

Software-Systeme zielgerichtet bewerten mit LASR

Stefan Zörner, embarc

JUG Saxony Day, Radebeul, 27. September 2024

embarc 

Abstract

Software-Systeme zielgerichtet bewerten mit LASR

Ein leichtgewichtiger Ansatz für euer Software-Review

Mit Architekturbewertungen ist es möglich, Schwächen und Potenziale von Softwarelösungen herauszuarbeiten, Entscheidungen abzusichern und Verbesserungsmaßnahmen zu bewerten. Klassische Analyseansätze aus diesem Umfeld wie ATAM sind fundiert, kommen aber gerade in beweglichen Softwarevorhaben etwas schwergewichtig, mitunter fast zeremoniell daher.

In diesem Vortrag gebe ich einen lebendigen Einstieg in die Welt der Architekturbewertung. Vor allem lernt Ihr eine leichtgewichtige Herangehensweise kennen. Ihr könnt diese mit Eurem Team unmittelbar anwenden, Euer Softwaresystem beleuchten und zügig zu ersten Erkenntnissen kommen. Wir greifen auf die Essenzen etablierter Bewertungsmethoden zurück. Und erarbeiten uns einen roten Faden durch ein Review, inkl. möglicher Vertiefungspunkte für eine höhere Konfidenz im Bewertungsergebnis.



embarc.de

LASR. Ein leichtgewichtiger Ansatz

1



Stefan Zörner

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

Schwerpunkte

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



DO.

Agenda

Software-Systeme
zielgerichtet
bewerten mit LASR

01. Warum leichtgewichtiger bewerten?

02. Verstehe, was Dich speziell macht

03. Durchleuchte die Architektur

04. Erhöhe die Konfidenz (bei Bedarf)

05. Weitere Informationen

01.

Warum leichtgewichtig bewerten?

01. Warum leichtgewichtig bewerten?

02. Verstehe, was Dich speziell macht

03. Durchleuchte die Architektur

04. Erhöhe die Konfidenz (bei Bedarf)

05. Weitere Informationen



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

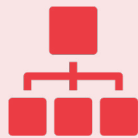
embarc.de

LASR. Ein leichtgewichtiger Ansatz

6

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?

embarc.de

LASR. Ein leichtgewichtiger Ansatz

7



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.



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?



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

embarc.de



Anlass 1 („Neuanfang“)

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

Leitfrage:

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

LASR. Ein leichtgewichtiger Ansatz

10



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

embarc.de



Anlass 2 („Wünsche“)

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

Leitfrage:

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

LASR. Ein leichtgewichtiger Ansatz

11



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

embarc.de



Anlass 3 („Unruhe“)

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

Leitfrage:

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

LASR. Ein leichtgewichtiger Ansatz

12



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

embarc.de



Anlass 4 („Legacy“)

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

Leitfrage:

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

LASR. Ein leichtgewichtiger Ansatz

13



Das Leistungsversprechen von Reviews

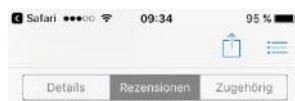
In all diesen Situationen unterstützen Review-Ansätze und helfen Euch und Eurem Team die Leitfragen zu beantworten.

Konkret: Software-Reviews ...

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



Nörgeln ist einfach ...



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

5. Keine gifs
★★★★☆ guthid - 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)

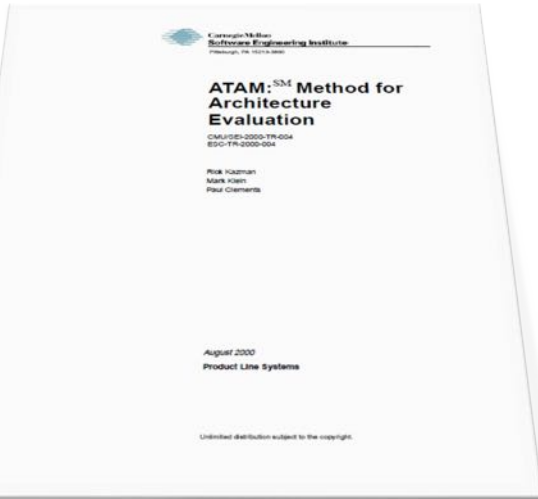
SAAM Software Architecture Analysis Method	ARID Architecture Review for Intermediate Designs
ATAM Architecture Tradeoff Analysis Method	DCAR Decision Centric Architecture Review
CBAM Cost-Benefit Analysis Method	DASE Decision and Scenario based architecture evaluation
TARA Tiny Architecture Review Approach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



ATAM

embarc.de

Leichtgewichtiger Ansatz



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



Herausforderungen ...

embarc.de

LASR. Ein leichtgewichtiger Ansatz

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**.

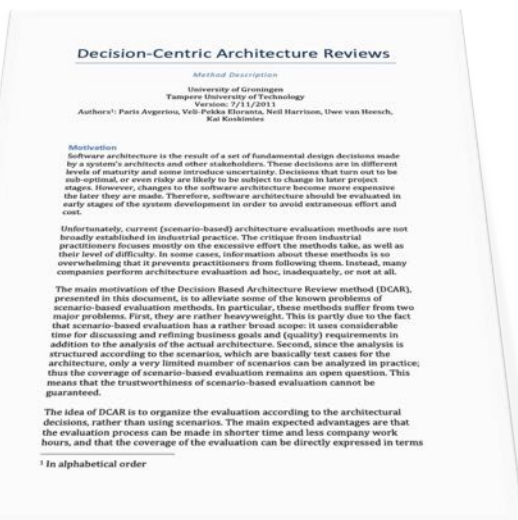


19

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

DCAR



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



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



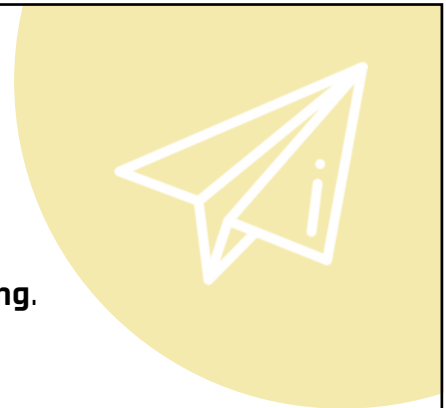
Was ist leichtgewichtig?

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

embarc.de

LASR - Ein leichtgewichtiger Ansatz

24



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

embarc.de

LASR - Ein leichtgewichtiger Ansatz

25

- 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



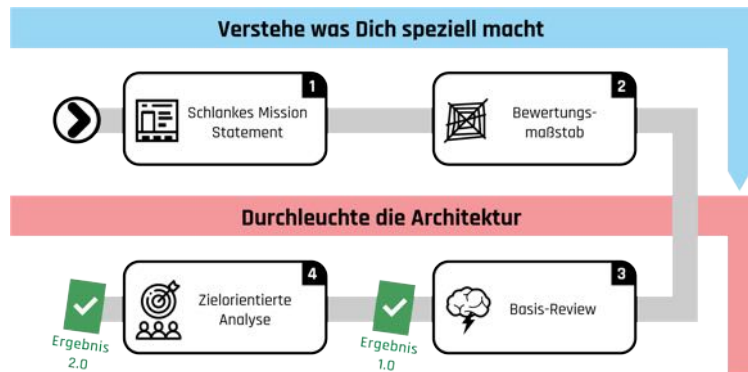


Ein schlanker Ansatz

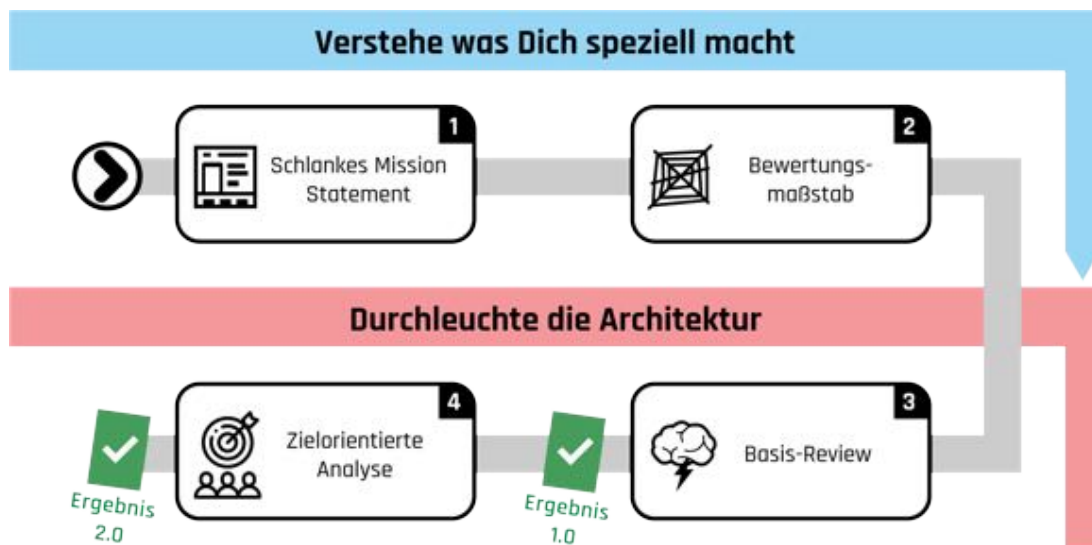
LASR (Lightweight Approach for Software-Reviews) ist ein strukturiertes Vorgehen für leichtgewichtige Software-Reviews.

LASR Kern-Review →

- 2 Tätigkeiten
- 4 Schritte
- Ein frühes erstes Ergebnis



Tätigkeiten und Schritte im Kern-Review



02.

Verstehe, was Dich speziell macht

01. Warum leichtgewichtig bewerten?

02. Verstehe, was Dich speziell macht

03. Durchleuchte die Architektur

04. Erhöhe die Konfidenz (bei Bedarf)

05. Weitere Informationen



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



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?



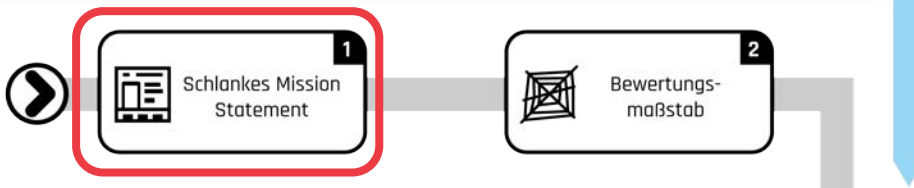
embarc.de

LASR. Ein leichtgewichtiger Ansatz

30

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



embarc.de

LASR. Ein leichtgewichtiger Ansatz

31



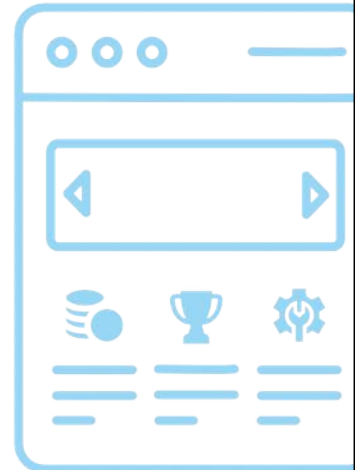
Mission Statement

Der erste Schritt des Reviews **macht** den **Bewertungsgegenstand** mit Hilfe eines schlanken Mission Statements **explizit**.

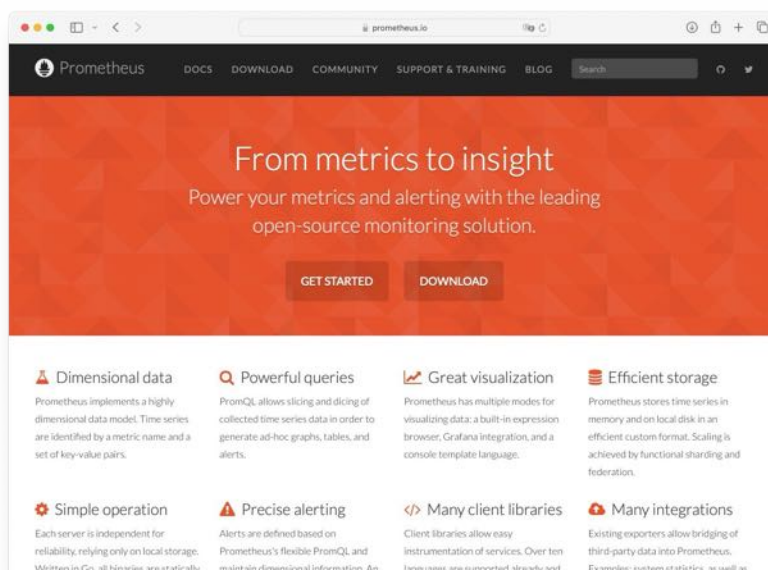
Landing-Page als Metapher

Ziel: Eine prägnante und kraftvolle Beschreibung des Systemzwecks.

Die Landing-Pages prominenter Software-Produkte bieten **Orientierung** bei der Erstellung.



Beispiel: Landing Page von Prometheus



Prometheus DOCS DOWNLOAD COMMUNITY SUPPORT & TRAINING BLOG Search

From metrics to insight

Power your metrics and alerting with the leading open-source monitoring solution.

[GET STARTED](#) [DOWNLOAD](#)

Dimensional data

Prometheus implements a highly dimensional data model. Time series are identified by a metric name and a set of key-value pairs.

Powerful queries

PromQL allows slicing and dicing of collected time series data in order to generate ad-hoc graphs, tables, and alerts.

Great visualization

Prometheus has multiple modes for visualizing data: a built-in expression browser, Grafana integration, and a console template language.

Efficient storage

Prometheus stores time series in memory and on local disk in an efficient custom format. Scaling is achieved by functional sharding and federation.

embarc.de

LASR. Ein leichtgewichtiger Ansatz

35

Erarbeiten im Workshop

LASR Schritt 1: Schlankes Mission Statement

Was würde prominent auf der Landing Page der zu betrachtenden Softwarelösung stehen?

1 Brainstorming

Leitfragen:
Was sind zentrale Features oder Eigenschaften unserer Software?
Was macht unsere Software besonders? Was besonders gut?
Welche Ansprüche haben wichtige Beteiligte an die Softwarelösung? ("Claims")

2 Clustern und Ausdünnen

Voting. Welche der gefundenen Punkte sollten Eurer Meinung nach auf jeden Fall auftauchen? Ziel: ca. 7 + 2.

← Beispiel im digitalen Whiteboard (Mural)

Quelle: S. Toth, S. Zörner: „Software-Systeme reviewen“, Leanpub 2023

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? ...



Kernelemente eines Reviews

Für eine Bewertung oder ein Review braucht es mindestens



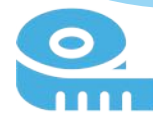
Einen Anlass

Warum bewerten wir?



Einen Gegenstand

Was bewerten wir?



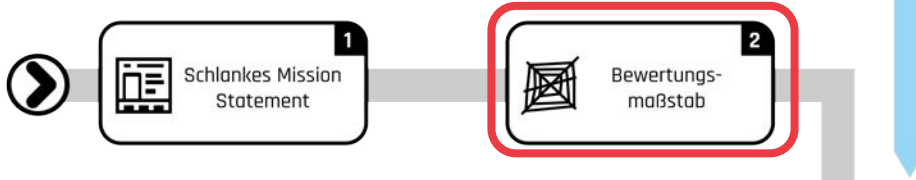
Einen Maßstab

Wonach (wo gegen) bewerten wir?



Verstehe, was Dich speziell macht

Verstehe was Dich speziell macht

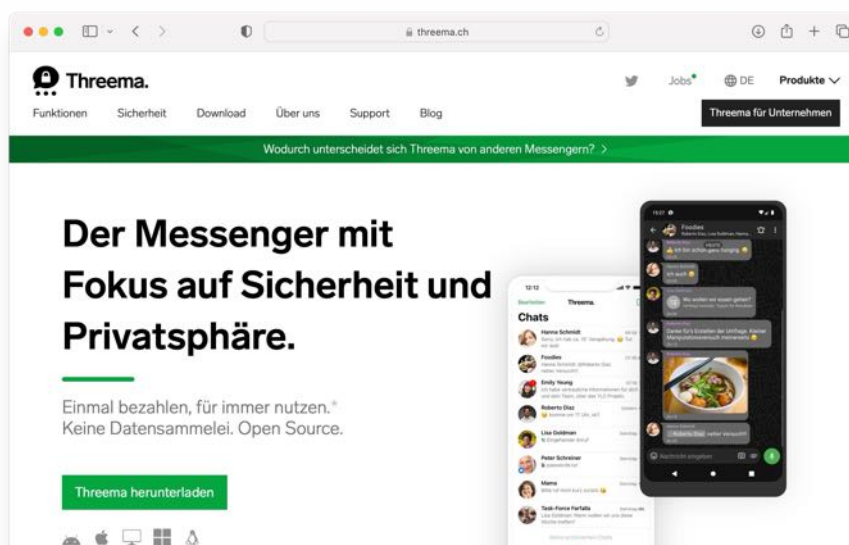


Was ist zu tun?

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








Weiteres Beispiel für eine Startseite





Die Qualitätsziele von Threema

- 
Sicherheit Kommunikations- und Nutzerdaten sind geschützt.
- 
Benutzbarkeit Einfach zu verwenden.
- 
Zuverlässigkeit Zuverlässig und effizient im Betrieb.
- 
Kompatibilität Interoperabel über alle Plattformen hinweg.
- 
Wartbarkeit Leicht erweiterbar und anpassbar.

Quelle: „Architektur-Porträt: Threema“, Stefan Zörner 2023





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



Ablauf einer Runde

embarc.de

LASR. Ein leichtgewichtiger Ansatz

44



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.



Nach 2 Runden, Beispiel (Beispiel)

embarc.de

LASR. Ein leichtgewichtiger Ansatz

45



Aktuelle Top-5



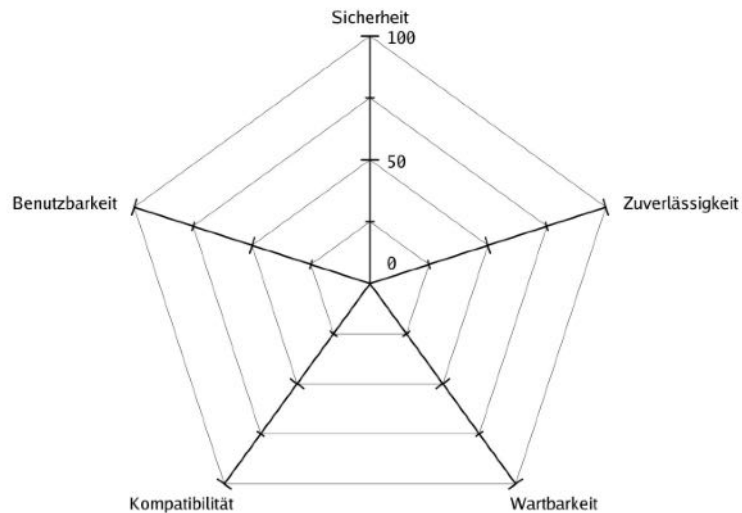
Ablagestapel



Kandidaten

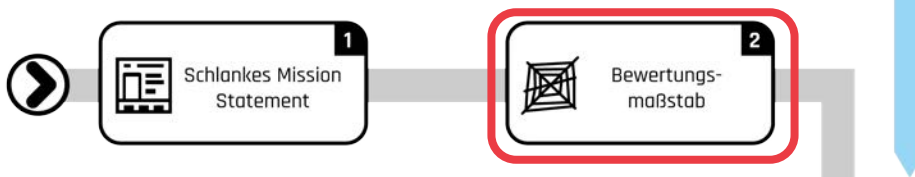


Ergebnis nach diesem Schritt (Beispiel)



Verstehe, was Dich speziell macht

Verstehe was Dich speziell macht

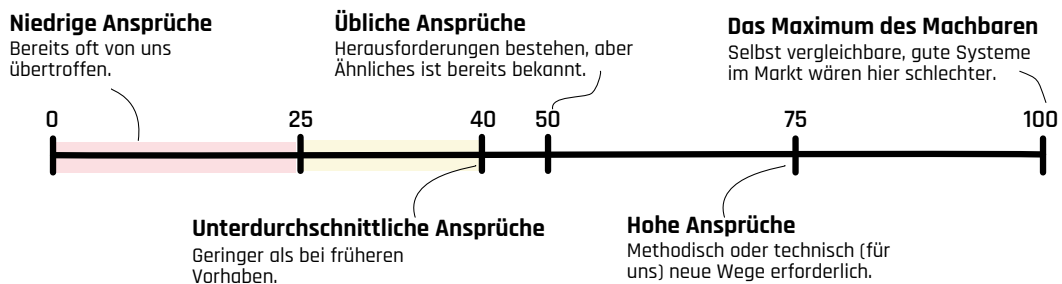


Was ist zu tun?

- 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

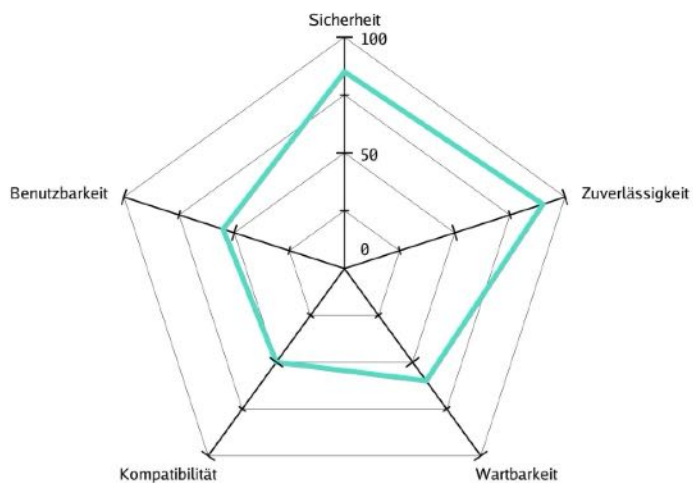
Der Zielwert zwischen 0 und 100 beschreibt jeweils, wie hoch die Erwartungen in dem Bereich sind. Im Vergleich zu anderen Zielen der Top-3-5 und anderen Vorhaben aus dem Kontext des Unternehmens / der Organisation.



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

embarc.de
LASR. Ein leichtgewichtiger Ansatz
48

Bewertungsmaßstab einzeichnen



LASR-Ergebnisdiagramm

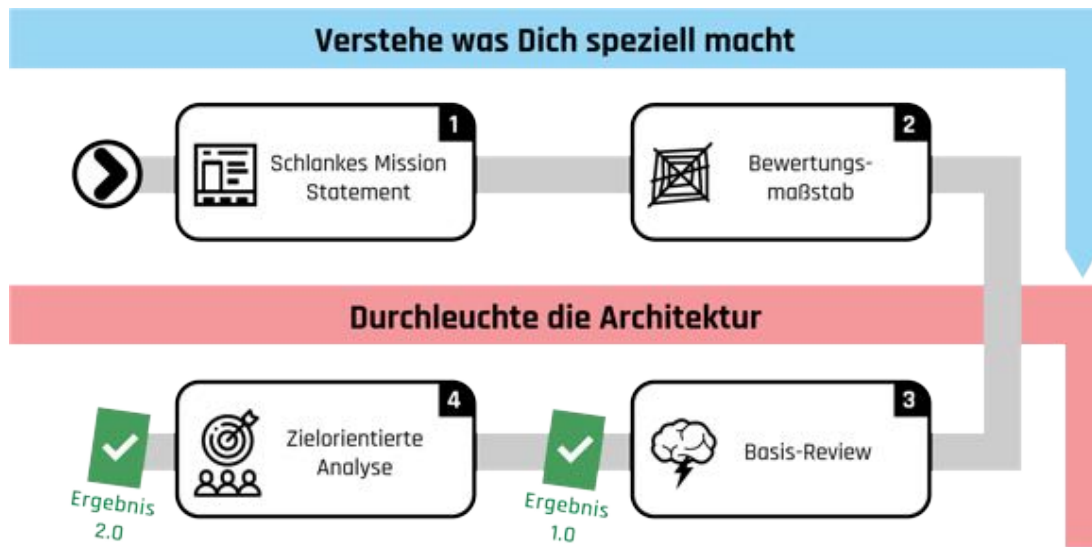
Die Top-3-5 Qualitätsziele bilden die Achsen des Diagramms.

Die Zielwerte spannen darauf eine grüne Linie auf.

← Beispiel

embarc.de
LASR. Ein leichtgewichtiger Ansatz
49

Mit Schritt 2 steht der Ergebnisrahmen.

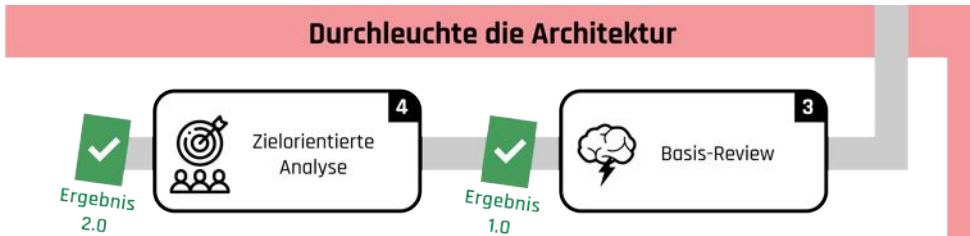


03.

Durchleuchte die Architektur

- 01. Warum leichtgewichtiger bewerten?
- 02. Verstehe, was Dich speziell macht
- 03. Durchleuchte die Architektur**
- 04. Erhöhe die Konfidenz (bei Bedarf)
- 05. Weitere Informationen

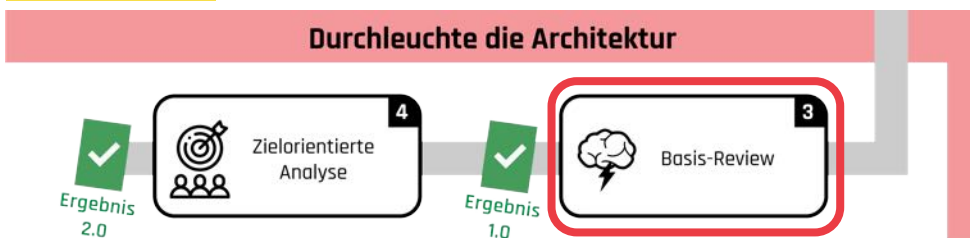
Durchleuchte die Architektur



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

Durchleuchte die Architektur

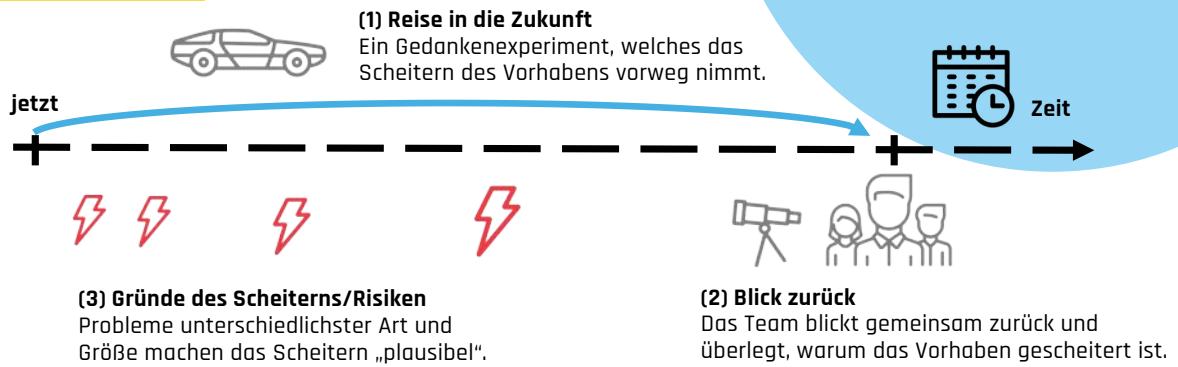


Was ist zu tun?

- **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

Was ist ein Pre-Mortem?

embarc.de
LASR. Ein leichtgewichtiger Ansatz
54



- 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

<p>Softwarelösung 1</p> <ul style="list-style-type: none"> • Unpassende Technologien • Komplexe Lösungen • Unausgereifte Fremdlösungen • Behindernde Frameworks / Libraries • Strukturprobleme • ... 	<p>Kompetenz und Erfahrung 2</p> <ul style="list-style-type: none"> • Fehlendes oder isoliertes Wissen • Zu kleine Lernfenster • Fehlendes Prozessverständnis • Tool-Probleme • Schwere Schätzbarkeit • ... 	<p>Zielsetzungen und Erwartungen 3</p> <ul style="list-style-type: none"> • Unrealistische Ziele • Überzogene Erwartungen • Knappe Deadlines • Instabile Anforderungen • Fehlender Kundenkontakt • ... 	<p>Fremdsysteme und Plattformen 4</p> <ul style="list-style-type: none"> • Instabile Plattformen • Fehleranfällige Fremdsysteme • Unpassende Buy/Make Wahl • Lizenzschwierigkeiten • Vendor Lock-in • Integrationsprobleme • ...
<p>Altsysteme und Altlasten 5</p> <ul style="list-style-type: none"> • Legacy-Behinderungen • Innovationsstau • Zerbrochene Lösungen • Fehlende Tests • Wenig Dokumentation • Fehlendes Verständnis • ... 	<p>Organisation und Prozesse 6</p> <ul style="list-style-type: none"> • Geschwollene Prozesse • Behindernde Rollenmodelle • Organisationsgrenzen • Politik • Einengende Standards • Isolierte Entscheidungskompetenzen • ... 	<p>Betrieb und Deployment 7</p> <ul style="list-style-type: none"> • Blockierende Prozesse • Geringe Automatisierung • Wenig Feedback • Fehlende Betriebs- / Ausfallkonzepte • Tool-Probleme • ... 	<p>Weiche Faktoren 8</p> <ul style="list-style-type: none"> • Unregelmäßigkeiten • Konflikte • Fehlende Disziplin • Kommunikationsbarrieren • Rollenhemmnisse • Unpassende Kultur • ...

Typische Risikothesen als Ideengeber

<p>Zu hohe Komplexität der Lösung</p> <p>Hat die Domäne eine sehr hohe Komplexität? Gibt es unüberlegte, schnelle Lösungen oder fehlende Abstraktionen? Komplexität gefährdet den Überblick bzw. Wartbarkeit, Korrektheit, Sicherheit ...</p> <p>Softwarelösung 1.1</p>	<p>Unpassende Lösungs-Strukturierung</p> <p>Folgt die Softwarestruktur der Domäne? Sind andere technische oder organisatorische Einflüsse (Canway...) gut aufgegriffen? Falls nicht könnte Wartbarkeit, Zuverlässigkeit etc. leiden.</p> <p>Softwarelösung 1.2</p>	<p>Inadäquates Daten-Handling</p> <p>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?</p> <p>Softwarelösung 1.3</p>	<p>Problematische Konzepte & Technologien</p> <p>Passen die eingesetzten Muster und Konzepte zu den Zielen? Sind sie konsistent angewendet? Sind die verwendeten Technologien und Frameworks passend und etabliert?</p> <p>Softwarelösung 1.4</p>
--	---	--	--

Softwarelösung 1

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

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



Disclaimer

- Die folgenden **Beispiel-Risiken** sind **realistisch** für Dinge, die Team-Mitglieder im Rahmen eines Risiko-Workshops im Pre-Mortem aufwerfen.
- Bezogen auf unser Beispiel **Threema** hier sind sie **rein fiktiv**. Sie dienen der Illustration.



Beispielkarte



Zu hohe Ziele oder Erwartungen

Sind Qualitätsziele (z.B. Performanz) zu ambitioniert? Stehen hohe Einzelziele guter Gesamtqualität im Weg? Werden Randbedingungen verletzt oder unnötige technische Risiken eingegangen?



Zielsetzung



Hypothetisches Risiko (1)



embarc.de

LASR. Ein leichtgewichtiger Ansatz

60



Zu hohe Ziele oder Erwartungen

Sind Qualitätsziele (z.B. Performanz) zu ambitioniert? Stehen hohe Einzelziele guter Gesamtqualität im Weg? Werden Randbedingungen verletzt oder unnötige technische Risiken eingegangen?



Zielsetzungen & Erwartungen 3.1

Aufhebung des Gruppen-Limits

„Wir haben die Begrenzung in Gruppen-Größen aufgehoben. Unsere User sind begeistert von den neuen Möglichkeiten. Später bricht der Chat-Server im Threema-Backend unter der massiven Last und der Menge zu speichernden Nachrichten zusammen.“



Hypothetisches Risiko (2)



embarc.de

LASR. Ein leichtgewichtiger Ansatz

61



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

Das Electron-Framework fällt weg

„Wir waren gezwungen das Electron-Framework in unserer Desktop-App durch eine Alternative / etwas Selbstgebautes zu ersetzen. Das zieht sich. Zwischenzeitlich haben wir den Download des Desktop-Clients von der Seite genommen, weil er auf aktuellen OS nicht mehr funktioniert.“



Hypothetisches Risiko (3)



Einengende Standards oder Randbedingungen

Werden Architekturentscheidungen (oft) durch Standards, Budget- oder Zeitbeschränkungen erschwert? Gibt es einschneidende rechtliche Aspekte rund um Daten oder Internationalisierung?

Organisation & Prozesse 6.4

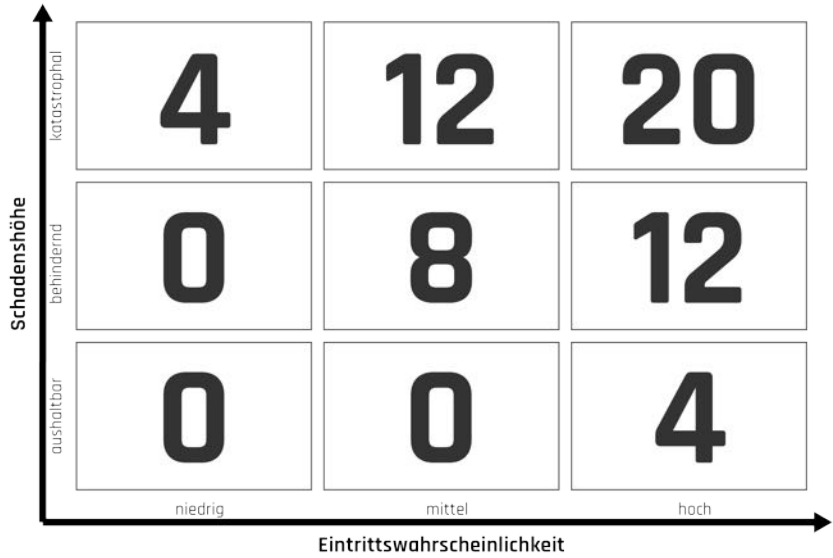
Private Kommunikation scannen

„Die europäische Gesetzgebung hat Messenger-Anbieter verpflichtet, sämtliche private Kommunikation und Dateien zu durchleuchten. Threema muss daraufhin die Lösung für private Nutzer komplett neu denken.“

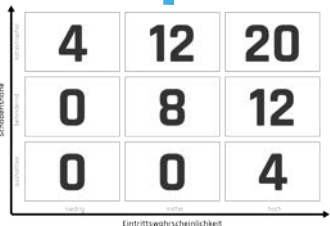
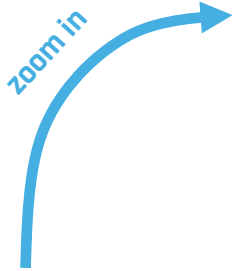




Risiken einordnen



Den betroffenen Zielen zuordnen



Risikobereich mittel/hoch

mittlere Wahrscheinlichkeit dass das Risiko zum Problem wird
 hoher Schaden für zumindest ein Qualitätsziel (katastrophal)

Anordnung im Spielbereich:

■	■	■
■	■	■
■	■	■

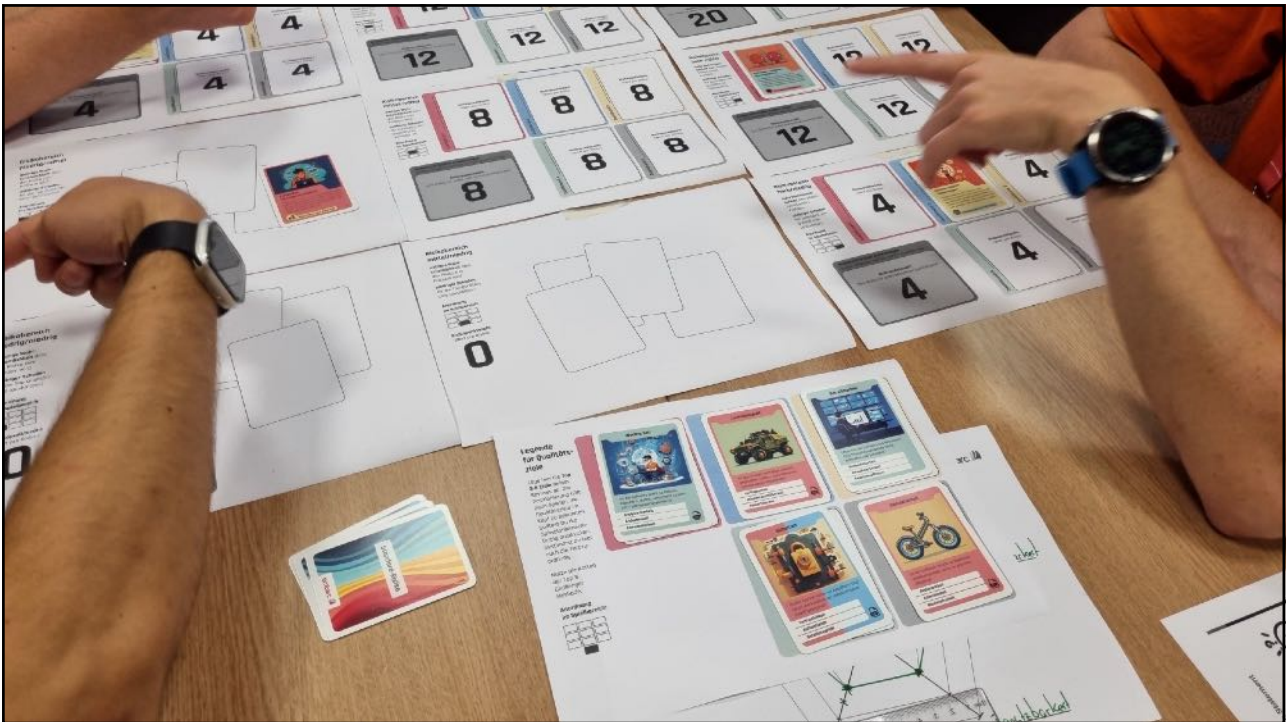
Qualitätsziel 1: Risikopunktzahl: (Wert pro Risiko) 12	Qualitätsziel 2: Risikopunktzahl: (Wert pro Risiko) 12	Qualitätsziel 3: Risikopunktzahl: (Wert pro Risiko) 12
--	--	--

Risiken mit breiten Auswirkungen

Großer negativer Einfluss auf 2-5 Qualitätsziele

Risikopunktzahl: (pro Risiko für jedes betroffene Qualitätsziel) 12
--

Qualitätsziel 4: Risikopunktzahl: (Wert pro Risiko) 12	Qualitätsziel 5: Risikopunktzahl: (Wert pro Risiko) 12
--	--



Durchleuchte die Architektur

Durchleuchte die Architektur

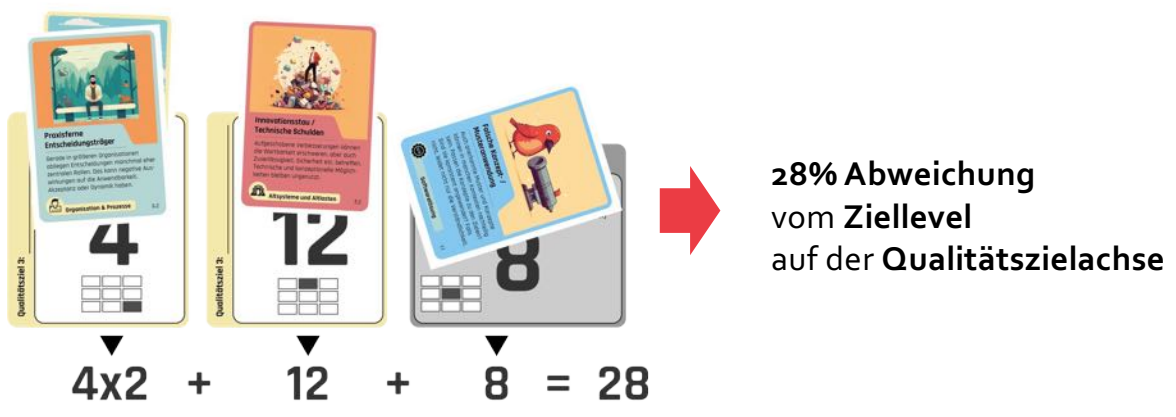


Was ist zu tun?

- 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

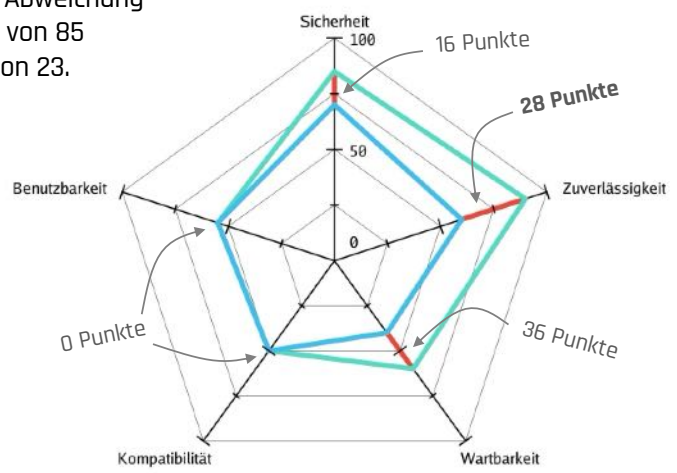


Von Risiken zu „Lücken“



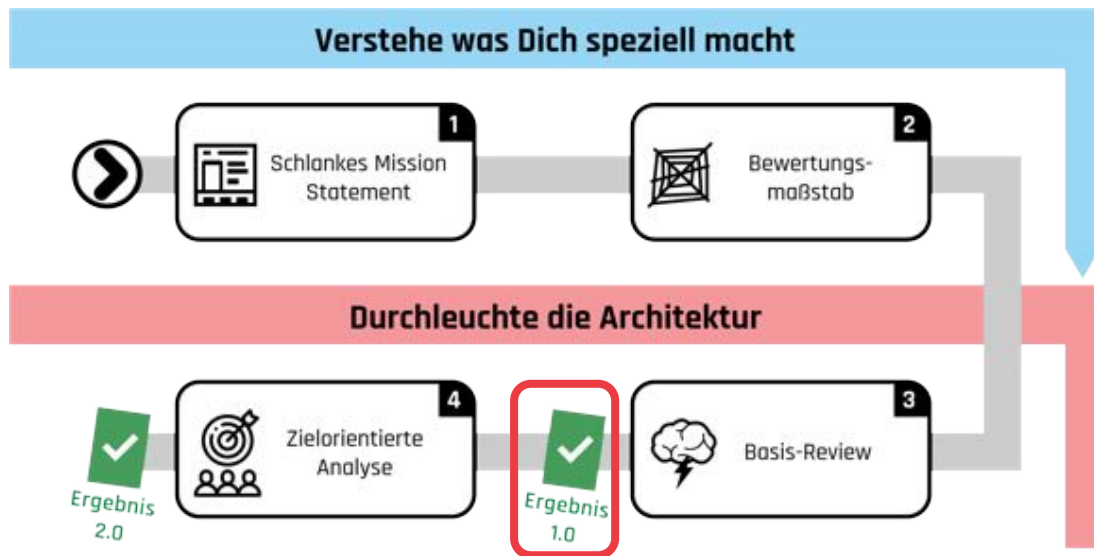
Abweichungen einzeichnen

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



- Ziellinie
- Abweichungen (Lücken)
- Ergebnislinie

Basis-Review, 1. Ergebnis nach Schritt 3



embarc.de

LASR. Ein leichtgewichtiger Ansatz

70

Ergebnis 1.0

LASR Ergebnis 1.0

Symbolhaft dargestellte Elemente des Ergebnisses von LASR nach Schritt 3 (Basis-Review)

Mission Statement
 [Ein kurzer Text (oder Slogan) zur gebauten Lösung, der die wichtigsten Merkmale und Eigenschaften der Software vermittelt und zentrale qualitative Anforderungen anklagen lässt - ggf. als Set von „Claims“ aus einem Brainstorming.]

LASR Ergebnisdigramm (Spinnennetzgrafik)

Lücken und Risiken

- [Kurze Lücken-Beschreibung + ausgewählte Details]
 - Risiko (2C)
 - Risiko (3B)
 - Risiko (2B)
 - Risiko (1C)
- [Kurze Lücken-Beschreibung + ausgewählte Details]
- [Kurze Lücken-Beschreibung + ausgewählte Details]

Stärken und Schwächen

Ansatz (+)	[kurze Beschreibung]	Ansatz (-)	[kurze Beschreibung]



embarc.de

LASR. Ein leichtgewichtiger Ansatz

71



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

04.

Erhöhe die Konfidenz (bei Bedarf)

01. Warum leichtgewichtig bewerten?

02. Verstehe, was Dich speziell macht

03. Durchleuchte die Architektur

04. Erhöhe die Konfidenz (bei Bedarf)

05. Weitere Informationen



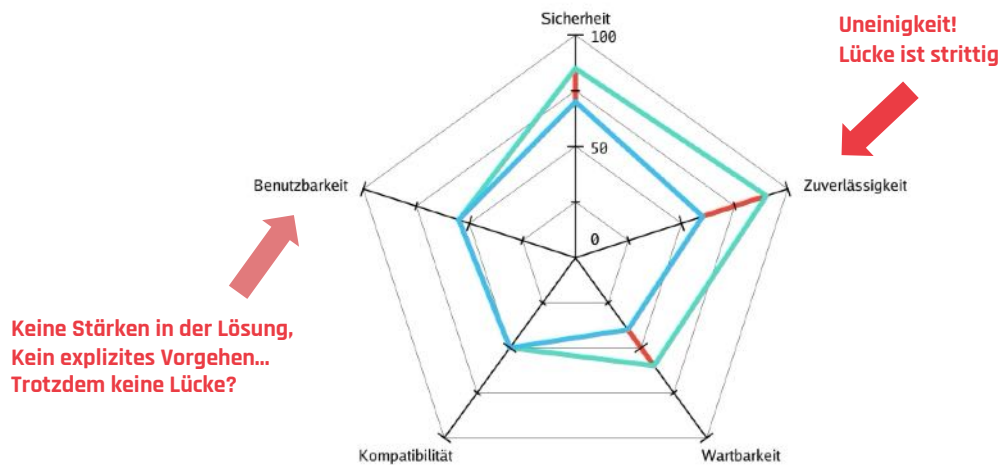


Fokus: Zielorientierte Analyse (Beispiele)

embarc.de

LASR. Ein leichtgewichtiger Ansatz

74

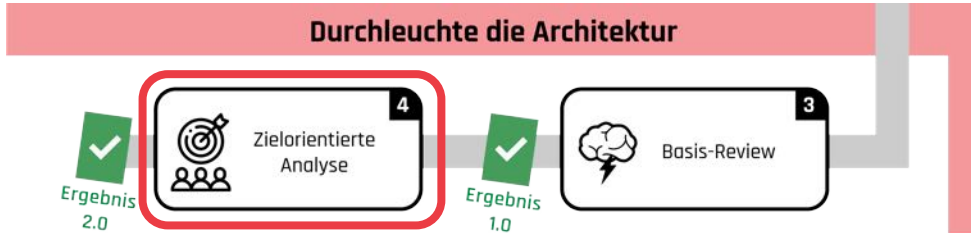


Durchleuchte die Architektur

embarc.de

LASR. Ein leichtgewichtiger Ansatz

75



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**

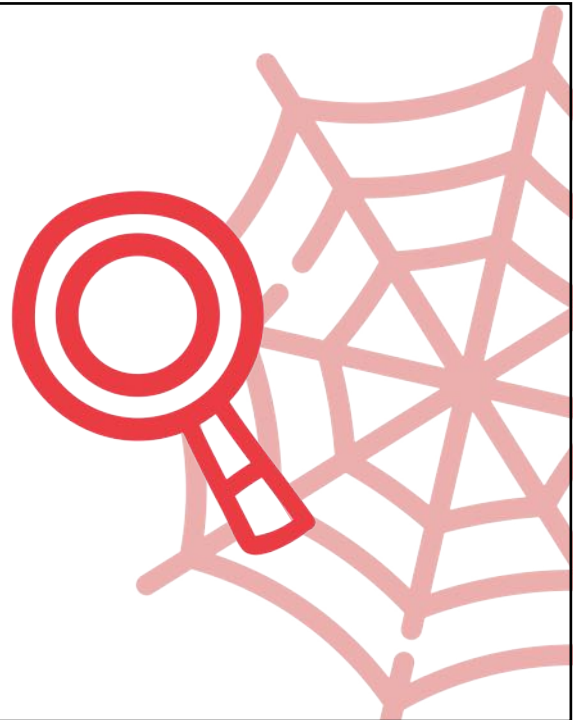
Zielorientierte Analyse

Anhand wichtiger **Qualitätsaussagen:**

- Architekturansätze finden
- Kompromisse aufdecken
- Weitere Risiken finden

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

- „Code-Stöbern“
- Vorhandene Messungen nutzen



Template

Vorlage, um die Ergebnisse der Analyse zu sammeln, zu verdichten und TODOS abzuleiten.

Zielachse Zuverlässigkeit	Analysierte Qualitätsaussagen Welche Qualitätsaussagen wurden zur Zielachse besprochen? Kernszenario: Ausfall auf GP-Site -> Backup trotzdem probierweise
Architekturansätze Komponenten, Technologien, Muster und Konzepte, die mit dem betrachteten Qualitätsziel zu tun haben (sowie eine Einschätzung ob der Zusammenhang positiv oder negativ ist). <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid gray; padding: 5px; font-size: small;"> Basisge-richtete Kommunikation (Sipcc, NetOp-Fähig) Versorg. der V-VR auch abgefragt - Daten gehen nicht verloren </div> <div style="border: 1px solid gray; padding: 5px; font-size: small;"> Produktion ohne externe Verbindung möglich (Anforderung bei Appl. umf.) Verarbeitungskomponente kann über eine Schnitt überbrücken </div> <div style="border: 1px solid gray; padding: 5px; font-size: small;"> Noch keine Erhebung zur Verarbeitungsproblemen </div> <div style="border: 1px solid gray; padding: 5px; font-size: small;"> Datenintensive Verbindung über VPN Fehlererkennung über Video kostet viel Bandbreite </div> </div>	
Kompromisse Seiteneffekte und Lösungsaspekte die sich widersprüchlich auf unterschiedliche Qualitätsziele auswirken. <div style="border: 1px solid gray; padding: 5px; font-size: small; margin-top: 10px;"> Operativ: Auslastung und Op. auf vs. Sicherheitsanforderungen </div>	Risiken / Probleme Technische Herausforderungen die bei der Umsetzung auftreten können. <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid gray; padding: 5px; font-size: small;"> Erkennung von Ausfall und Ursache via VPN </div> <div style="border: 1px solid gray; padding: 5px; font-size: small;"> Verbindungsbruch bei Problem Momentan nicht lösbar </div> <div style="border: 1px solid gray; padding: 5px; font-size: small;"> Risiko zu Netzwerkeinstellung umkehrbar </div> </div> <div style="border: 1px solid gray; padding: 5px; font-size: small; margin-top: 5px;"> Alternativen wären bei Netzwerkausfall: Rückgängige Probleme? </div>
Todos Aus Risiken, offenen Punkten oder negativ bewerteten Architekturansätzen abgeleitete Tätigkeiten <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid gray; padding: 5px; font-size: small;"> Finden eines guten Service Providers (geringe Latenz, Failback) </div> <div style="border: 1px solid gray; padding: 5px; font-size: small;"> Notfallszenario definieren, 4G, 5G, Radio, ... </div> </div>	

Beispiel ➔



Ergebnis 2.0

LASR Ergebnis 2.0
Symbolhaft dargestellte Elemente des Ergebnisses von LASR nach Schritt 4 (Zielorientierte Analyse)

Mission Statement (unverändert)

Zielachsen Detaillierung **NEU**

Zielachse	[Qualitätsaussage]			
Zielachse				
Zielachse				
Zielachse				

LASR Ergebnisdiagramm (Spinnennetzgrafik) **UPDATED**

Lücken, Risiken und Kompromisse **UPDATED**

- 1 [Kurze Lücken-Beschreibung + ausgewählte Details]
- 2 [Kurze Lücken-Beschreibung + ausgewählte Details]
- 3 [Kurze Lücken-Beschreibung + ausgewählte Details]
- 4 [Kurze Beschreibung der Übererfüllung]

Risiko (ZC), **Risiko (ZB)**, **Kompromiss**

Stärken und Schwächen **UPDATED**

Ansatz (+)	[kurze Beschreibung]

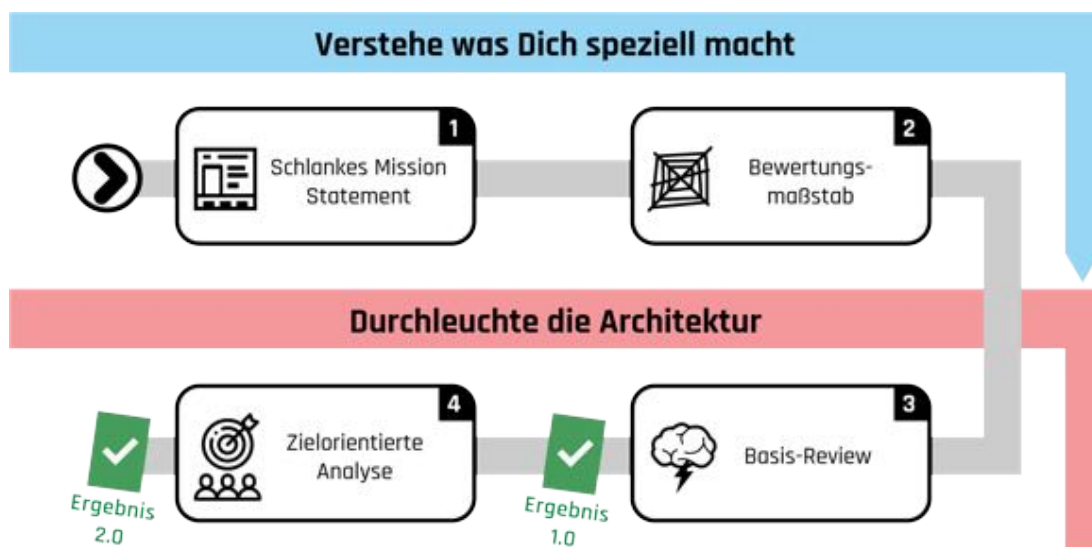
Ansatz (-)	[kurze Beschreibung]

embarc.de

LASR. Ein leichtgewichtiger Ansatz

78

Tätigkeiten und Schritte im Kern-Review



embarc.de

LASR. Ein leichtgewichtiger Ansatz

79

Typische Unsicherheiten im Anschluss

Hängt da noch mehr dran?

Gefundene **Risiken** sind in ihrer Natur oder Auswirkung **zu wenig verstanden**, um direkt hilfreiche Aktivitäten daraus abzuleiten.

In der Theorie OK, aber was ist mit ...?

Rahmenbedingungen (z.B. Standards) oder die eigene **Organisation** wurde als Risikofaktor **zu wenig betrachtet**.

Wo ist der Code?

Die **Architekturebene** wird als **zu abstrakt** für die tatsächlichen Probleme des Vorhabens wahrgenommen.

Wo fangen wir an?

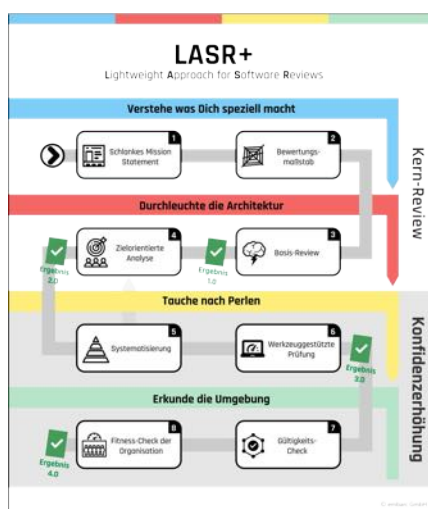
Die **Priorisierung** der identifizierten Probleme ist **schwierig**, ggf. fehlt es bei kleinteiligen Ergebnissen an Überblick oder Einordnung.

Ist nicht auch XY wichtig?

Der **Fokus** auf max. 5 Qualitätsziele wirkt **problematisch**.



Ausblick: Nach dem Kern-Review ...



LASR gliedert sich grob in zwei Modi, von denen der zweite optional ist („LASR+“):

I. Kern-Review

Liefert rasch ein erstes Review-Ergebnis, das im vierten Schritt bereits geschärft wird.

II. Konfidenzerhöhung (optional)

Bietet bei Unsicherheiten mit dem Ergebnis mehr Tiefe und schließt Code-Ebene und Organisationsseite mit ein.



05.

Weitere Informationen

01. Warum leichtgewichtig bewerten?

02. Verstehe, was Dich speziell macht

03. Durchleuchte die Architektur

04. Erhöhe die Konfidenz (bei Bedarf)

05. Weitere Informationen



TL;DR - Too long; didn't read.

Reviews decken Schwächen von Softwarelösungen auf und **sichern** technische und architektonische Ideen **ab**. **Konventionelle** Bewertungsmethoden wie ATAM sind **mitunter** zu **schwer**gewichtig.

Ein **leichtgewichtiges Review** zeichnet sich durch geringere Laufzeit und Aufwände, weniger Beteiligte und **raschere Ergebnisse** aus.

LASR skizziert in seinem Kern-Review erprobte Schritte für eben solch ein Review. Ihr könnt es **direkt in Eurem Team** in 2- 3 h „durchspielen“. Bei Bedarf bietet LASR Vertiefungspunkte zur **Konfidenzerhöhung** an.

Artikel zum Einstieg in LASR (online)

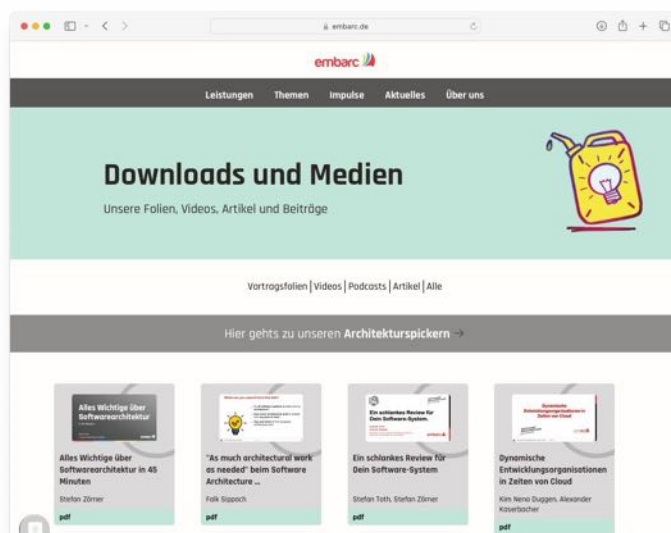


Stefan Toth & Stefan Zörner:
„Leichtgewichtige Software-
Reviews mit LASR“
Online, Informatik Aktuell

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



Folien als PDF zum Download



➔ embarc.de/download/





Buchtipp zum Thema

embarc.de

LASR. Ein leichtgewichtiger Ansatz

86



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/



Rabatt-Code für Leanpub

embarc.de

LASR. Ein leichtgewichtiger Ansatz

87



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

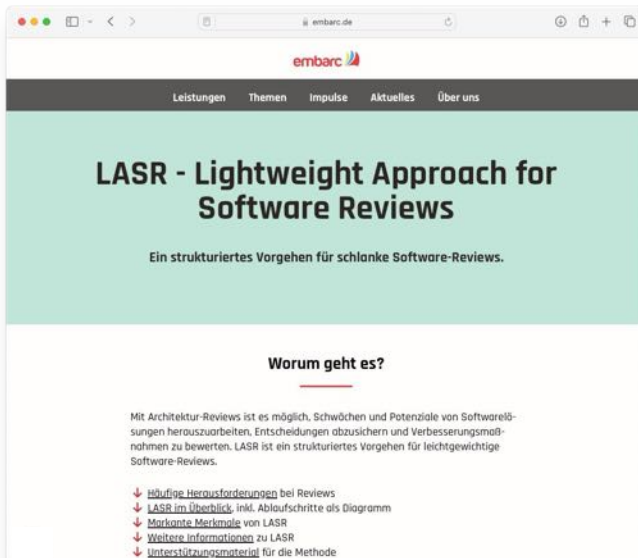
Für alle Teilnehmenden der
Berlin Expert Days, reduziert
den Preis um 45%

Coupon: JSD2024



→ leanpub.com/software-systeme-reviewen/c/JSD2024

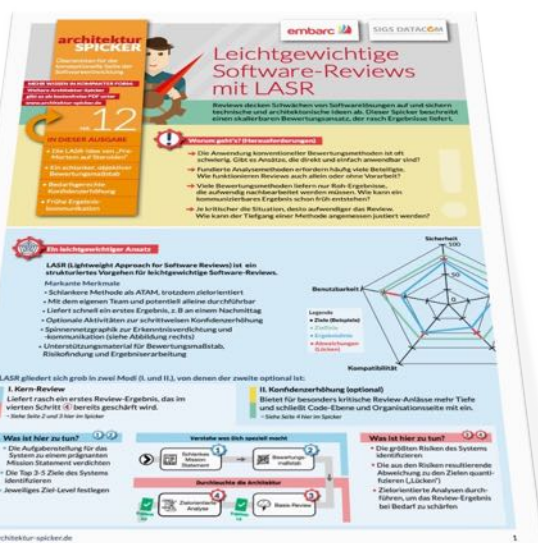
LASR-Themenseite bei embarc



➔ embarc.de/lasr-reviews/

embarc.de
LASR - Ein leichtgewichtiger Ansatz
88

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.

➔ architektur-spicker.de

embarc.de
LASR - Ein leichtgewichtiger Ansatz
89

Architektur-Punsch 

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.**

embarc.de

Ich freue mich auf Eure Fragen!

 Stefan.Zoerner@embarc.de

 [linkedin.com/in/stefan-zoerner](https://www.linkedin.com/in/stefan-zoerner)

 [@StefanZoerner@mastodon.social](https://mastodon.social/@StefanZoerner)





LASR. Ein leichtgewichtiger Ansatz

91