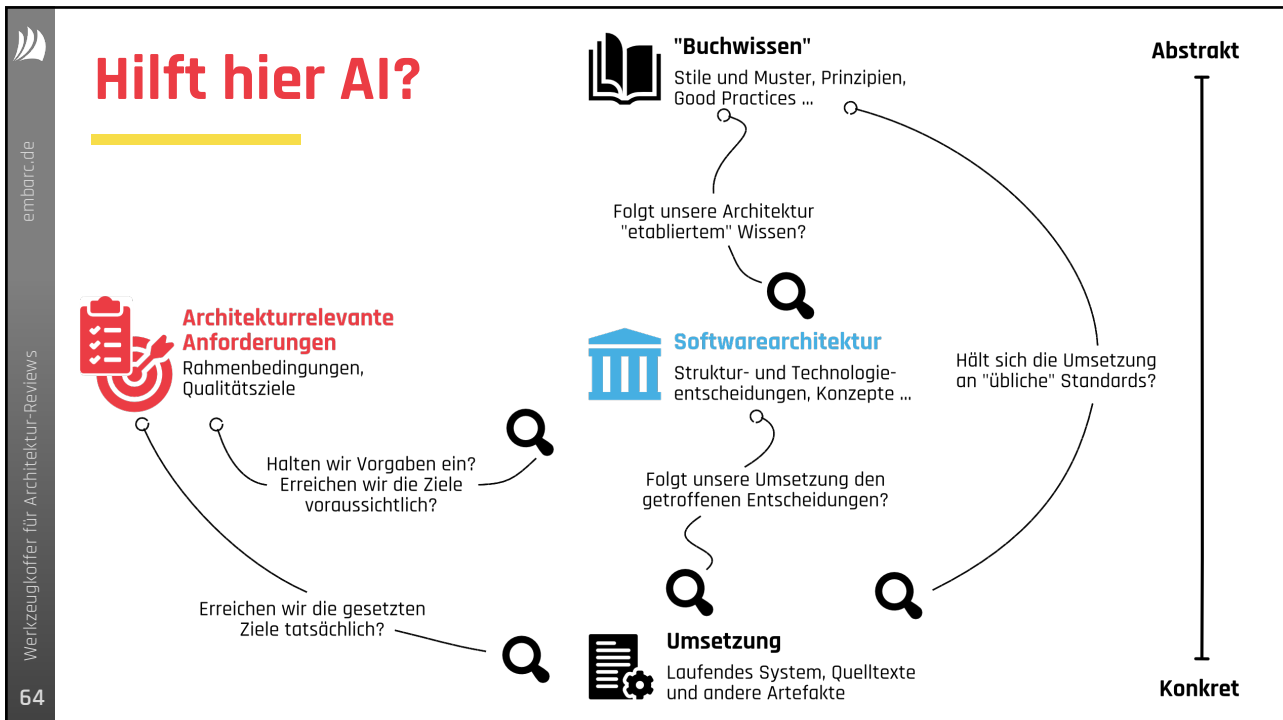
01. Warum Softwarearchitektur reviewen?

02. Wie funktionieren Architektur-Reviews?

03. Was ist drin im Werkzeugkoffer?

04. Wie entscheide ich, welche(s) Werkzeug(e) ich verwende?

05. Ausblick und weitere Informationen



64

Wo kann GenAI unterstützen?

Tätigkeiten, welche die Abgleiche konkretisieren, vor- und nachbereiten. Und für die es auch passendes Handwerkszeug gibt.

- Review-Ziel **fixieren** und **absichern**
- Maßgebliche Stakeholder **identifizieren**
- Review strukturieren, **planen** und **durchführen**
- Architekturrelevante Anforderungen **sichten** und **präzisieren**
- Lösungsstrategie und -ansätze **kommunizieren** (ggf. **identifizieren**)
- Stärken und Schwächen **herausarbeiten**
- Kompromisse **sichtbar machen**
- Risiken **aufdecken** und **bewerten**
- Erkenntnisse **aufbereiten** und **verdichten**
- Ergebnisse **kommunizieren**

65

embarc.de
Werkzeugkoffer für Architektur-Reviews

Review-Methoden

Veröffentlichte Vorgehensweisen für Reviews mit Rollen, Phasen ...

Allgemeine Review-Methoden

Offen für alle Qualitätsmerkmale, System-Arten und Technologien.

- ATAM** Architecture Tradeoff Analysis Method
- DCAR** Decision Centric Architecture Review
- TARA** Tiny Architecture Review Approach
- LASR** Lightweight Approach for Software Reviews

... Unterscheiden sich in Review-Ansatz und Aufwand.

Spezifische Review-Methoden

Zugeschnitten auf einen Aspekt (z.B. Kosten, bestimmtes Qualitätsmerkmal, Technologie). Beispiele für solche Methoden bzw. Ressourcen, die Methodik beinhalten:

- CBAM** Cost-Benefit Analysis Method
- OWASP Software Assurance Maturity Model (Security)**
- AWS Well Architected Framework (Cloud)**

Unterstützungsmaterialien

Hilfestellung beim Erarbeiten von Ergebnissen, z.B. in Form von Vorlagen, Check-Listen, Fragenkatalogen ...

Ablaufpläne für Reviews

Schritt für Schritt, jeweils mit Beteiligten, Inputs, (Zwischen-)ergebnissen ...

- Phasen, Schritte ... (z.B. aus den Methoden links im Koffer)

Vorlagen für Review-Erkenntnisse

Zum strukturierten Festhalten von Roh- oder Zwischenergebnissen, etwa für ...

- Szenariendurchsprachen (z.B. aus ATAM)
- Entscheidungen (z.B. aus DCAR)
- Vertiefungen (z.B. aus LASR)

Gliederungsvorschläge für Architekturbeschreibungen

Zur Aufbereitung des Betrachtungsgegenstandes oder Teilen davon.

- Architekturdokumentation (z.B. arc42)
- Prägnante Überblicke (z.B. Convoyses)
- Architekturentscheidungen (ADRs)

Sammlungen und Check-Listen

Ideengeber in Form von Listen, Tabellen, Baumstrukturen ... z.B.

- Qualitätsmerkmale (z.B. ISO 25010, Q42)
- Rahmenbedingungen (z.B. aus arc42)
- Risiken (z.B. aus LASR oder OWASP)
- Entwurfsprinzipien (allgemein oder Technologie-spezifisch, z.B. für cloud)

Software-Tools

Für Analyse-Tätigkeiten, vor allem im quantitativen Bereich.

- In Software gegessene Checklisten und Fragenkataloge**
z.B. aus DWASP, AWS Well Architected Framework ...
- Werkzeuge zur Exploration der Implementierung**
z.B. zur Generierung von Dokumentationsartefakten
- Werkzeuge zur statischen Code-Analyse**
z.B. für Abhängigkeiten bzgl. des Umfangs (LOCs), Clone Coverage, Code Smells ...
- Auf Review-Aufgaben spezialisierte KI-Agenten**
z.B. unterfüttert mit Methodik, Check-Listen etc.
- Analyse-Werkzeuge für Ereignisse im laufenden System**
z.B. für Logs, Tracing, Monitoring
- Analyse-Werkzeuge für Ressourcen-Verbrauch**
z.B. für Antwortzeiten, Speicherbedarf ...
- CI/CD-Pipeline zur Automatisierung**
Integriert Analyse-Werkzeuge und gibt kontinuierlich Rückmeldung

66

embarc.de
Werkzeugkoffer für Architektur-Reviews

Beispiel: Semantische Anker

ChatGPT 5.1

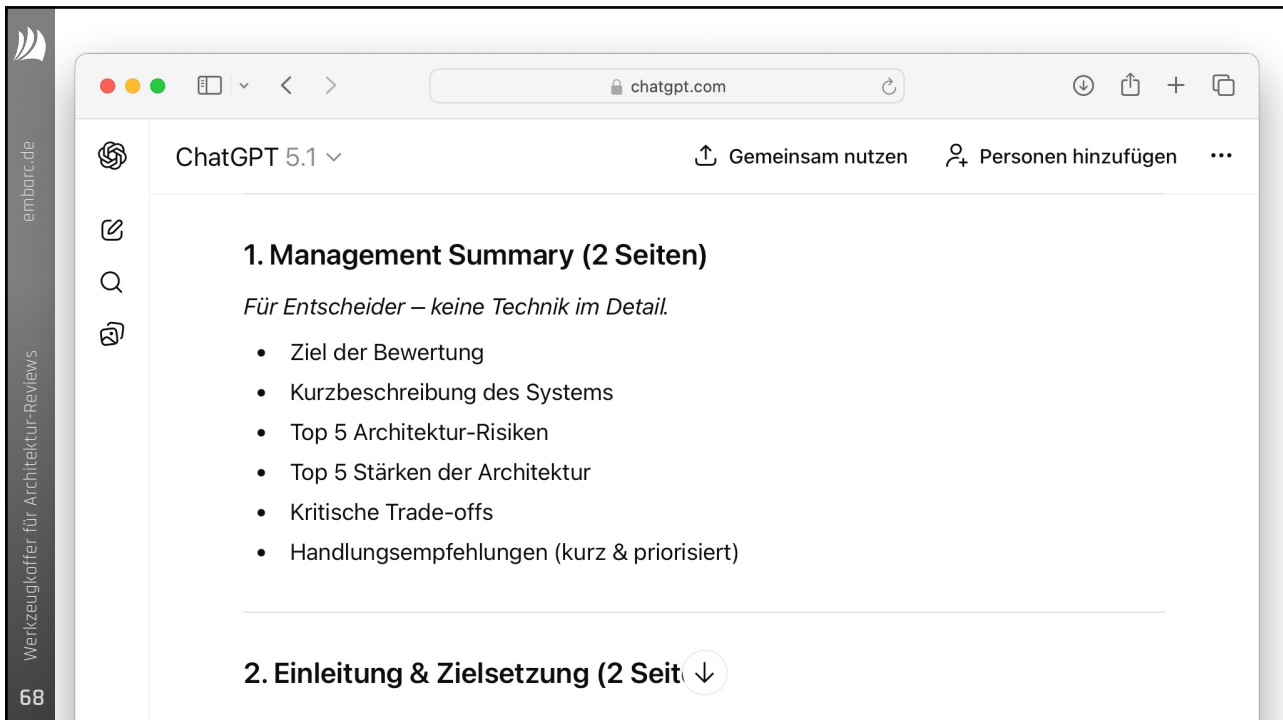
Gemeinsam nutzen
Personen hinzufügen

Ich möchte die Ergebnisse einer qualitativen Architekturbewertung in einem Ergebnisbericht zusammenfassen. Umfang ca. 30 Seiten. Kannst Du mir eine Gliederung vorschlagen, die sich an ATAM orientiert?

Sehr gerne – für einen ca. 30-seitigen Ergebnisbericht einer qualitativen Architekturbewertung nach ATAM eignet sich eine klar strukturierte, nachvollziehbare Gliederung, die sowohl Management-Leser als auch Architekten & Entwickler abholt.

Ich gebe dir zuerst die empfohlene Hauptgliederung mit Seitenrichtwerten, danach eine Detail-Feingliederung je Kapitel.

67



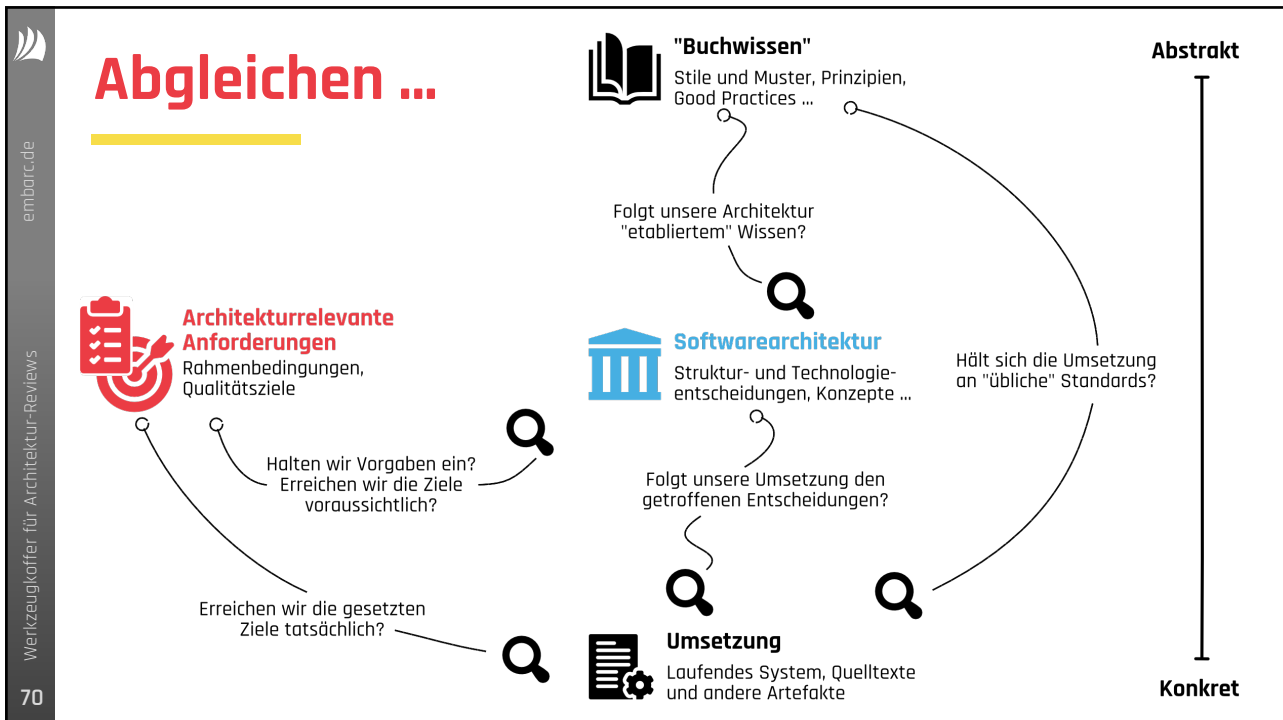
68

Meine Erfahrung mit GenAI und Reviews

Auch in qualitativer Analyse bereits einiges möglich, wenn ...

- tatsächlich die Architektur in Form von Dokumentation explizit vorliegt (z.B. ADRs), und nicht vom Tool ausschnittshaft und mitunter irrtumsanfällig ermittelt werden muss.
- wenn architekturelevante Anforderungen in die Bewertung mit einbezogen werden, und nicht nur „Buchwissen“ und Best Practices.

69



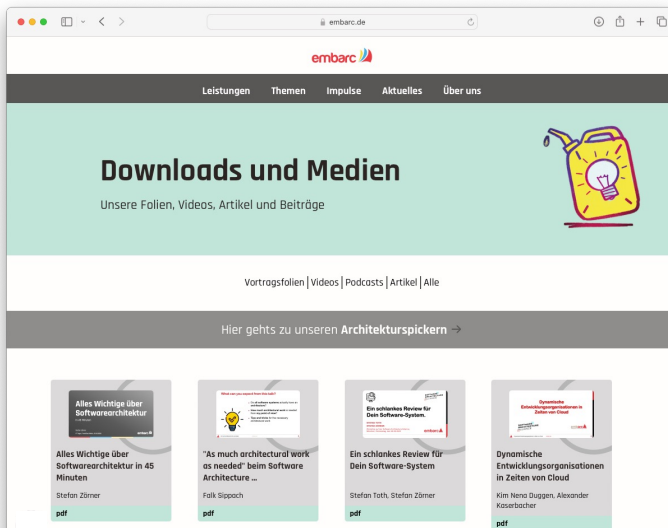
70

TL;DR (Too long; didn't read)

- Für die **Auswahl** geeigneter **Vorgehensweisen** ist vor allem entscheidend,
 - woher der **Impuls** für das Review kommt,
 - was das **Ziel** ist und
 - wie **reif** der Betrachtungsgegenstand bereits ist.
- Schärft euer **Methodenrepertoire** und bleibt offen für neue Möglichkeiten, wie sie perspektivisch mit spezialisierten KI-Agenten bereits an die Tür klopfen.

71

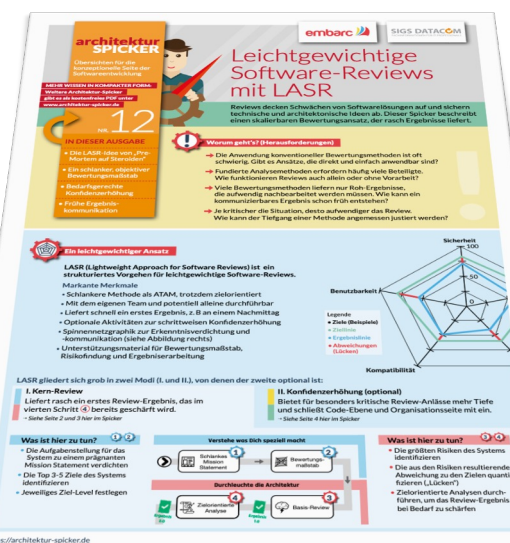
Folien als PDF zum Download



→ embarc.de/download/

72

Architektur-Spicker zum Thema ...



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

Architektur-Spicker #12
Leichtgewichtige Software
Reviews mit LASR



PDF, 4 Seiten
Kostenloser Download.

→ embarc.de/spicker

73

Zum Mitnehmen



Den LASR-Spicker könnt ihr euch direkt vorne bei mir abholen (solange der Vorrat reicht).

Exklusiver Rabatt-Code für die IT-Tage



Software-Systeme reviewen mit dem Lightweight Approach for Software Reviews - LASR

Für alle Teilnehmenden der IT-Tage
2025, reduziert den Preis um 45%

Coupon: ITTAGE2025



➔ leanpub.com/software-systeme-reviewen/c/ITTAGE2025




embarc.de

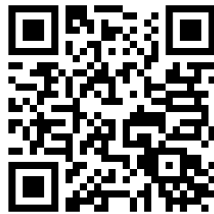
Werkzeugkoffer für Architektur-Reviews

76

Vielen Dank.

Gerne Vernetzen auf LinkedIn ...

 linkedin.com/in/stefan-zoerner



76



embarc.de

Werkzeugkoffer für Architektur-Reviews

77

Exklusiver Rabatt-Code für die IT-Tage



Software-Systeme reviewen

mit dem Lightweight Approach
for Software Reviews - LASR

Für alle Teilnehmenden der IT-Tage
2025, reduziert den Preis um 45%

Coupon: ITTAGE2025



➔ leanpub.com/software-systeme-reviewen/c/ITTAGE2025

77