

Leichtgewichtige Software-Reviews 2.0

Stefan Toth

Nürnberg, 22. Jänner 2024





Stefan Toth



Stefan.Toth@embarc.de



@st_toth



linkedin.com/in/sto-embarc



www.embarc.de



Buch zum Thema



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

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



➔ leanpub.com/software-systeme-reviewen/

00.

Agenda

Was euch erwartet

"Leichtgewichtig"?

01

Warum Reviews?

02

LASR Schritte 1-2

03

LASR Schritte 3-4

04

Konfidenzerhöhung

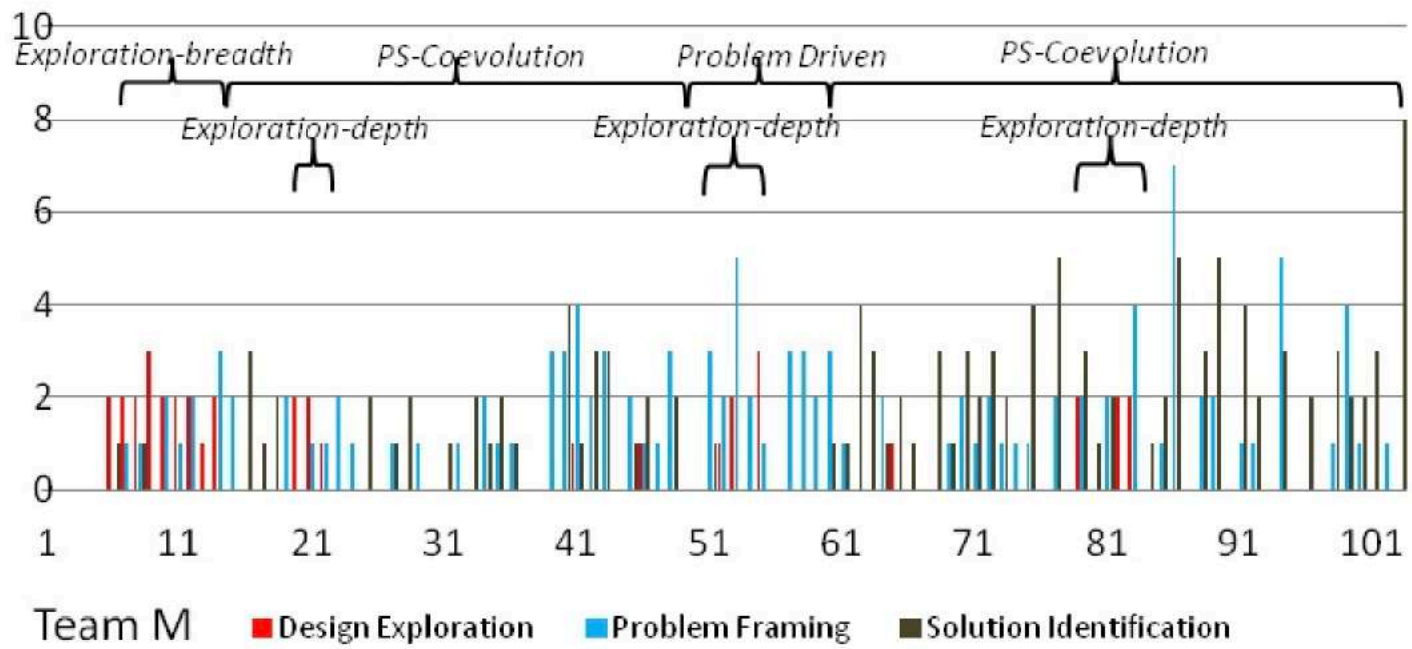
05

01.

Warum Reviews?

Kostet etwas und schafft keinen direkten Wert...

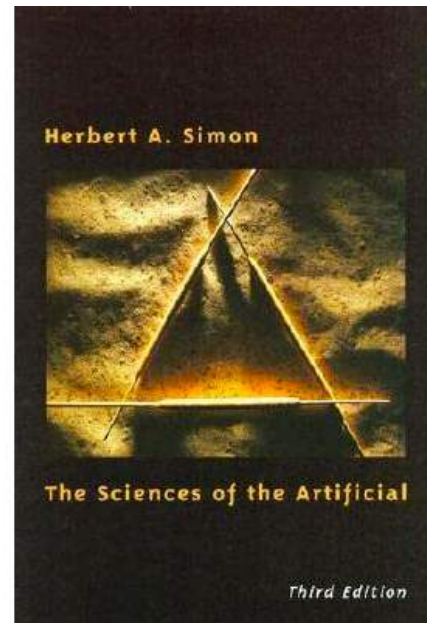
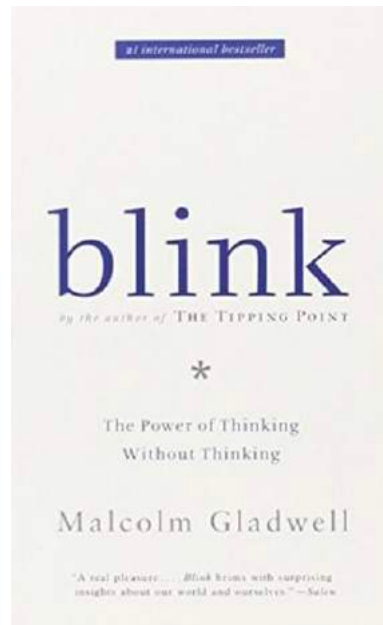
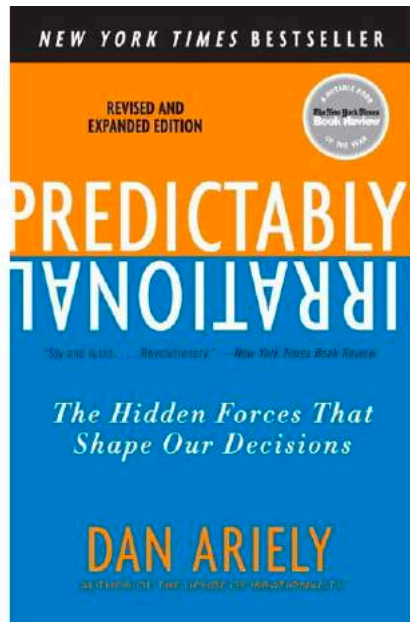
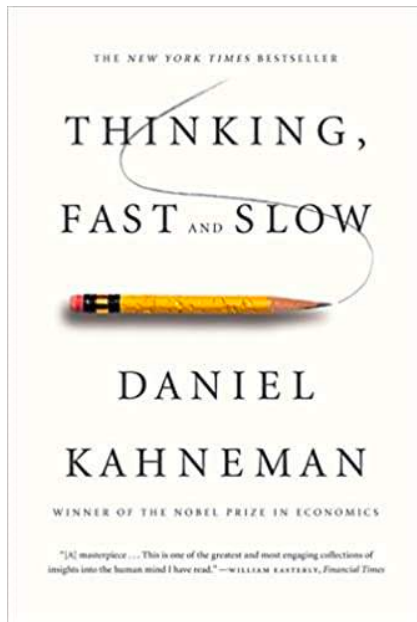
Wie wir Systeme designen



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

erfahrungsbasiert

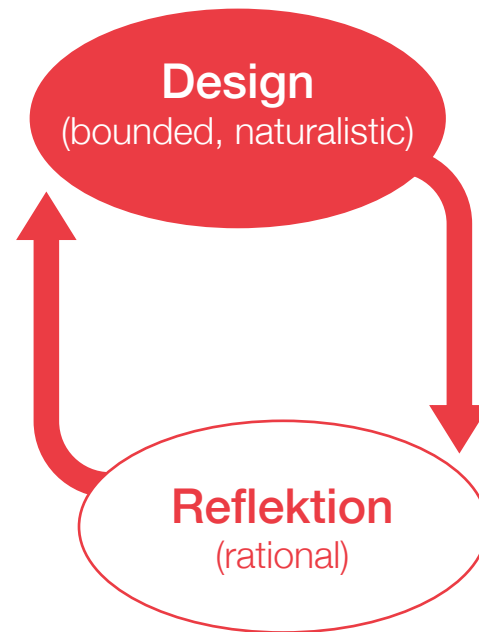
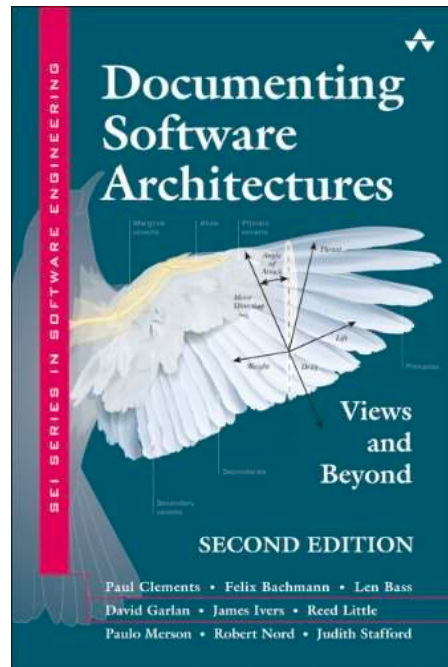
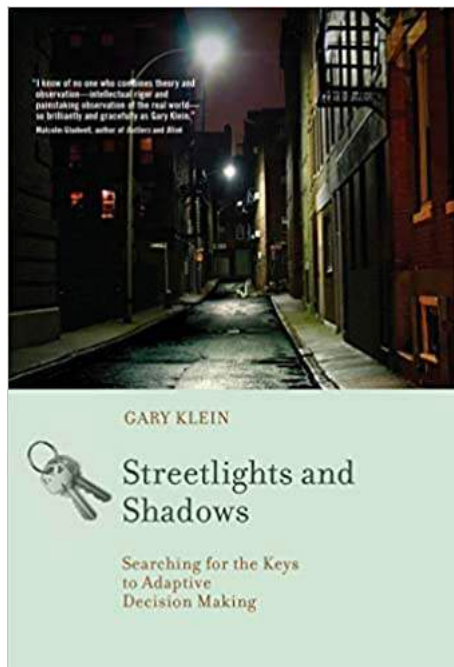
Wie Entscheidungen funktionieren...



**Wir agieren nicht
isoliert...**



Rationalität im nachhinein hilft



02.

„Leichtgewichtig“

Was macht Reveiws
leichtgewichtig?

Bewertungsmethoden

SAAM

Software Architecture Analysis Method

ATAM

Architecture Tradeoff Analysis Method

CBAM

Cost-Benefit Analysis Method

TARA

Tiny Architecture Review Approach

PBAR

Pattern Based Architecture Review

ARID

Architecture Review for Intermediate Designs

DCAR

Decision Centric Architecture Review

DASE

Decision and Scenario based arch. evaluation

Pre-Mortem

Risiko-Brainstorming and Mitigation

LASR

Lightweight Approach for Software Reviews

„High ceremony“



Was heißt „leichtgewichtig“?



wenig (notwendige)
~~Viele~~ **Beteiligte**



Schnell in die
~~Voll in die Breite, voll in die Tiefe, Ergebnis am Schluss:~~
fokussiert *früh und iterativ verbessert*



Zeiteinsatz mind. 2 ~~Tage~~ *Stunden*

Bewertungsmethoden

SAAM

Software Architecture Analysis Method

ATAM

Architecture Tradeoff Analysis Method

CBAM

Cost-Benefit Analysis Method

TARA

Tiny Architecture Review Approach

PBAR

Pattern Based Architecture Review

ARID

Architecture Review for Intermediate Designs

DCAR

Decision Centric Architecture Review

DASE

Decision and Scenario based arch. evaluation

Pre-Mortem

Risiko-Brainstorming and Mitigation

LASR

Lightweight Approach for Software Reviews

Bewertungsmethoden

SAAM

Software Architecture Analysis Method

ATAM

Architecture Tradeoff Analysis Method

CBAM

Cost-Benefit Analysis Method

TARA

Tiny Architecture Review Approach

PBAR

Pattern Based Architecture Review

ARID

Architecture Review for Intermediate Designs

DCAR

Decision Centric Architecture Review

DASE

Decision and Scenario based arch. evaluation

Pre-Mortem

Risiko-Brainstorming and Mitigation

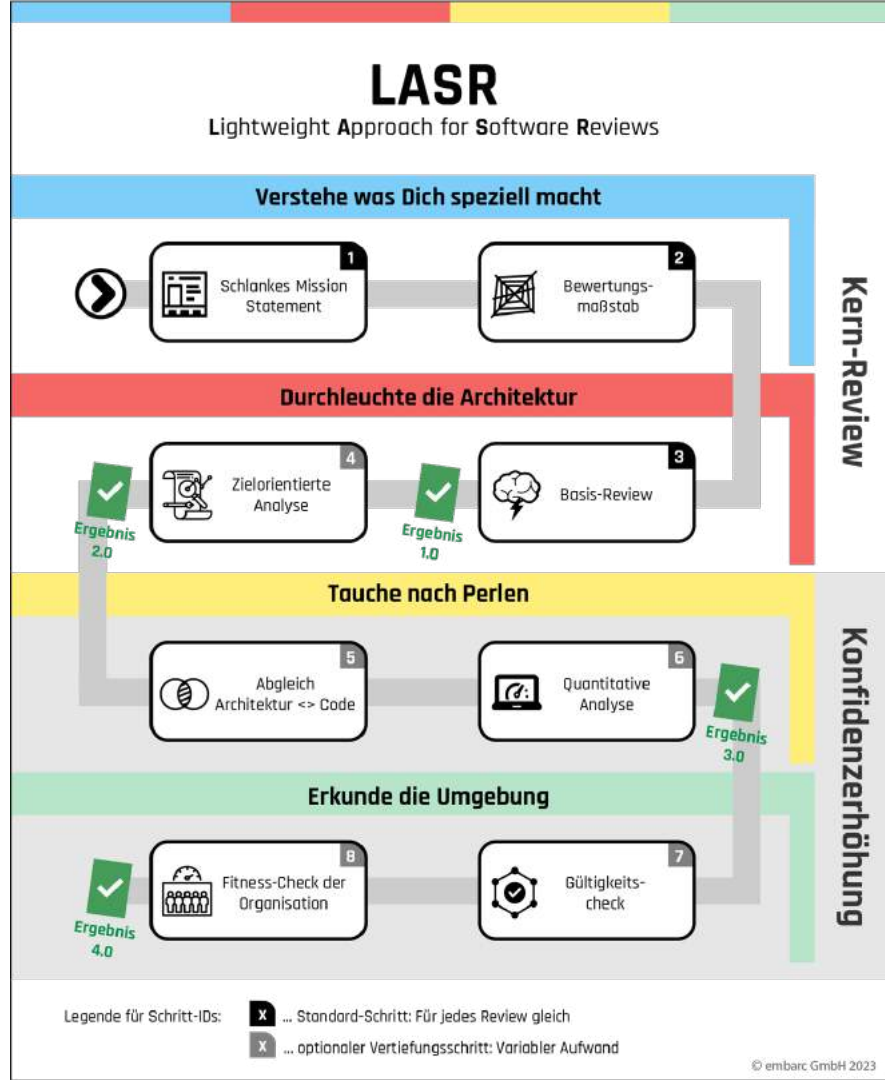
LASR

Lightweight Approach for Software Reviews

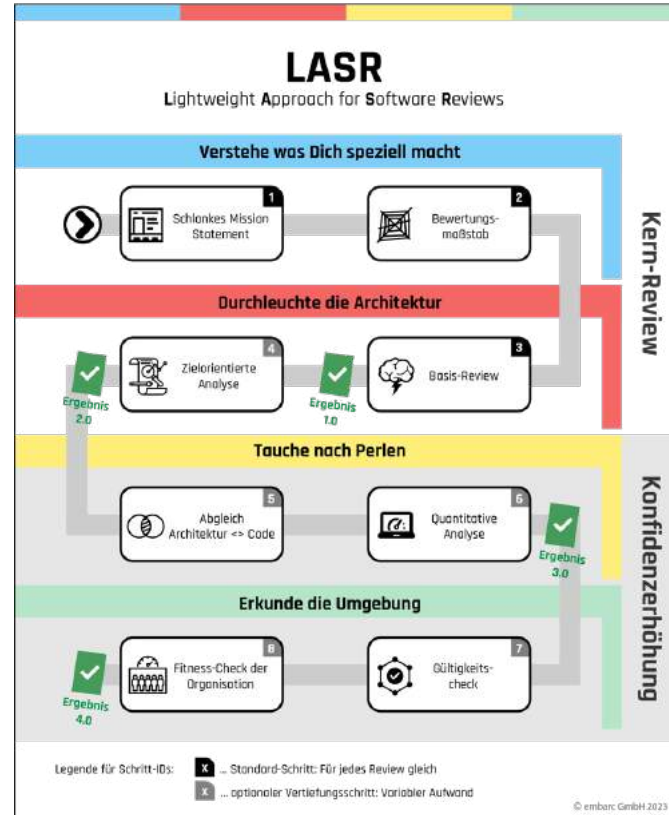
Ein schlanker Ansatz

LASR (Leightweight Approach for Software-Reviews) ist ein strukturiertes Vorgehen für leichgewichtige Software-Reviews.

Gesamtüberblick ➔



LASR Review



Kern-Review:

- Du erarbeitest bereits ein brauchbares Review-Ergebnis, das für 80% aller typischen Anlässe ausreichend ist.

Konfidenzerhöhung (optional):

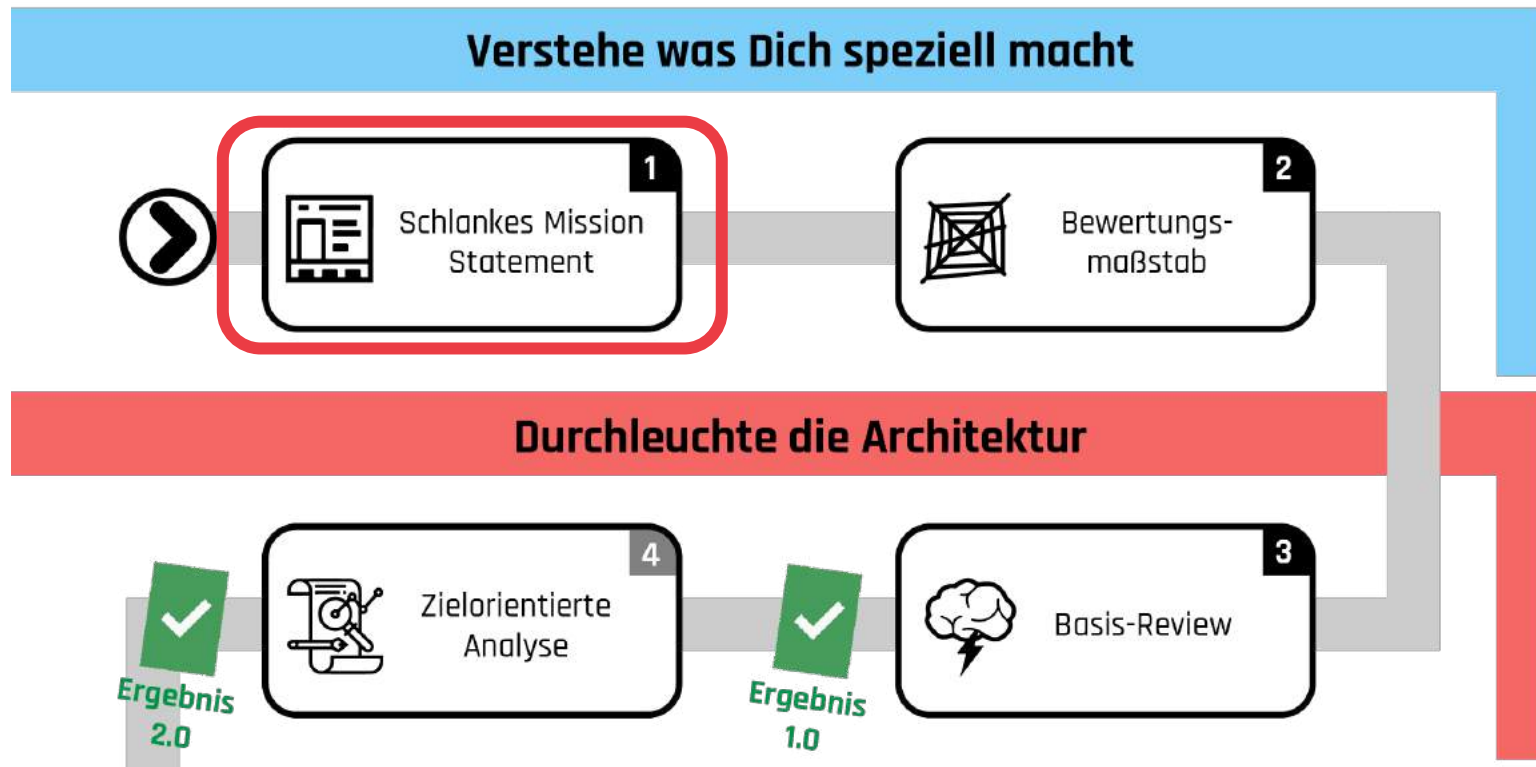
- Erhöht bei Bedarf Dein Vertrauen in das Bewertungsergebnis.
- Sucht intensiver auf Code-Ebene und Organisationsseite nach Problemen.

03.

LASR Schritte 1-2

Verstehe was Dich speziell
macht...

Schritt 1: Schlankes Mission Statement



Mission Statements auf Webseiten...



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



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."

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)

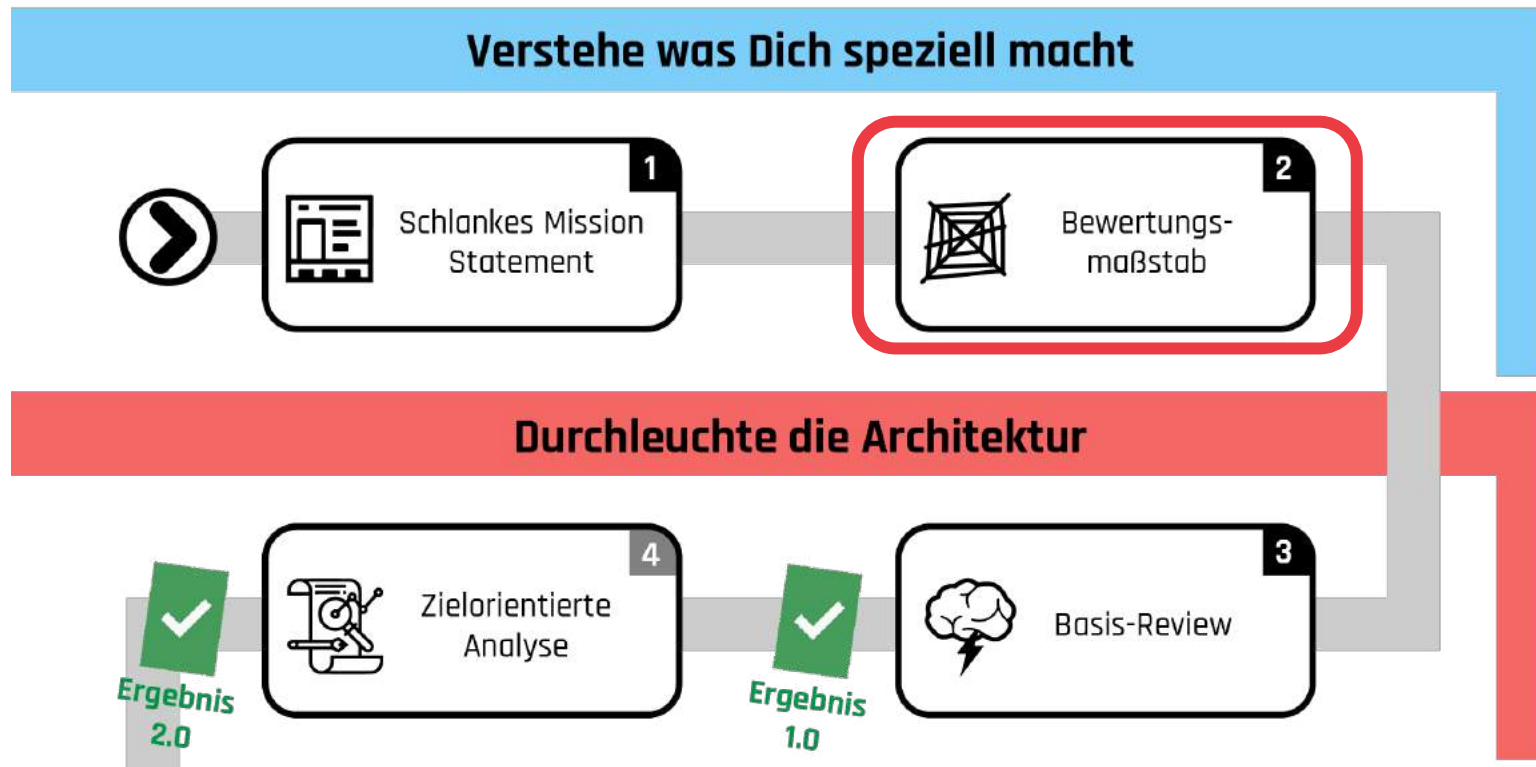
Ist das System sicher vor Angriffen?
Sind Daten und Funktion vor
unberechtigtem Zugriff geschützt? ...



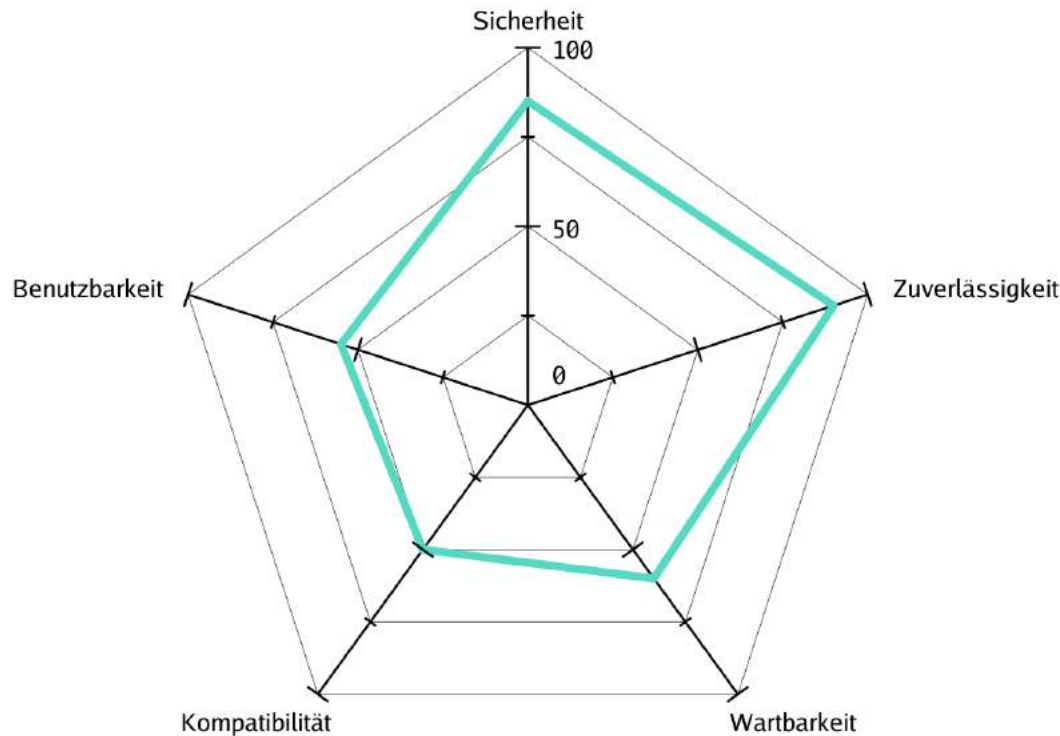
Wartbarkeit (Maintainability)

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

Schritt 2: Bewertungsmaßstab



(Beispiel-)Ergebnis für diesen Schritt





Umfrage in Mentimeter

Besuchen Sie menti.com | und benutzen Sie den Code **7440 2347**

**Leichtgewichtige und fokussierte
Software-Reviews**



Vortrag am Softwarearchitektur Meetup in Nürnberg 2024,
Stefan Toth (embarc)



 **Mentimeter**

→ menti.com

Umfrage in Mentimeter



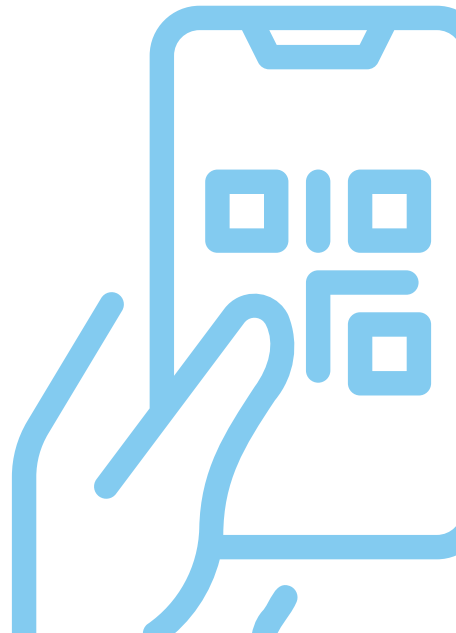
Besucht

menti.com

und gebt diesen
Code ein:

7440 2347

◀ Oder scannt
den QR-Code



Top-5-Challenger (aus 14 Karten die 5 wichtigsten wählen)



Start Top-5

Kandidaten

Nach 2 Runden, Beispiel (Beispiel)



Start Top-5

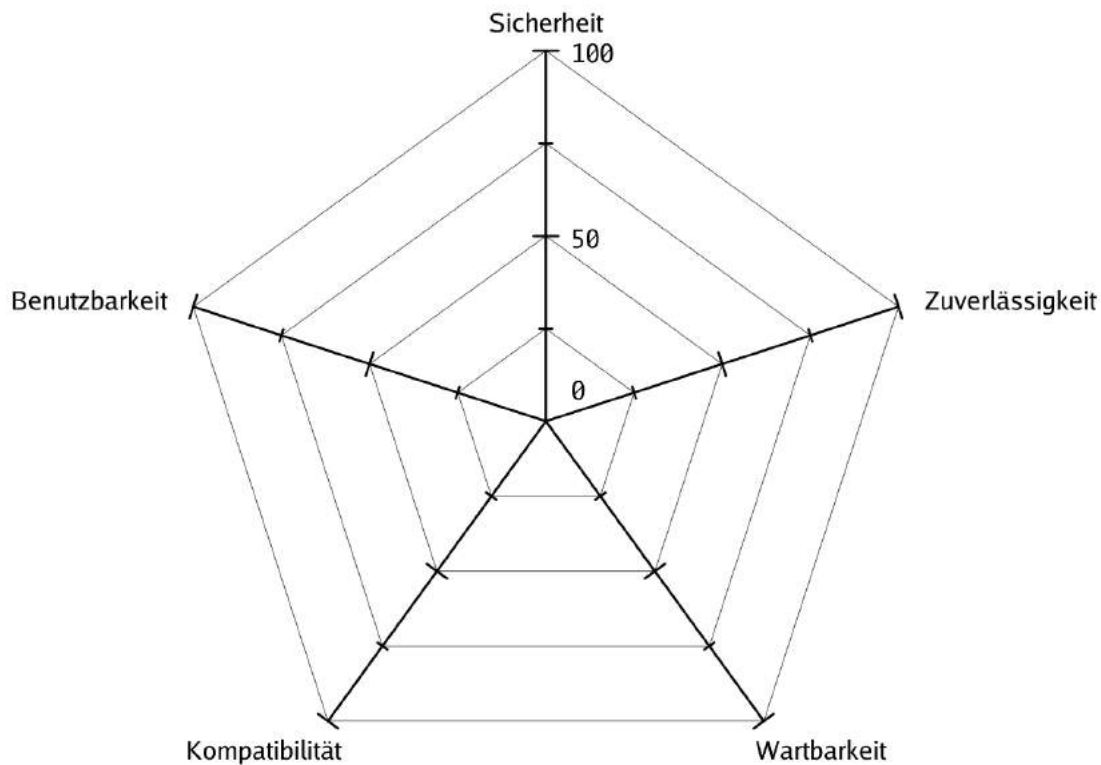


Ablagestapel



Kandidaten

Die Top 3-5 Qualitätsziele identifiziert (Beispiel)



Umfrage in Mentimeter



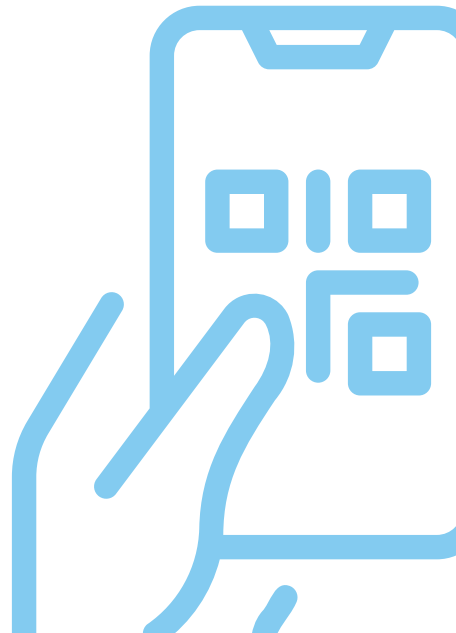
Besucht

menti.com

und gebt diesen
Code ein:

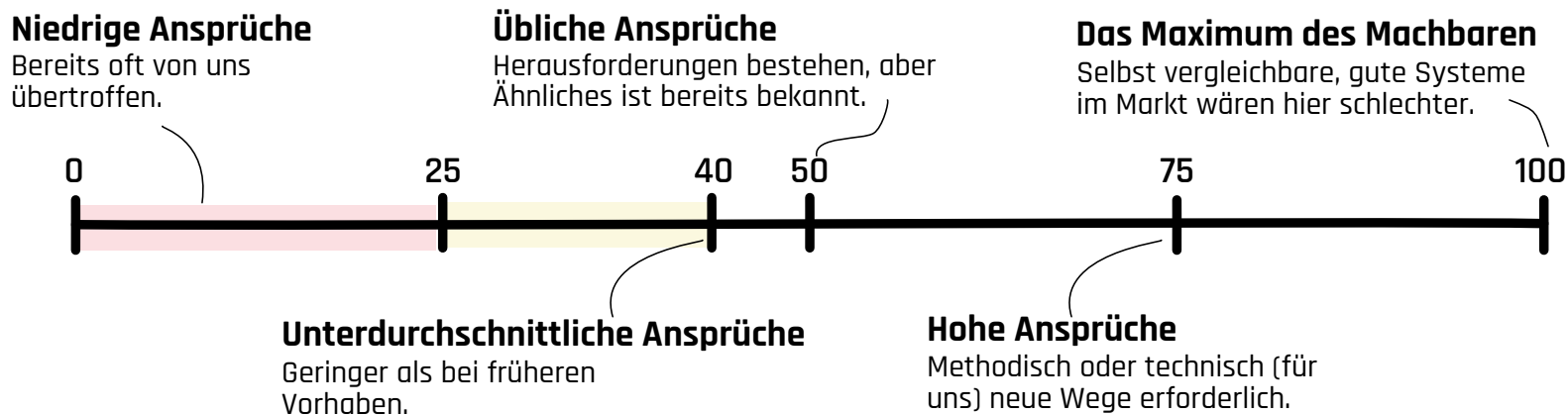
7440 2347

◀ Oder scannt
den QR-Code



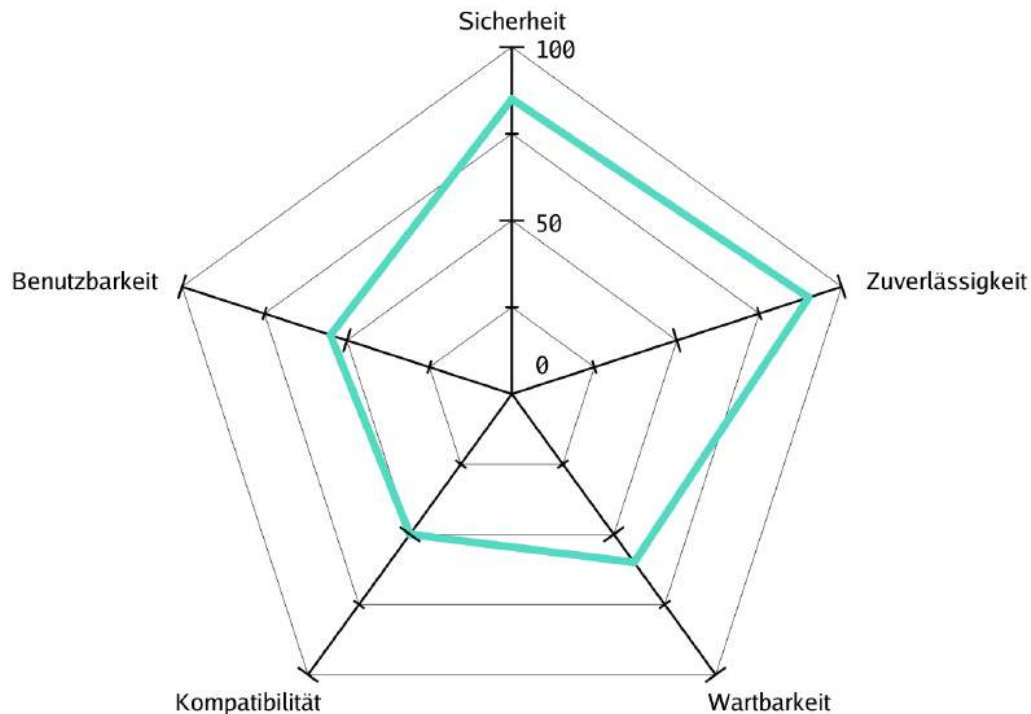
Zielwerte bestimmen

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 eingezeichnet (Beispiel)



LASR-Ergebnisdiagramm

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

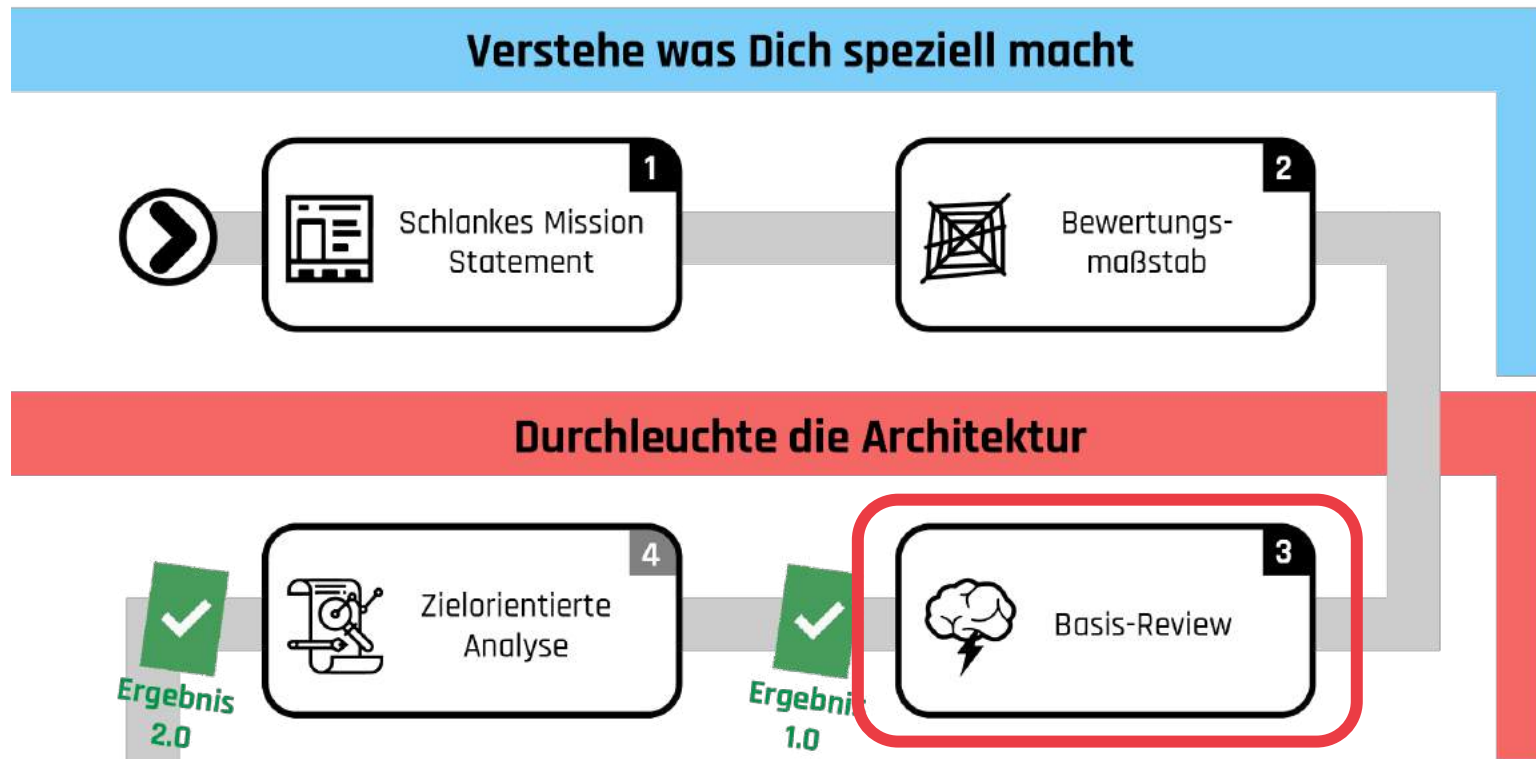
Die Zielwerte spannen darauf eine grüne Linie auf.

04.

LASR Schritte 3-4

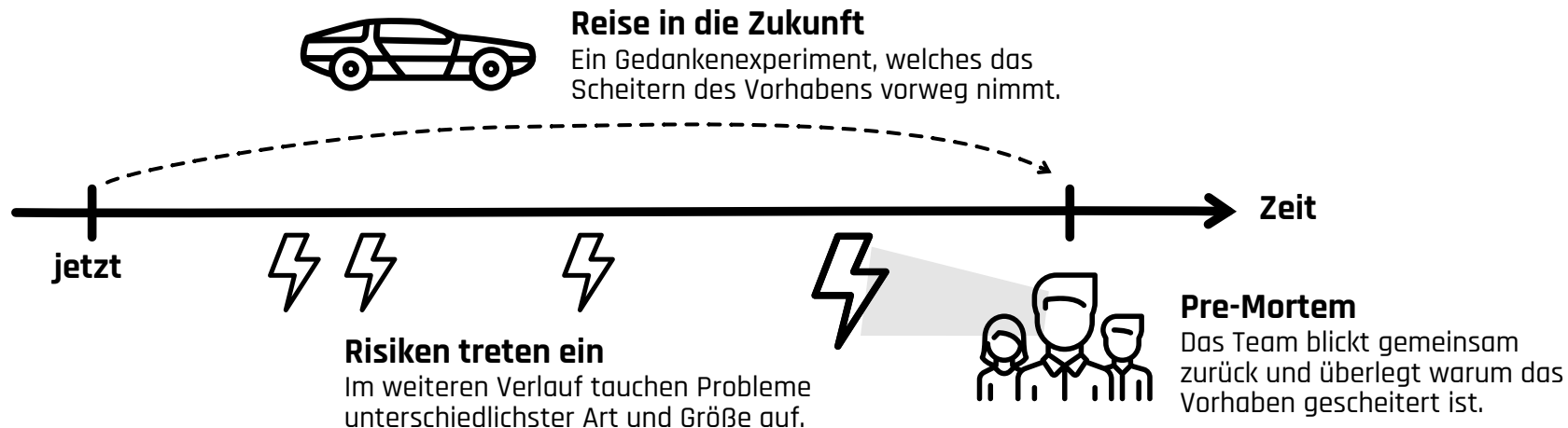
Durchleuchte die Architektur

Schritt 3: Basis-Review



Grundidee: Pre-Mortem?

Pre-Mortems sind proaktive Risikobewertungen eines möglichen Misserfolges. In einer Brainstorming-Session findet das Team heraus, warum sein Vorhaben scheitern könnte. Und es bewertet, wie wahrscheinlich und schwerwiegend die Gründe sind.



Typische Risikothemen als Ideengeber



Zu hohe Komplexität der Lösung

Hat die Domäne eine sehr hohe Komplexität? Gibt es unüberlegte, schnelle Lösungen oder fehlende Abstraktionen? Komplexität gefährdet den Überblick bzw. Wartbarkeit, Korrektheit, Sicherheit, ...

 Softwarelösung 1.1



Unpassende Lösungs-Strukturierung

Folgt die Softwarestruktur der Domäne? Sind andere technische oder organisatorische Einflüsse (Conway...) gut aufgegriffen? Falls nicht könnte Wartbarkeit, Zuverlässigkeit etc. leiden.

 Softwarelösung 1.2



Inadäquates Daten-Handling

Ist die Ablage, der Transport und das Mapping von Daten konzeptionell und technisch passend gelöst? Sind Daten ausreichend schnell und rechtssicher, im richtigen Format an den richtigen Stellen?

 Softwarelösung 1.3



Problematische Konzepte & Technologien

Passen die eingesetzten Muster und Konzepte zu den Zielen? Sind sie konsistent angewendet? Sind die verwendeten Technologien und Frameworks passend und etabliert?

 Softwarelösung 1.4



Softwarelösung

1

- Unpassende Technologien
- Komplexe Lösungen
- Unausgereifte Fremdlösungen
- Behindernde Frameworks / Bibliotheken
- Strukturprobleme
- ...


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

Die 8 Risiko-Kategorien von LASR




Softwarelösung 1

- Unpassende Technologien
- Komplexe Lösungen
- Unausgereifte Fremdlösungen
- Behindernde Frameworks / Libraries
- Strukturprobleme
- ...



Kompetenz und Erfahrung 2

- Fehlendes oder isoliertes Wissen
- Zu kleine Lernfenster
- Fehlendes Prozessverständnis
- Tool-Probleme
- Schwere Schätzbarkeit
- ...



Zielsetzungen und Erwartungen 3

- Unrealistische Ziele
- Überzogene Erwartungen
- Knappe Deadlines
- Instabile Anforderungen
- Fehlender Kundenkontakt
- ...



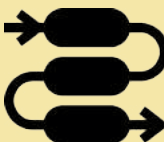
Fremdsysteme und Plattformen 4

- Instabile Plattformen
- Fehleranfällige Fremdsysteme
- Unpassende Buy/Take Wahl
- Lizenzschwierigkeiten
- Vendor Lock-In
- Integrationsprobleme
- ...



Altsysteme und Altlasten 5

- Legacy-Behinderungen
- Innovationsstau
- Zerbrechliche Lösungen
- Fehlende Tests
- Wenig Dokumentation
- Fehlendes Verständnis
- ...



Organisation und Prozesse 6

- Geschwollene Prozesse
- Behindernde Rollenmodelle
- Organisationsgrenzen
- Politik
- Einengende Standards
- Isolierte Entscheidungskompetenzen
- ...



Betrieb und Deployment 7

- Blockierende Prozesse
- Geringe Automatisierung
- Wenig Feedback
- Fehlende Betriebs- / Ausfallkonzepte
- Tool-Probleme
- ...



Weiche Faktoren 8

- Uneinigkeiten
- Konflikte
- Fehlende Disziplin
- Kommunikationsbarrieren
- Rollenethematiken
- Unpassende Kultur
- ...

Hypothetisches (!) Problem für Netflix



Vendor-Lock-In oder Support-Probleme

Gibt es Abhängigkeiten von Lieferanten, deren Ziele oder Einsatzzwecke abweichen? Stehen Abkündigungen im Raum? Gibt es potentielle Probleme bei Support, Lizenzmodellen, Kosten, ...?

 **Fremdsysteme & Plattformen** 4.4



Streaming-Anbieter wie Netflix werden von Amazon als Konkurrenz zu Prime gesehen. Die AWS Preispolitik ist für Großkunden so umgestaltet worden, dass Netflix nicht mehr profitabel betrieben werden kann..

Umfrage in Mentiometer



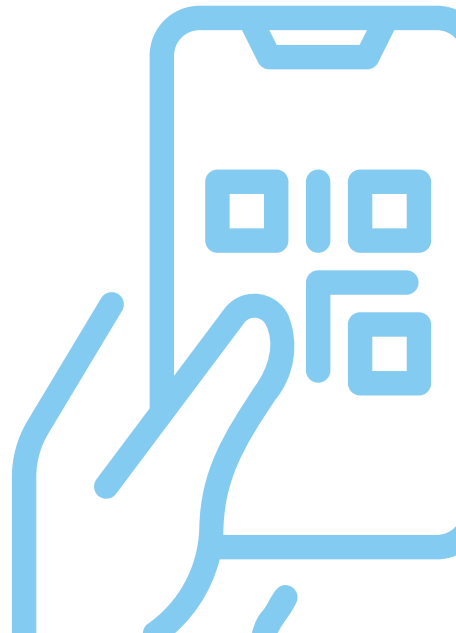
Besucht

menti.com

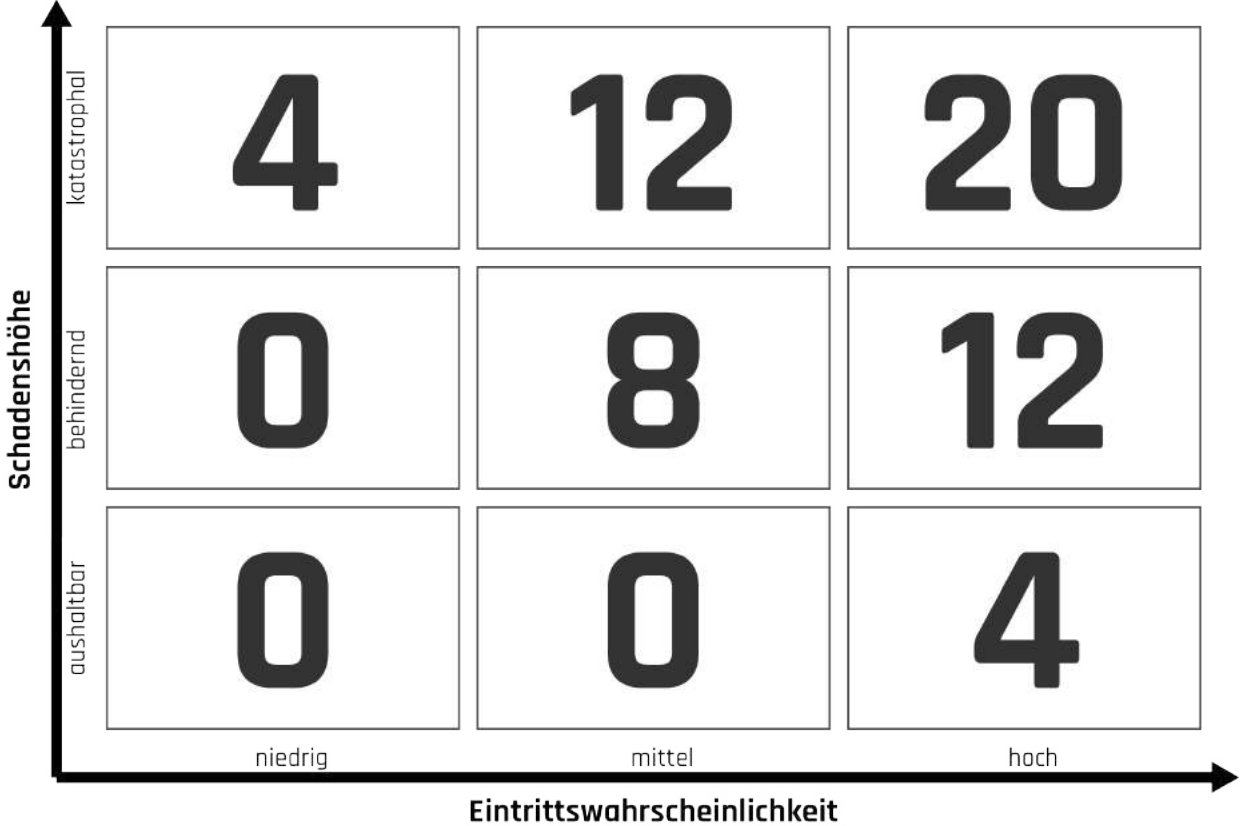
und gebt diesen
Code ein:

7440 2347

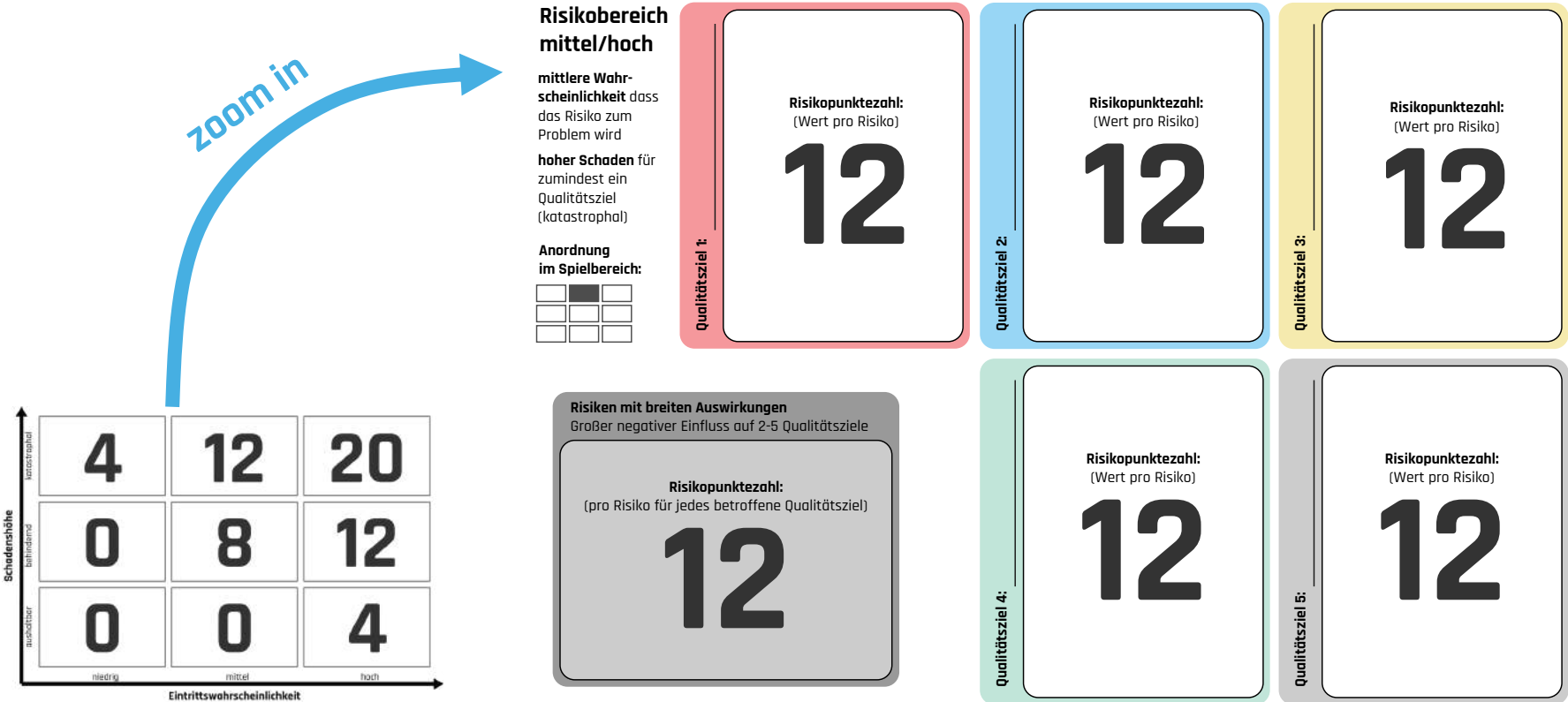
◀ Oder scannt
den QR-Code



Risikokarten einordnen



Den betroffenen Zielen zuordnen





Die „Lücke“ quantifizieren



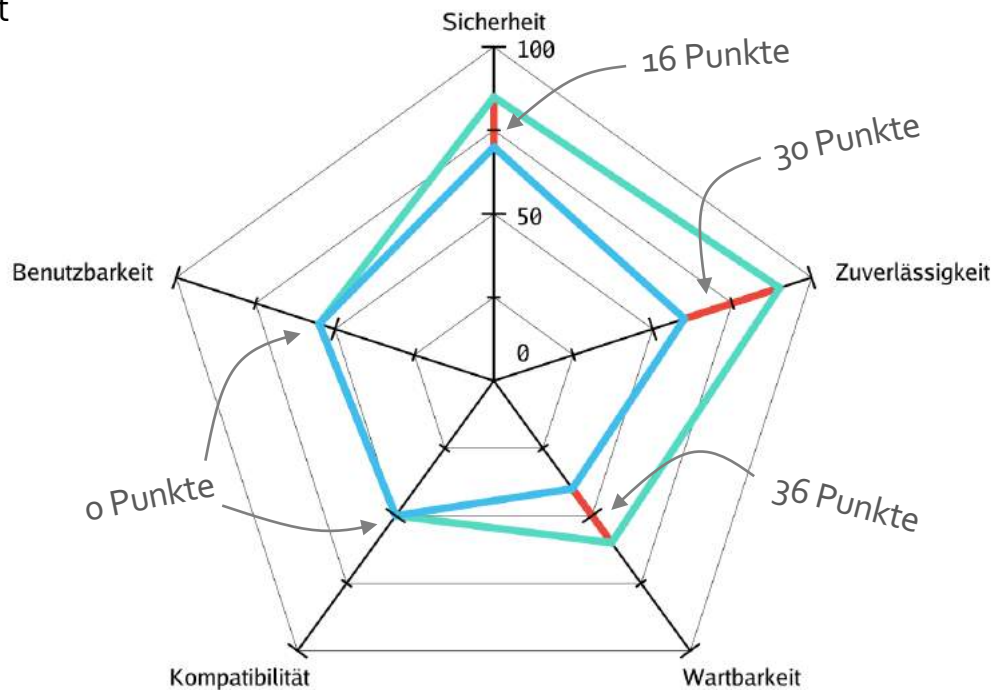
$4 \times 2 + 12 + 8 = 28$



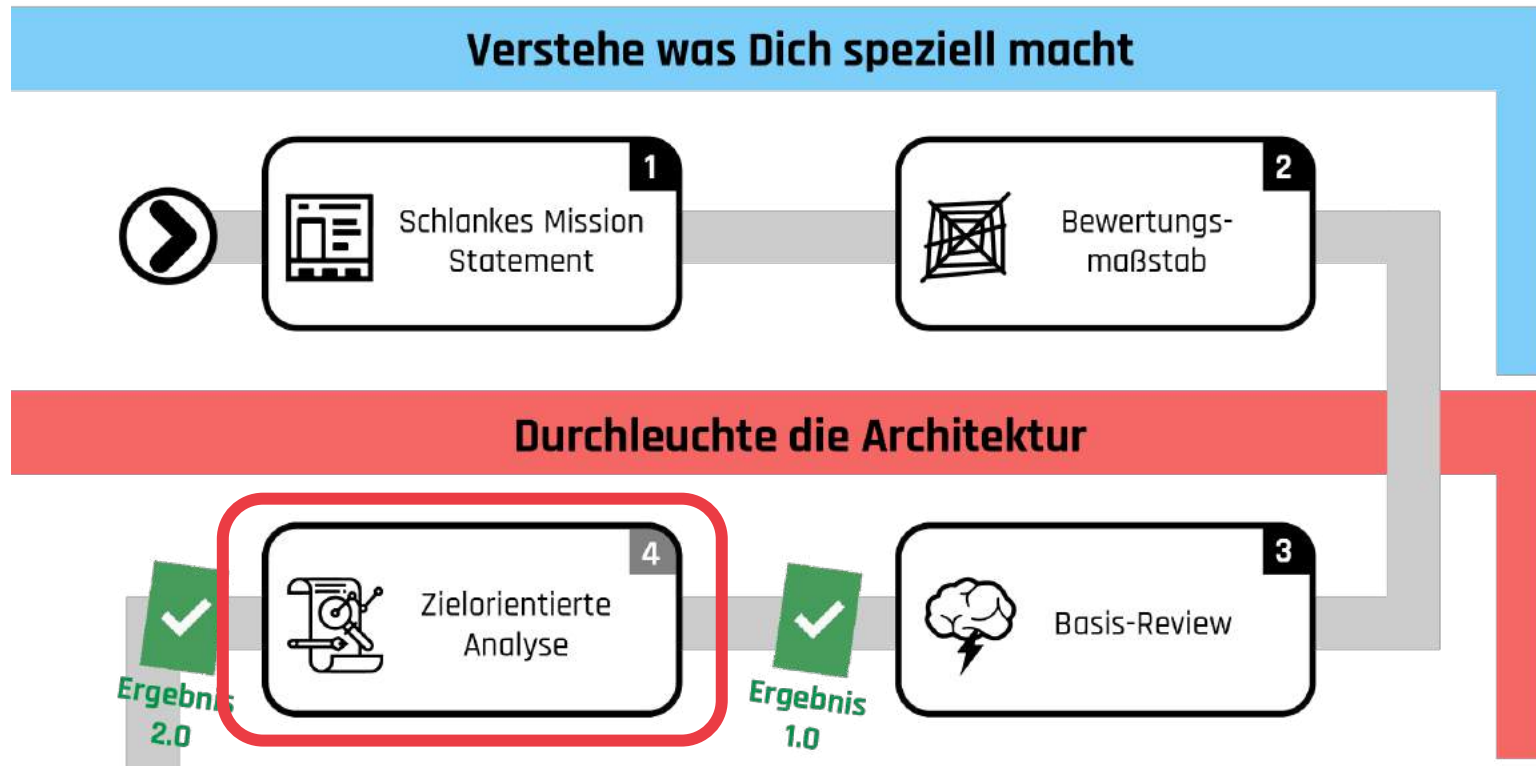
28% Abweichung vom Ziellevel
Auf der Qualitätszielachse

Lücken einzeichnen

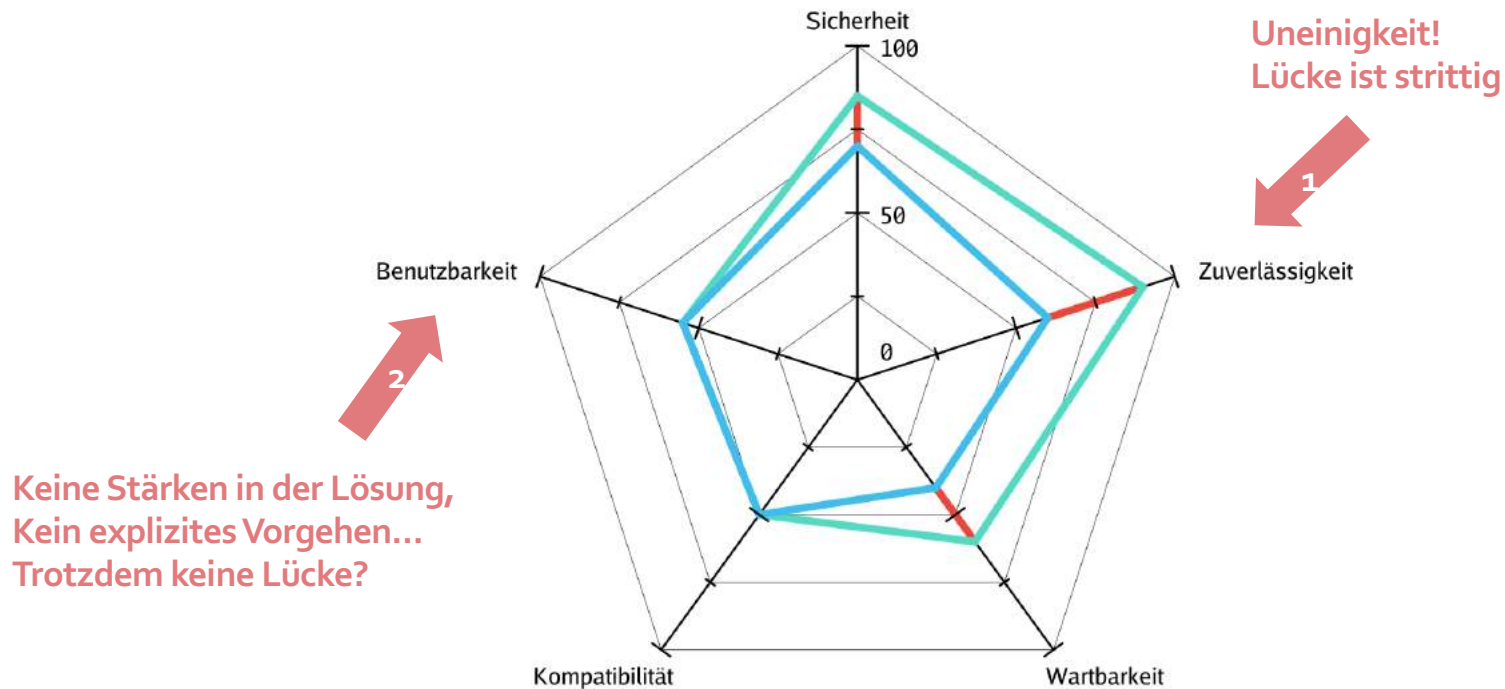
Gesamtpunktezahl der dem Qualitätsziel zugeordneten Karten = Abweichung von grüner Linie in Prozent
($>100 = 100$)



Schritt 4: Zielorientierte Analyse



Fokus für Zielorientierte Analyse - Bsp



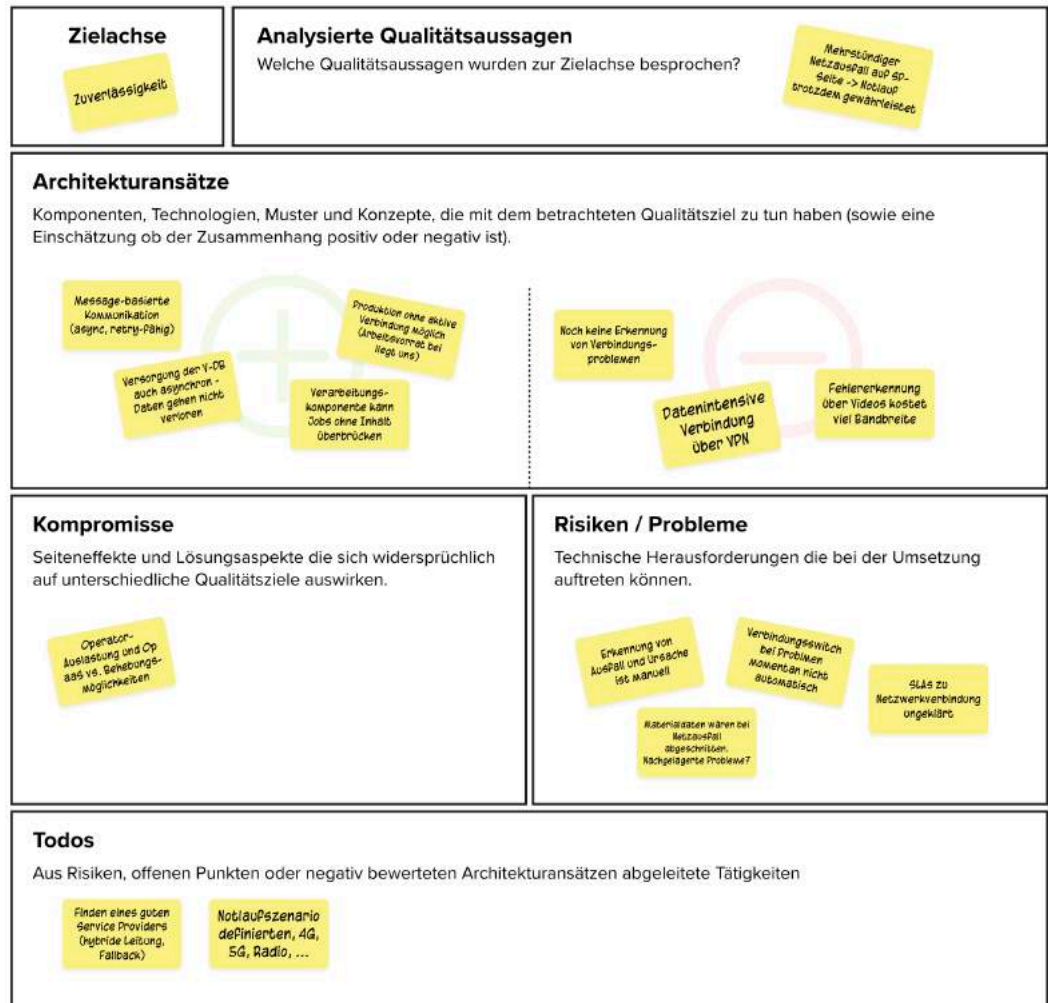
Analyse

Anhand wichtiger **Qualitätsaussagen**:

- Architekturansätze finden
- Kompromisse aufdecken
- Weitere Risiken finden

Anschließend ggf. auf **Lösungsebene** tiefer graben:

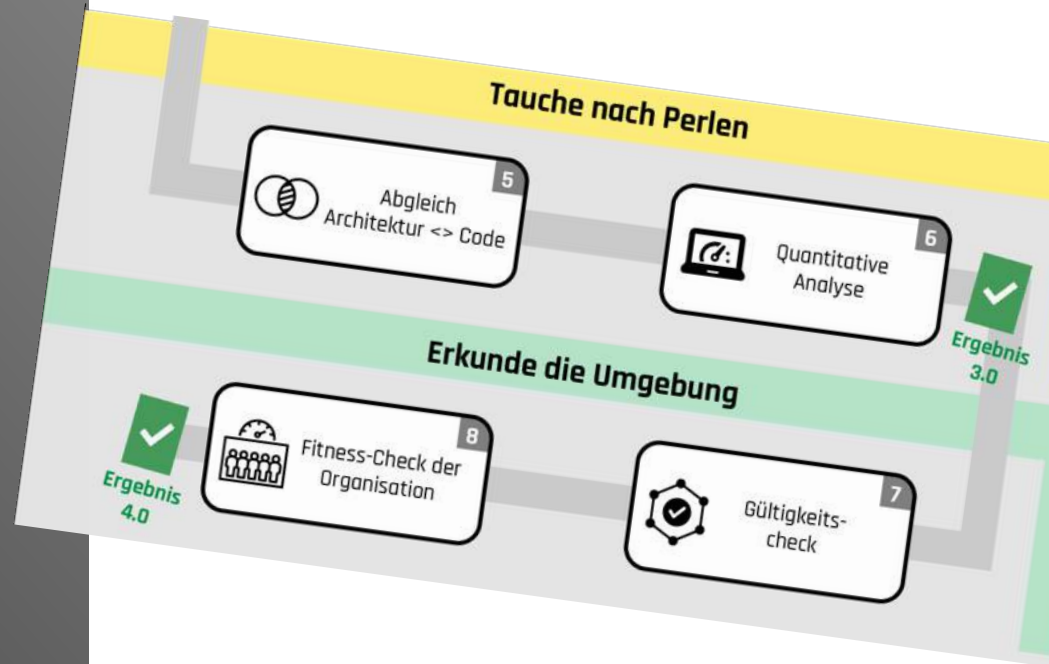
- „Code-Stöbern“
- Vorhandene Messungen nutzen



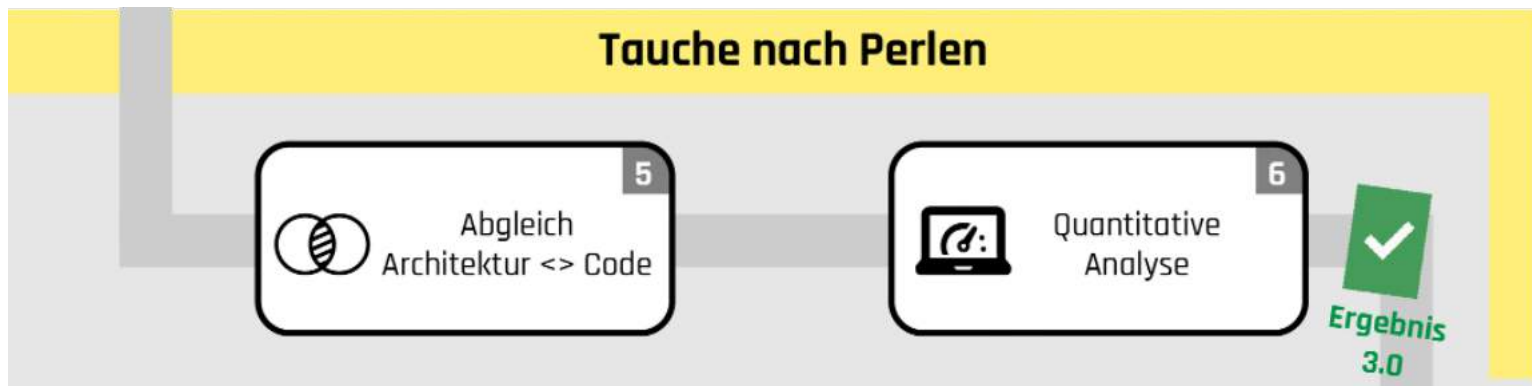
05.

Konfidenzerhöhung

Für wichtige Anlässe...



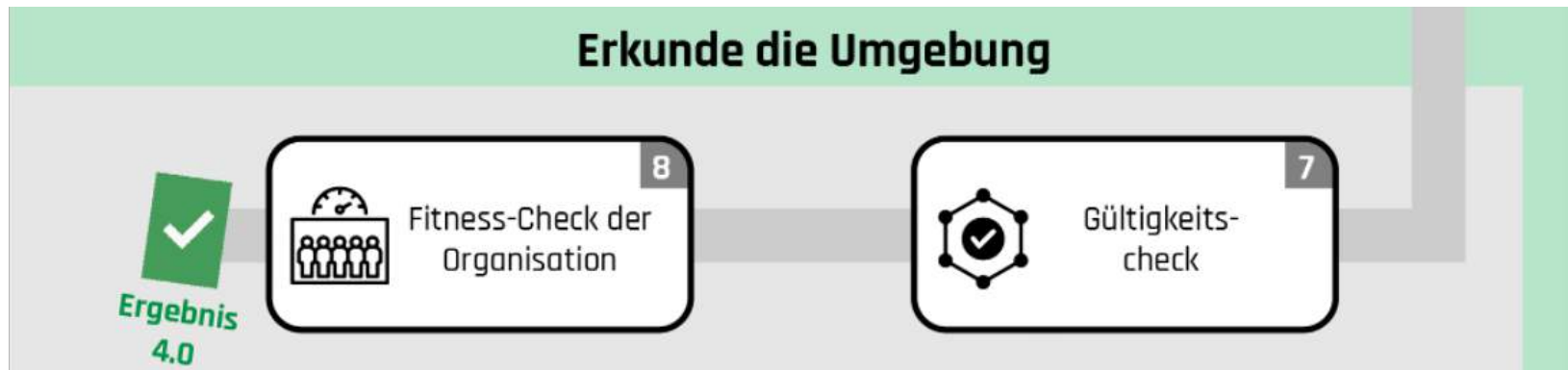
Tauche nach Perlen



Wann anwenden?

- Wenn über Quelltext und Implementierung bisher nur oberflächlich gesprochen wurde, Wartbarkeit aber zu den Top-Zielen gehört.
- Wenn große Lücken in Themenbereichen identifiziert sind, wo Messungen deutlich mehr Gewissheit bringen (z.B. Performance)

Erkunde die Umgebung



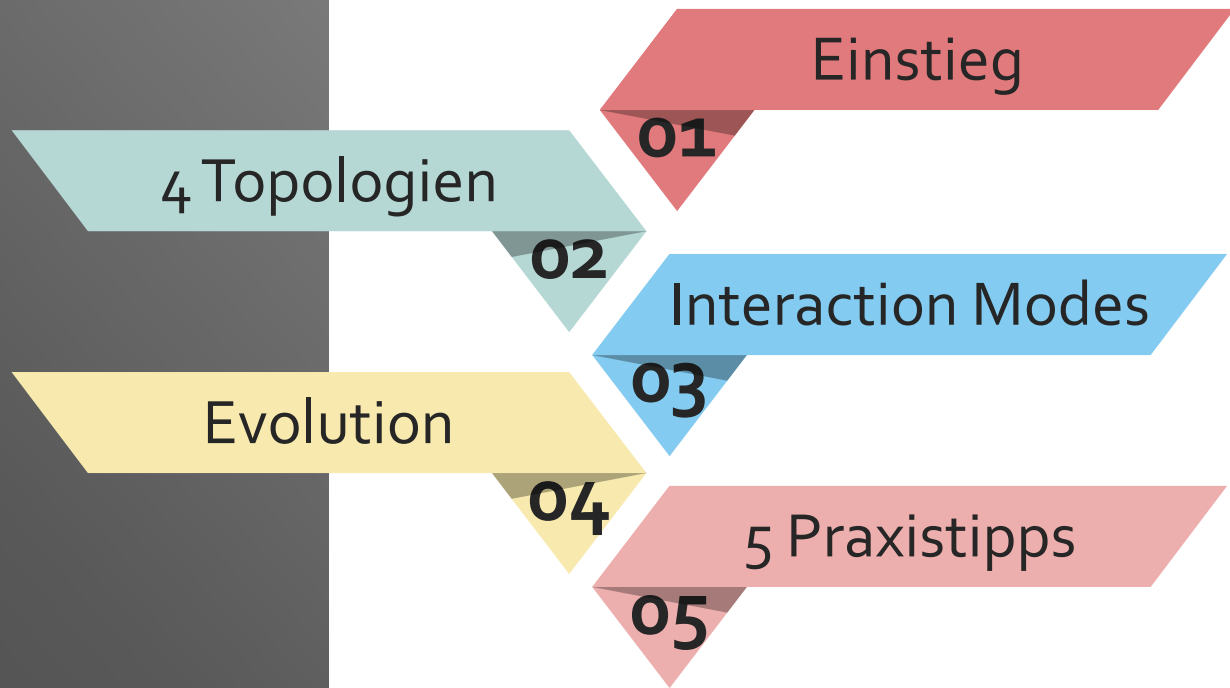
Wann anwenden?

- Wenn sich bereits im Basis-Review Probleme offenbaren die mit Organisation, Methodik oder Rahmenbedingungen zu tun haben.
- Wenn der Kontext in den bisherigen Schritten nur dünn beleuchtet wurde, das Review-Ergebnis aber belastbar sein muss (z.B. wegen darauffolgenden Management-Entscheidungen)

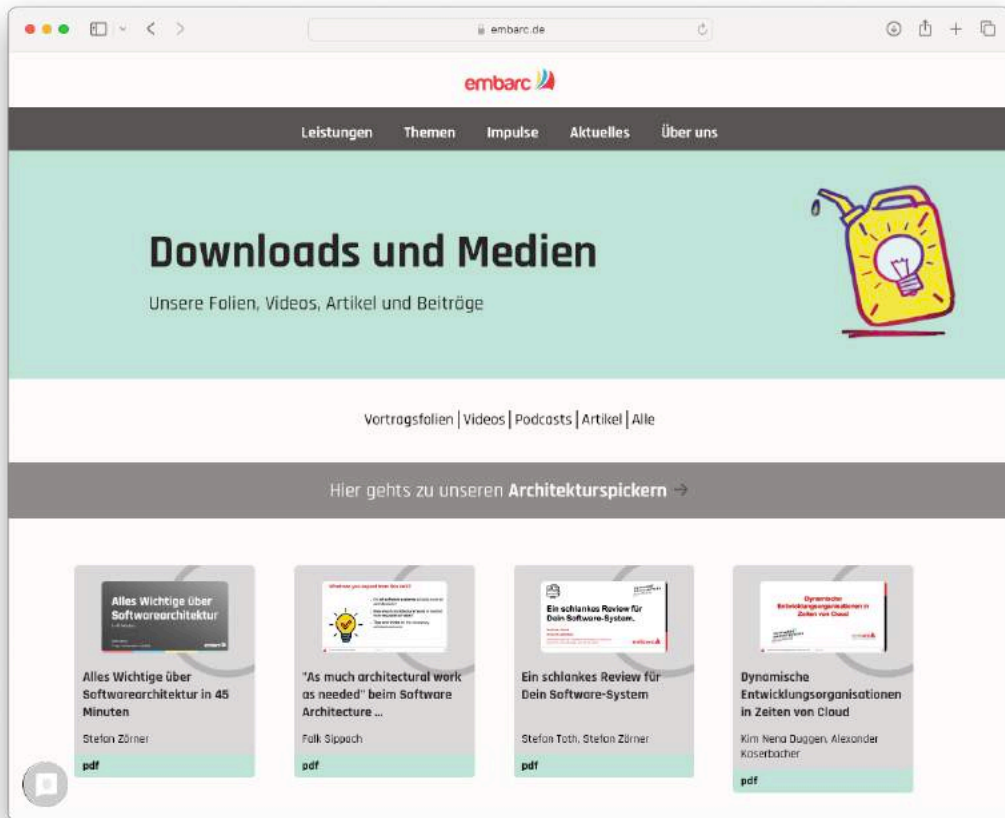
00.

Das wars...

Weitere Informationen



Folien als PDF zum Download



embarc.de/download/

Ein Promo-Code fürs Buch...



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

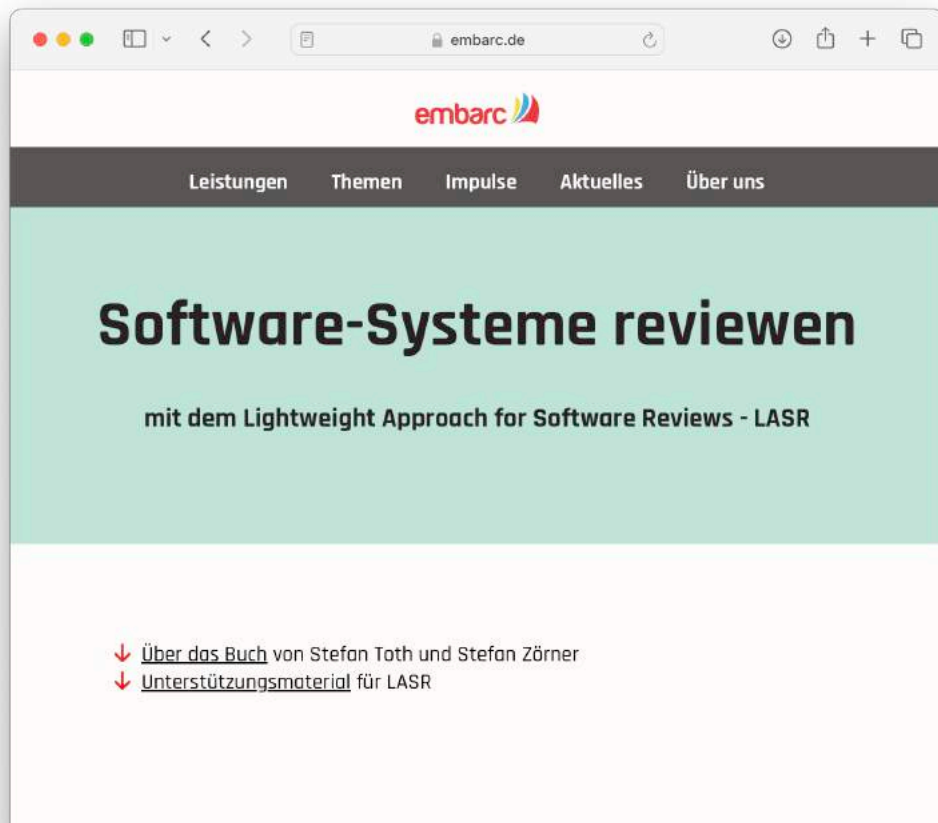

Leanpub



→ leanpub.com/software-systeme-reviewen/

\$7.99 statt \$17.99

Seite zum Buch bei embarc



→ embarc.de/lasr-booklet/

Feedback & Fragen?

Ich freue mich auf Fragen,
Diskussionen, Pizza!

