

Training Bite: Architekturbewertung

Stefan Zörner, embarc

INNOQ Technology Day
Online, 05. Dezember 2024



Abstract

Training Bite: Architekturbewertung

Architekturbewertung ist ein zentrales Werkzeug, um Sicherheit im Architekturentwurf zu gewinnen und um bestehende Lösungen oder Lösungsansätze fokussiert und nachhaltig weiterentwickeln zu können. Oder auch um sie informiert zu verwerfen. Vor allem fördert Architekturbewertung Kommunikation und Transparenz, richtig angewendet passt sie daher auch in agile Vorgehen.

In diesem Training Bite gebe ich einen Einblick in das iSAQB CPSS-A Modul "Architekturbewertung" (ARCEVAL). Zunächst diskutieren wir die unterschiedlichen Herangehensweisen an Architekturbewertung (inkl. konkreter Methoden dazu wie ATAM oder DCAR). Ihr erfahrt, wann was passt und welche Vor- und Nachteile die Ansätze jeweils haben.

Mit LASR (Lightweight Approach for Software Reviews) lernt Ihr im Anschluss eine schlanke Methode etwas genauer kennen. LASR greift auf die Essenz etablierter Bewertungsansätze zurück und erlaubt es Eurem Team schnell zu wertvollen Erkenntnissen zu gelangen.

Nach diesem Training Bite könnt Ihr im kleinen Team oder notfalls sogar allein ein leichtgewichtiges Review Eurer Software-Lösung selber durchzuführen. Ich gebe diesbezüglich Orientierung und stelle Unterstützungsmaterial bereit.



embarc.de

Architekturbewertung

1



Stefan Zörner

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

Schwerpunkte

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



Warum Architekturbewertung?

Training Bite: Architekturbewertung



Was ist Softwarearchitektur?

Softwarearchitektur :=

Σ wichtige Entscheidungen

wichtig =

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



Themen für Entscheidungen

Zerlegung

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



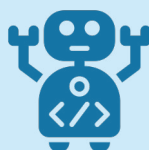
Zielumgebung

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, ...





Was ist ein Review?

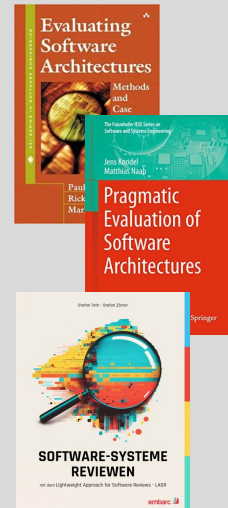


Wörtlich:
Review == etwas **(über)prüfen**, besprechen, ...

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.



Eine Neuentwicklung steht an und erste Lösungsansätze stehen im Raum.



Typische Anlässe (... findet Ihr Euch wieder?)



Anlass 1 („Neuanfang“)

Eine **Neuentwicklung** steht an und **erste Lösungsansätze** stehen im Raum.

Leitfrage:

Seid Ihr und Euer Team auf dem **richtigen Weg**?

Unterschiedliche Stakeholder verfolgen widersprüchliche Ziele mit Eurer Software.





Typische Anlässe (... findet Ihr Euch wieder?)



Anlass 2 („Wünsche“)

Unterschiedliche Stakeholder verfolgen **widersprüchliche Ziele** mit Eurer Software.

Leitfrage:

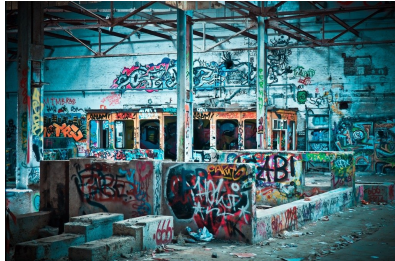
Wie **konkretisiert** und **priorisiert** Ihr deren Wünsche?

Das Management hat das Vertrauen in
Eure Lösung verloren.





Typische Anlässe (... findet Ihr Euch wieder?)



Anlass 3 („Unruhe“)

Das **Management** hat das **Vertrauen** in Eure Lösung verloren.

Leitfrage:

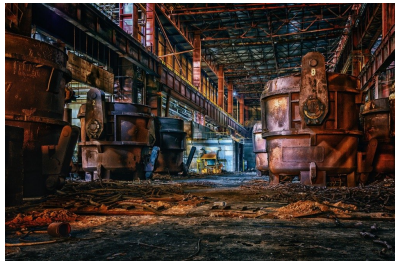
Wie gewinnt Ihr es zurück und strahlt **Sicherheit** aus?

Größere Umbaumaßnahmen in Eurer Software stehen an.





Typische Anlässe (... findet Ihr Euch wieder?)



Anlass 4 („Legacy“)

Größere **Umbaumaßnahmen** in Eurer Software stehen an.

Leitfrage:

Wie wählt Ihr passende Lösungsansätze **nachvollziehbar** aus?

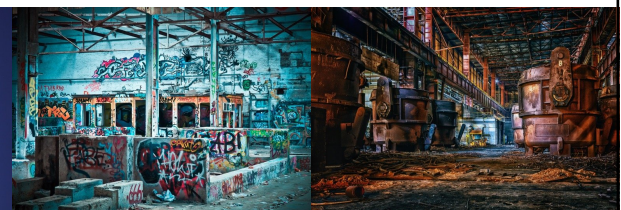
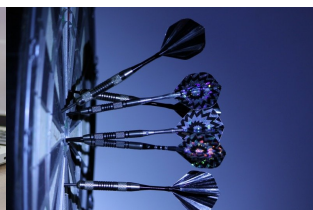


Das Leistungsversprechen von Reviews

In all den gezeigten Situationen unterstützen Review-Ansätze und helfen Euch und Eurem Team die Leitfragen zu beantworten.

Konkret: Architekturbewertungen ...

- ... **decken** Kompromisse und **Risiken** von Softwarelösungen **auf**.
- ... **sichern** technische und architektonische **Ideen ab**.



Bewertungs- methoden

Training Bite: Architekturbewertung



embarc.de

Architekturbewertung

17

Kernelemente eines Reviews

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?



Kernelemente eines Reviews

Für eine Bewertung oder ein Review braucht es mindestens



Einen Anlass

Warum bewerten wir?



Einen Gegenstand

Was bewerten wir?

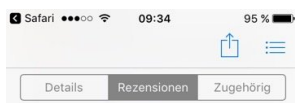


Einen Maßstab

Wonach (wo gegen) bewerten wir?



Nörgeln ist einfach ...



4. Sehr gut

★★★★★ Browniiiiii - 08.05.
Funktioniert prima. Es wäre noch super, wenn man den Ton bei den exportierten Videos deaktivieren könnte.

5. Keine gifs

★★★★★ guttiis - 13.09.
App selbst funktioniert super, aber gespeicherte sind nur als Fotos abrufbar

6. Geht besser !

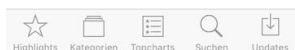
★★★★★ Mrs.happysmile - 18.09.
Ich kann leider keine Live Bilder von der Front K bearbeiten ...
großer minus Punkt !
Bitte um update (iPhone SE)

7. Schrott !!!

★★★★★ Hashtag 2 - 06.10.
Kack App hier geht gar nichts !!!

7. Schrott !!!

★★★★★ Hashtag 2 - 06.10.
Kack App hier geht gar nichts !!!





Bewertungsmethoden (Auswahl)

SAAM Software Architecture Analysis Method	ARID Architecture Review for Intermediate Designs
ATAM Architecture Tradeoff Analysis Method	DCAR Decision Centric Architecture Review
CBAM Cost-Benefit Analysis Method	DASE Decision and Scenario based architecture evaluation
TARA Tiny Architecture Review Approach	Pre-Mortem Risk-Brainstorming and Mitigation
PBAR Pattern Based Architecture Review	LASR Lightweight Approach for Software Reviews

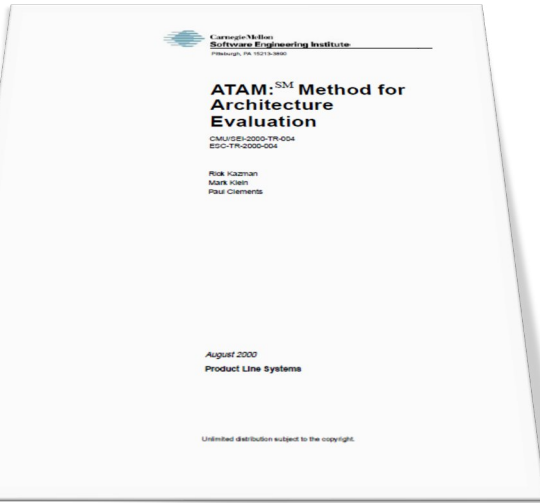


Bewertungsmethoden (Auswahl)

SAAM Software Architecture Analysis Method	ARID Architecture Review for Intermediate Designs
ATAM Architecture Tradeoff Analysis Method	DCAR Decision Centric Architecture Review
CBAM Cost-Benefit Analysis Method	DASE Decision and Scenario based architecture evaluation
TARA Tiny Architecture Review Approach	Pre-Mortem Risk-Brainstorming and Mitigation
PBAR Pattern Based Architecture Review	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
- Früh anwendbar, geht von den **Zielen** aus



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)

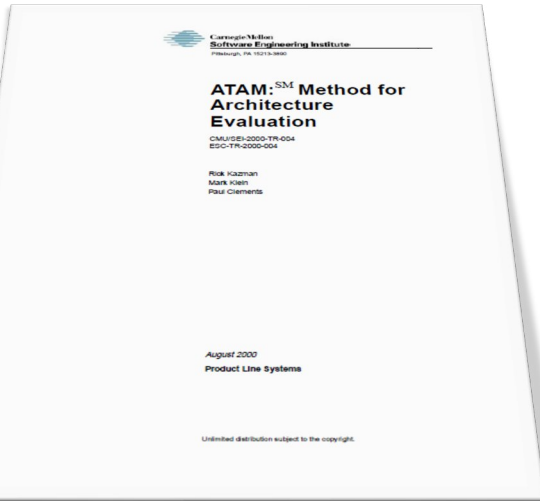
Ist die Software leicht auf andere Situationen oder Zielumgebungen (z.B. anderes OS) übertragbar? ...

Betriebssicherheit (Safety)

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



ATAM



Architecture Tradeoff Analysis Method

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



ATAM, Kernphasen: Evaluierung

Phase 1: Bewertung – Architektur-zentriert

- Schritt 1: ATAM präsentieren
- Schritt 2: Geschäftsziele präsentieren
- Schritt 3: Architektur präsentieren
- Schritt 4: Architekturansätze identifizieren
- Schritt 5: Utility-Tree erstellen
- Schritt 6: Architekturansätze analysieren

Phase 2: Bewertung – Stakeholder-zentriert

- Schritt 7: Szenario-Brainstorming
- Schritt 8: Architekturansätze analysieren
- Schritt 9: Ergebnisse präsentieren



Was ist ein Szenario?

Ein (Qualitäts-)Szenario ...

- ... ist ein kurzer Text (1-3 Sätze).
- ... beschreibt **beispielhaft** die Verwendung des Systems, und zwar so dass ein **Qualitätsmerkmal** die Hauptrolle spielt.

Wie konkret?



Man muss sinnvoll drüber reden können.

Man muss es (theoretisch) überprüfen können.

(Kein Abnahmekriterium, kein Testfall!)



Beispiel Verwendungsszenarien

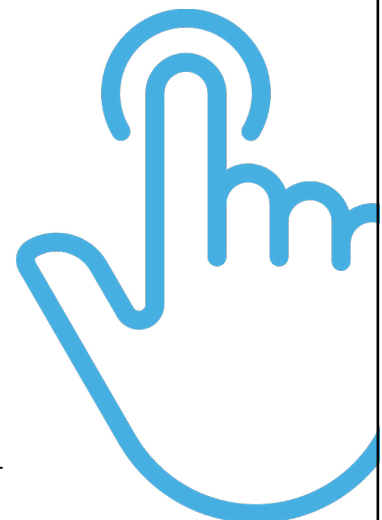
(oder auch: Use Case-Szenarien)

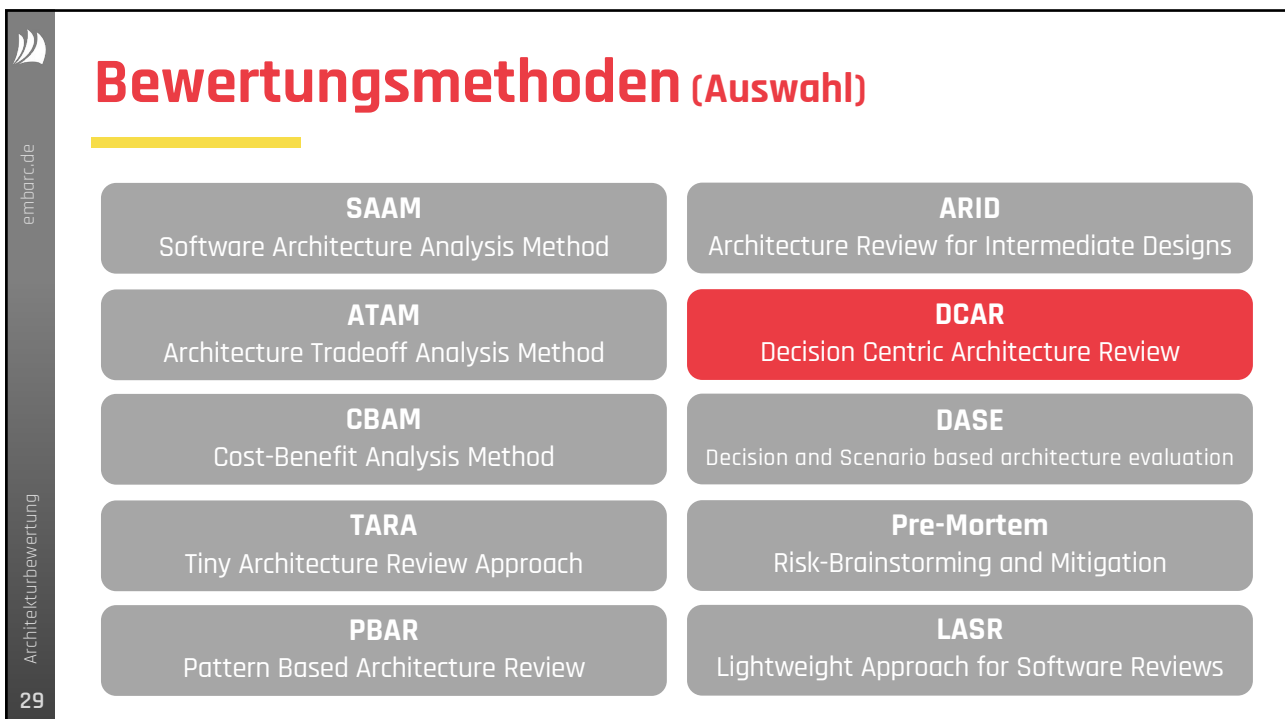
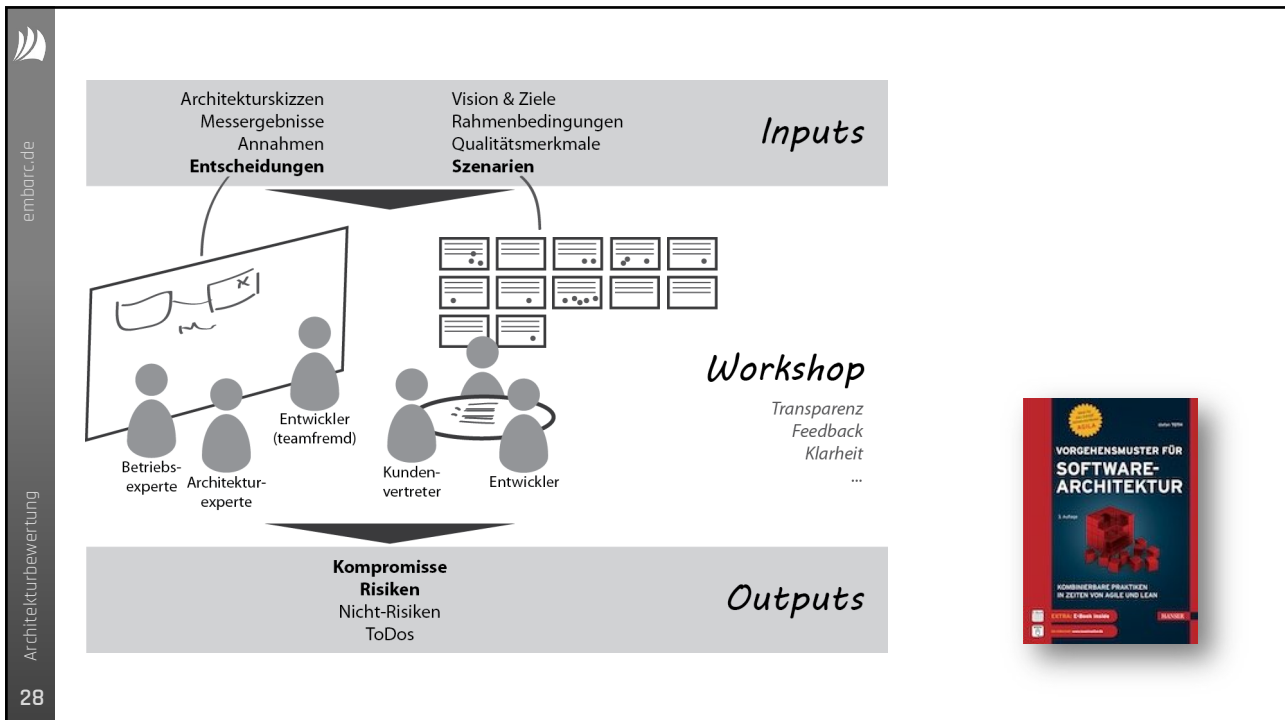
- „Normale“ Verwendung des Systems durch einen Akteur
- Beispiel-Quelle: Ein Benutzer im Systemkontext

Beispiel-Szenario für eine Online-Schachplattform



Eine Spielerin setzt ihren Gegner bei einer Online-Partie matt. Beide werden darüber maximal 2 Sekunden später informiert.







DCAR

Decision-Centric Architecture Reviews

Method Description

University of Groningen
Tampere University of Technology

Version: 2/11/2011

Authors: Paris Avgeriou, Veli-Pekka Eloranta, Neil Harrison, Uwe van Hoesch, Kai Koskimies

Motivation

Software architecture is the result of a set of fundamental design decisions made by a system's architects and other stakeholders. These decisions are in different levels of maturity and some introduce uncertainty. Decisions that turn out to be sub-optimal, or even risky are likely to be subject to change in later project stages. However, changes to the software architecture become more expensive the later they are made. Therefore, software architecture should be evaluated in early stages of the system development in order to avoid extraneous effort and cost.

Unfortunately, current (scenario-based) architecture evaluation methods are not broadly established in industrial practice. The critique from industrial practitioners focuses mostly on the excessive effort the methods take, as well as their level of difficulty. In some cases, information about these methods is so overwhelming that it prevents practitioners from following them. Instead, many companies perform architecture evaluation ad hoc, inadequately, or not at all.

The main motivation of the Decision Based Architecture Review method (DCAR), presented in this document, is to alleviate some of the known problems of scenario-based evaluation methods. In particular, these methods suffer from two major problems. First, they are rather heavyweight. This is partly due to the fact that scenario-based evaluation has a rather broad scope: it uses considerable time for discussing and refining business goals and (quality) requirements in addition to the analysis of the actual architecture. Second, since the analysis is structured according to the scenarios, which are basically test cases for the architecture, only a very limited number of scenarios can be analyzed in practice; thus the coverage of scenario-based evaluation remains an open question. This means that the trustworthiness of scenario-based evaluation cannot be guaranteed.

The idea of DCAR is to organize the evaluation according to the architectural decisions, rather than using scenarios. The main expected advantages are that the evaluation process can be made in shorter time and less company work hours, and that the coverage of the evaluation can be directly expressed in terms

¹ In alphabetical order

Decision-Centric Architecture Review

- Ursprung: Universitäten von Groningen (NL), Tampere (FI), 2011
- Initial ähnlicher Ablauf zu ATAM, produziert auch ähnliche Ergebnisse
- Geht in der Analyse von den (Architektur-)Entscheidungen aus



Herausforderungen ...

Die Anwendung fundierter Bewertungsmethoden ist mitunter schwierig.

- Der Einsatz erfordert häufig **viele Beteiligte**.
- Es sind oftmals **Vorarbeiten nötig**, beispielsweise die Aufbereitung der Geschäftsziele und das Anfertigen eines Architekturüberblicks.
- Sie liefern **nur Rohergebnisse**, die für eine effiziente Kommunikation aufwendig nachzubearbeiten sind.
- Durchführung und Moderation verlangen einiges ab - die Methoden **unterstützen** dabei **wenig**.



Leichtgewichtige Reviews

Training Bite: Architekturbewertung



embarc.de

Architekturbewertung

33

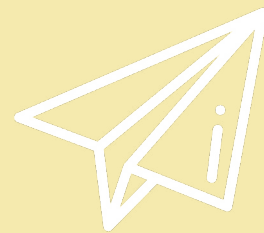
Was ist leichtgewichtig?

Merkmale eines leichtgewichtigen Reviews

- Die **Anzahl** der Beteiligten / **Stakeholder** ist **gering**.
- **Aufwand** und Dauer sind **überschaubar**.
- **Ergebnisse** liegen vergleichsweise **schnell** vor.

Konkret z.B.

- Entwicklungsteam führt Review **allein** und **ohne Vorbereitung** durch.
- Bereits nach einem halben Tag liegt ein **kommunizierbares Ergebnis** vor.





Erfahrungswissen



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



Autoren: Stefan Toth, Stefan Zörner
Verlag: Leanpub, September 2023
Sprache: Deutsch, EPUB, PDF

→ leanpub.com/software-systeme-reviewen/



Markante Merkmale von LASR

- Schlankere Methode als ATAM, trotzdem **zielorientiert**
- Mit dem **eigenen Team** und potentiell **alleine durchführbar**
- Liefert **schnell** ein **erstes Ergebnis**, z.B. an einem Nachmittag
- **Spinnennetzgraphik** zur Erkenntnisverdichtung und -kommunikation
- optionale Aktivitäten zur schrittweisen **Konfidenzerhöhung bei Bedarf**
- **Unterstützungsmaterial** für Bewertungsmaßstab, Risikofindung und Ergebniserarbeitung



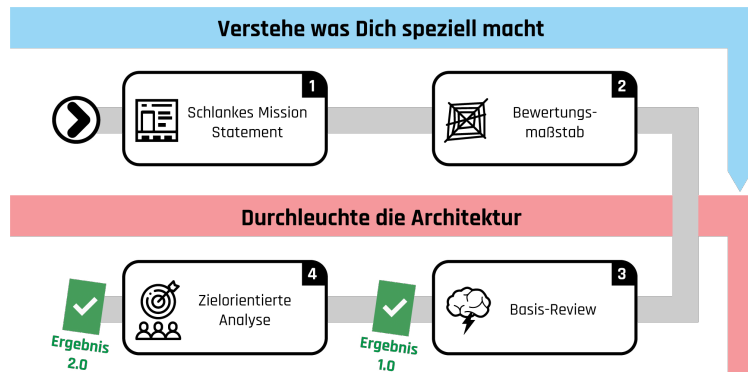


Ein schlanker Ansatz

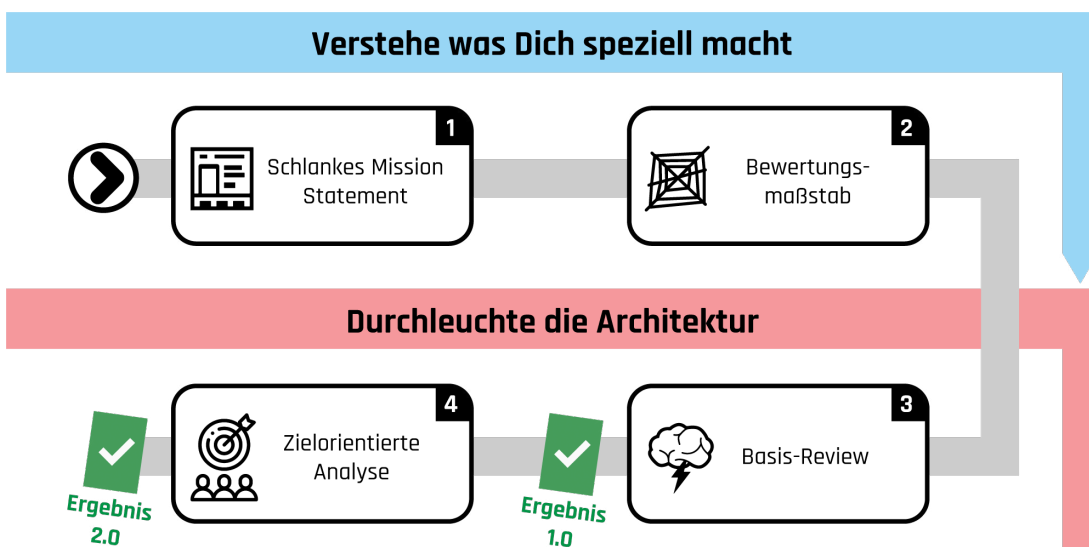
LASR (Lightweight Approach for Software-Reviews) ist ein strukturiertes Vorgehen für leichtgewichtige Software-Reviews.

LASR Kern-Review →

- 2 Tätigkeiten
- 4 Schritte
- Ein frühes erstes Ergebnis



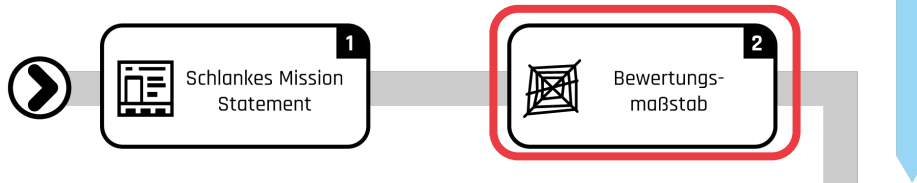
Tätigkeiten und Schritte im Kern-Review





Verstehe, was Dich speziell macht

Verstehe was Dich speziell macht



Was ist zu tun?

- Die Aufgabenstellung für das System zu einem prägnanten Mission Statement verdichten
- **Die Top 3-5 Ziele des Systems identifizieren**
- Jeweiliges Ziel-Level festlegen





Top-5-Challenger (Unterstützungsmaterial)

Material:

- 14 Karten mit Qualitätsmerkmalen



Vorbereitung:

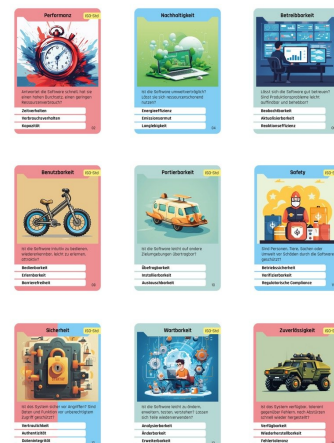
- Mischt die Karten in einem verdeckten Stapel.
- Deckt 5 Karten zufällig auf und legt sie nebeneinander. (das ist Eure Start Top-5)
- Legt die übrigen 9 in einer 3 x 3 Matrix für alle gut sichtbar offen aus.



Top-5-Challenger, Vorbereitung (Beispiel)



Start Top-5



Kandidaten



embarc.de

Architekturbewertung

42

Ablauf einer Runde



Einen „Challenger“ nominieren

- Team-Mitglied schlägt Ziel vor, das in Top-5 fehlt.
- Die Gruppe diskutiert darüber.
- Der Challenger verdrängt eine Karte oder landet selbst auf dem Ablagestapel.



embarc.de

Architekturbewertung

43

Nach 2 Runden, Beispiel (Beispiel)



Aktuelle Top-5



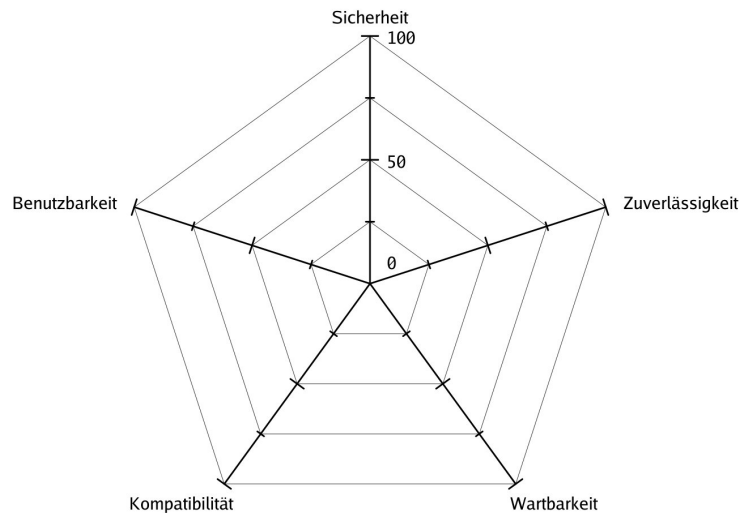
Ablagestapel



Kandidaten

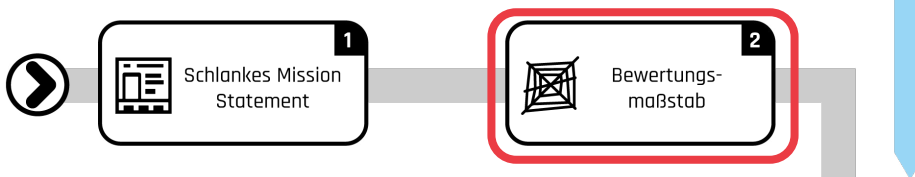


Ergebnis nach diesem Schritt (Beispiel)



Verstehe, was Dich speziell macht

Verstehe was Dich speziell macht



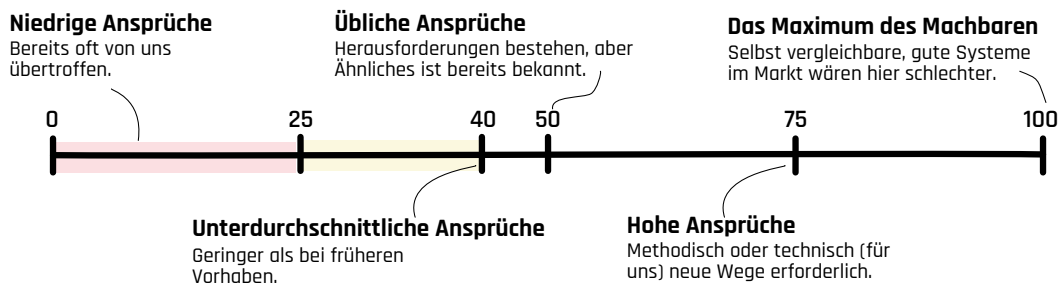
Was ist zu tun?

- Die Aufgabenstellung für das System zu einem prägnanten Mission Statement verdichten
- Die Top 3-5 Ziele des Systems identifizieren
- **Jeweiliges Ziel-Level festlegen**



Zielwerte bestimmen

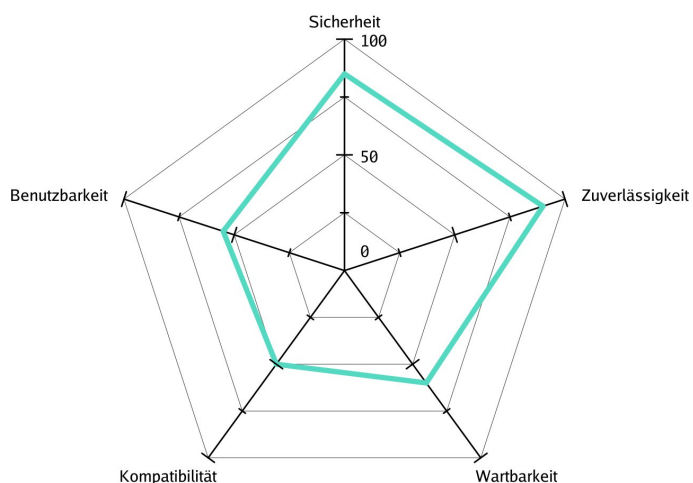
Der Zielwert zwischen 0 und 100 beschreibt jeweils, wie hoch die Erwartungen in dem Bereich sind. Im Vergleich zu anderen Zielen der Top-3-5 und anderen Vorhaben aus dem Kontext des Unternehmens / der Organisation.



Hinweis: Es handelt sich hier um eine Einschätzung, nichts Messbares.



Bewertungsmaßstab einzeichnen



LASR-Ergebnisdiagramm

Die Top-3-5 Qualitätsziele bilden die Achsen des Diagramms.

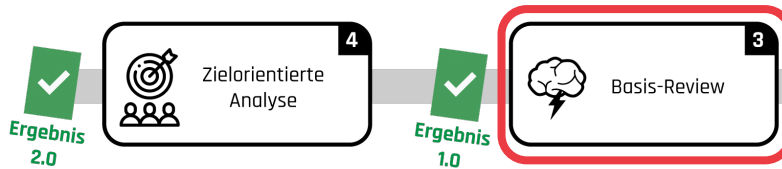
Die Zielwerte spannen darauf eine grüne Linie auf.

← Beispiel



Durchleuchte die Architektur

Durchleuchte die Architektur



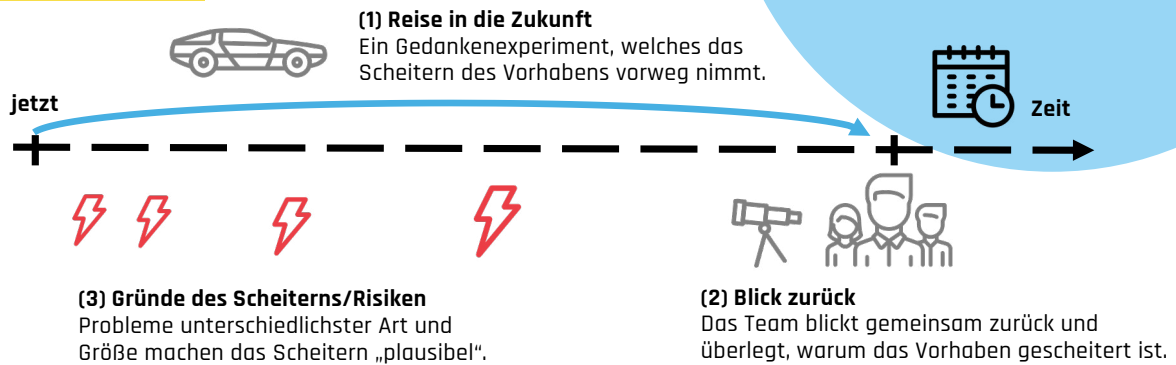
Was ist zu tun?

- Die größten Risiken des Systems identifizieren
- Die aus den Risiken resultierende Abweichung zu den Zielen quantifizieren („Lücken“)
- Zielorientierte Analysen durchführen, um das Review-Ergebnis bei Bedarf zu schärfen

Grundidee: Pre Mortem



Was ist ein Pre-Mortem?



- Pre-Mortems werfen einen kritischen Blick auf Vorhaben. Und zwar aus einer hypothetischen Zukunft, in der das Vorhaben gescheitert ist.
- Aus dieser Perspektive ergeben sich „Gründe des Scheiterns“, die aus heutiger Sicht mögliche Risiken darstellen.





Die 8 Risiko-Kategorien von LASR



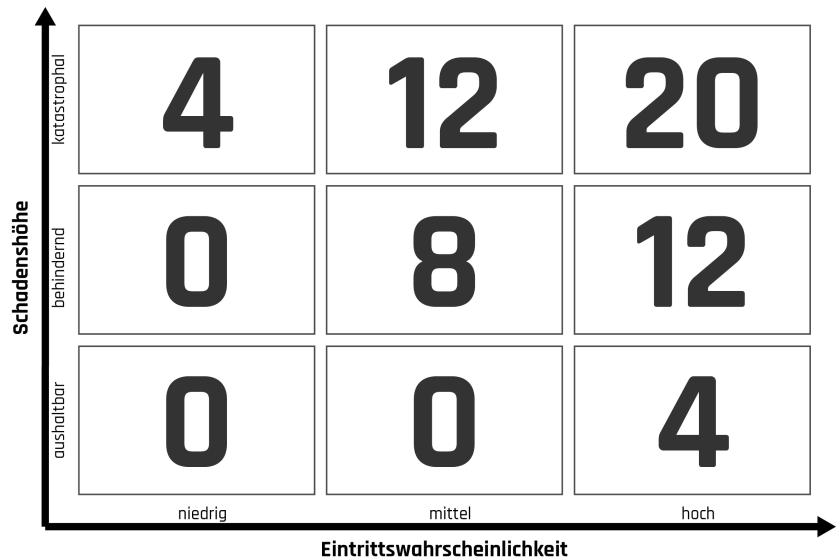
Typische Risikothemen als Ideengeber



Brainstorming-Unterstützung: Das LASR-Kartenset hält insgesamt 32 konkrete Risikokarten als Ideengeber parat, 4 in jeder der 8 Kategorien.



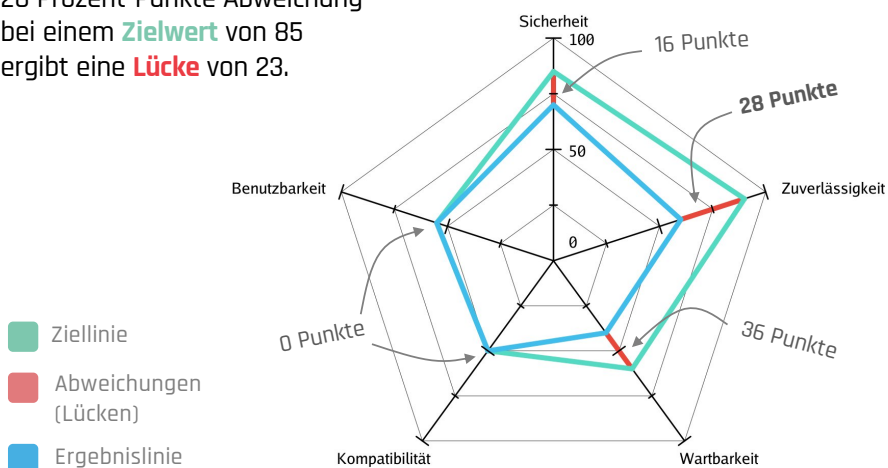
Risiken einordnen



Abweichungen einzeichnen

Beispiel

28 Prozent-Punkte Abweichung bei einem Zielwert von 85 ergibt eine Lücke von 23.





LASR an einem Vormittag

Review-Workshop System YX -- Agenda

Die Zeiten dienen Euch zur groben Orientierung.
Wir machen aber pünktlich Schluss.



09:00 - 10:00

Begrüßung,
Einstieg LASR

ca. 10 Minuten

1
Abgrenzung,
Mission Statement
("Claims")

ca. 15-20 Minuten

2
Bewertungsmaßstab
festlegen

ca. 20-30 Minuten

kurze Pause
(+ Basis-Review
vorbereiten)

10:15 - 11:30

3
Basis-Review
("Pre-Mortem")

ca. 60 Minuten

3
Zielerreichung
bestimmen

ca. 15 Minuten

kurze Pause

11:45 - 12:15

3
Ergebnisse
diskutieren

ca. 10 Minuten

3
Ziel-orientierte
Analyse planen,
TODOs

ca. 10-20 Minuten

Ende des Workshops

(c) embarc GmbH

← Beispiel-Agenda

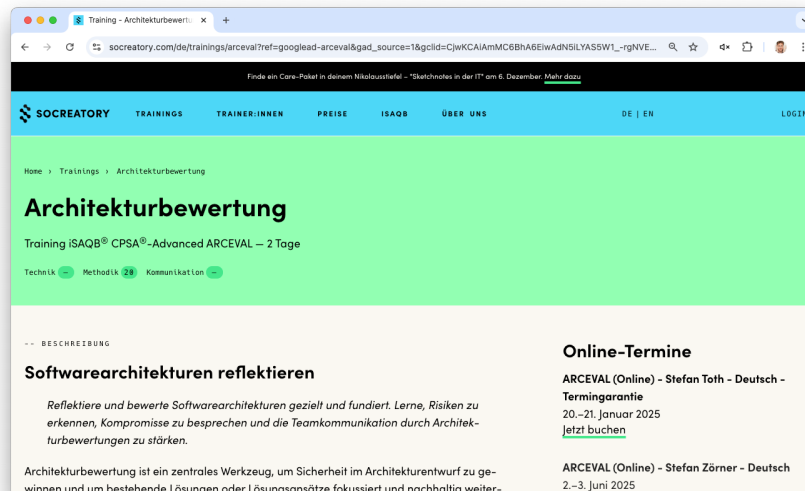
Quelle: S. Toth, S. Zörner:
„Leichtgewichtige Software-
Reviews mit LASR“, Informatik
Aktuell 2024

Weitere Informationen

Training Bite: Architekturbewertung

Seminar zu Architekturbewertung

→ socreatory.com/de/trainings/arceval



Artikel zum Einstieg in LASR (online)



Stefan Toth & Stefan Zörner:
„Leichtgewichtige Software-
Reviews mit LASR“
Online, Informatik Aktuell

<https://www.informatik-aktuell.de/entwicklung/methode/n/leichtgewichtige-software-reviews-mit-lasr.html>

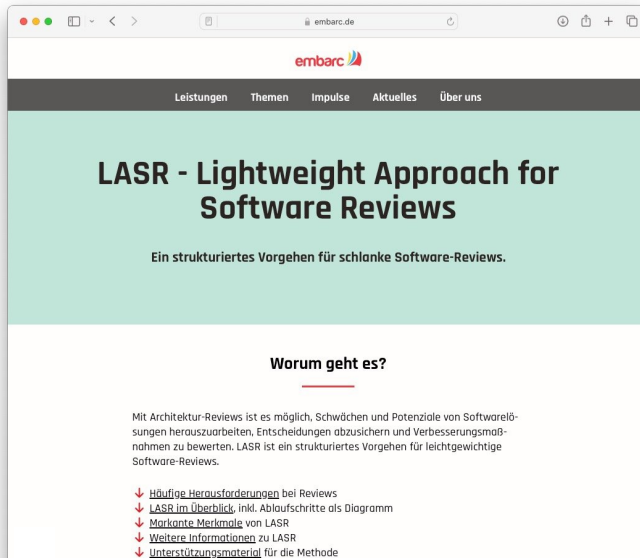


embarc.de

Architekturbewertung

60

LASR-Themenseite bei embarc



→ embarc.de/lasr-reviews/



Unsere Speaker 2024



Alex Kaserbacher



Stefan Toth



Matthias Naab



Stefan Zörner



Anja Kammer



Michael Wyss



Kim Duggen



Felix Kammerlander

16. Dezember | online

Infos, Programm & Anmeldung
embarc.de/events/punsch-2024





Vielen Dank.

Ich freue mich auf Eure Fragen!



Stefan.Zoerner@embarc.de



linkedin.com/in/stefan-zoerner



@StefanZoerner@mastodon.social

