

# architektur SPICKER

Übersichten für die  
konzeptionelle Seite der  
Softwareentwicklung

MEHR WISSEN IN KOMPAKTER FORM:

Weitere Architektur-Spicker

gibt es als kostenfreies PDF unter

[www.architektur-spicker.de](http://www.architektur-spicker.de)

4  
NR.

## IN DIESER AUSGABE

- Was leisten Architektur-Reviews?
- Welche Methoden und Werkzeuge helfen?
- Wer sollte wann und wie oft in Reviews eingebunden sein?

# Architektur- Reviews

Architektur-Reviews schaffen Transparenz in der Softwareentwicklung, identifizieren Risiken und decken Kompromisse auf. Dieser Spicker bietet einen fundierten Überblick.



## Worum geht's? (Herausforderungen)

- ➔ Eine Neuentwicklung steht an und erste Lösungsansätze stehen im Raum. Sind Sie und Ihr Team auf dem richtigen Weg?
- ➔ Das Management hat das Vertrauen in Ihre Lösung verloren. Wie gewinnen Sie es zurück und strahlen Sicherheit aus?
- ➔ Unterschiedliche Stakeholder verfolgen widersprüchliche Ziele mit Ihrer Software. Wie konkretisieren und priorisieren Sie deren Wünsche?
- ➔ Größere Umbaumaßnahmen stehen an. Wie wählen Sie passende Lösungsansätze nachvollziehbar aus?
- ➔ Sie möchten Ihre Architekturarbeit um regelmäßige Reflektion ergänzen. Wie gehen Sie vor?



## Aspekte von Architektur-Reviews

In Architektur-Reviews können unterschiedliche Aspekte der Architektur unter die Lupe genommen werden. Isoliert oder auch gemeinsam.

### Angemessenheit der Architektur prüfen (AA)

- Gibt es ein gemeinsames Ziel- und Architekturverständnis?
- Sind die architektonischen Konzepte für die Zielerreichung geeignet?
- Sind Architekturentscheidungen mit Risiken behaftet?
- Welche Kompromisse werden durch die Architektur ausgeprägt?

### Qualität der Architektur prüfen (QA)

- Folgt die Architektur gängigen Best-Practices?
- Ist die Architektur verständlich, nachvollziehbar und einfach?
- Ist konzeptionelle Integrität gegeben?
- Ist die Architektur gut kommuniziert und etabliert?

### Umsetzungsabgleich durchführen (UA)

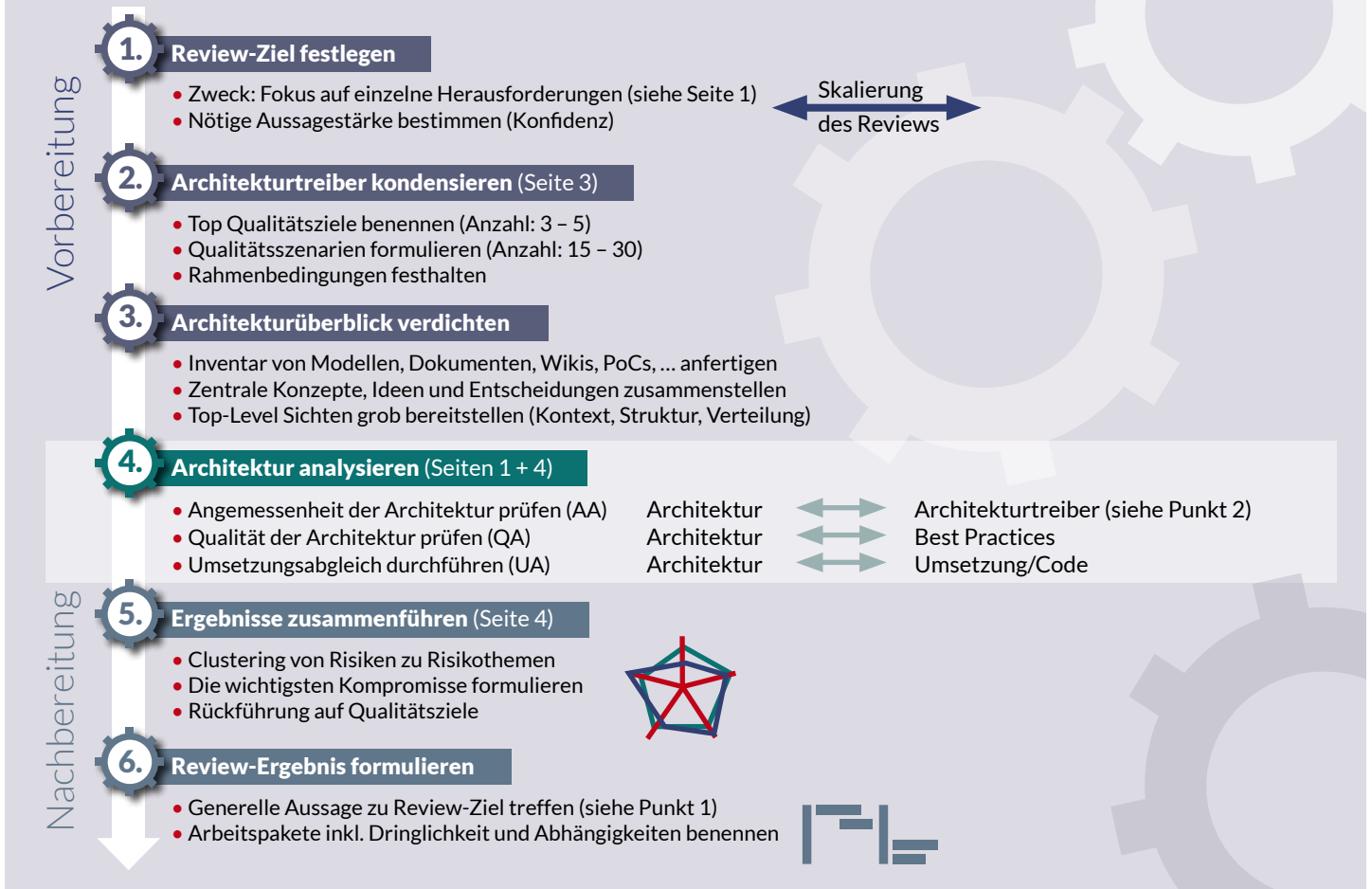
- Sind Architekturentscheidungen auf Code-Ebene nachvollziehbar?
- Gibt es Ziel-gefährdende Auffälligkeiten in der Umsetzung?
- Entsprechen die Code-Strukturen den architektonisch vermuteten?

Szenariobasiert  
(Inhaltliche Architektur-  
prüfung gegen Treiber)

Ad-hoc möglich  
(Vorbereitung kann  
sehr dünn sein)

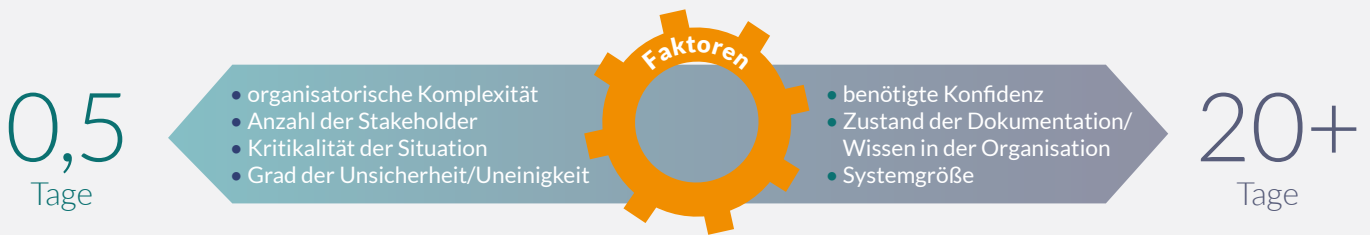
Toolgestützt  
(siehe Spicker #2 -  
Quantitative Analyse)

## Typischer Ablauf eines Reviews



## Skalierung von Reviews

Wie groß Reviews angelegt werden sollten hängt von vielen Faktoren ab. Im Folgenden einige wichtige:



### ATAM - Architecture Tradeoff Analysis Method

Die ATAM ist eine theoretische Basis an der sich viele Architektur-Reviews orientieren wenn es um die Angemessenheit der Architektur geht (AA). Die Architekturbewertungsmethode wurde am Software Engineering Institute (SEI) der Carnegie-Mellon Universität entwickelt und im Jahr 2000 erstmals veröffentlicht. In zwei Kernphasen-Workshops wird die Architektur detailliert und analysiert. Wegen ihres Umfangs wird die Methode für kleinere Reviews meist nur ausschnittsweise angewendet (siehe Skalierung).

#### Ablauf

#### Phase 1

1. ATAM präsentieren
2. Geschäftstreiber präsentieren
3. Architektur präsentieren
4. Architekturansätze identifizieren
5. Utility Tree erstellen
6. Architektur analysieren

#### Pause

(Tage bis Wochen, je nach Erkenntnissen von Phase 1)

#### Phase 2

7. Szenarien brainstormen
8. Architektur analysieren
9. Ergebnisse präsentieren

*breitere Stakeholder-Beteiligung als in Phase 1, auch Benutzer, Betrieb, Auftraggeber, Wartungsvertreter, ...*

#### Fokus:

Vom Architektur-Review-Ablauf (s. o.) werden in den ATAM-Kernphasen vor allem **2.**, **3.** und **4.** unterstützt.

#### Skalierung:

mittel (2 Workshoptage exkl. Vorbereitung) bis voll (10-20 Analyse-Tage)

# Rollen in Reviews

Rolle	Mögliche Besetzung	Aufgaben	Relevant in Analyse-Art
Reviewer	(Teamfremde) Entwickler, (Projekt-fremde) Architekten, Linien-Architekten, externe Experten	Review-Fokus definieren, Fragen stellen, Entscheidungen mit Treibern verbinden, Risiken identifizieren, Eigene Erfahrung einbringen	AA, QA, UA
Architekturvertreter	Entwickler, Senior-Entwickler, Architekten, Tester	Lösungen und Entscheidungen vorstellen, Probleme erläutern, Kompromisse verdeutlichen	AA, QA, UA
Stakeholder	Product Owner, Kunde, Fachabteilung, Projektmanager, Unternehmensarchitekt, Betrieb	Einflüsse und Rahmenbedingungen vertreten, Kompromisse bewerten, Relevanz verdeutlichen	AA, (UA)
Moderator	Projektfremder Mitarbeiter, externer Review-Leiter	Organisation, Vorbereitung, Moderation...	AA (Workshopteile)

## 2. Architekturtreiber

### Qualitätsszenarien

Qualitätsszenarien konkretisieren qualitative Anforderungen. Ähnlich wie User Stories haben auch Qualitätsszenarien einen typischen Aufbau:



### Arten von Szenarien

Um Qualitätsmerkmale zu konkretisieren stehen Ihnen unterschiedliche Arten von Szenarien zur Verfügung:

#### Verwendungsszenarien

Synonyme: Nutzungsszenarien

**Kennzeichen:** „Normale“ Verwendung des Systems

**Typische Quellen:** Benutzeraktion (wie in den Anforderungen definiert)

**Oft genutzt für:** Benutzbarkeit, Effizienz, Funktionalität

#### Änderungsszenarien

Synonyme: Wachstumsszenarien

**Kennzeichen:** Etwas kommt hinzu oder ändert sich

**Typische Quellen:** Mehr Last, neue Features, technische Migration ...

**Oft genutzt für:** Wartbarkeit, Kapazität, Portierbarkeit, Kompatibilität

#### Stressszenarien

Synonyme: Fehler-, Katastrophenszenarien

**Kennzeichen:** Etwas „Unvorhergesehenes“ passiert

**Typische Quellen:** (Fremd-/Teil-)Systemausfall, Überlast, Fehlbenutzung ...

**Oft genutzt für:** Zuverlässigkeit, Sicherheit

Ein Sachbearbeiter erfasst einen Schaden. Er navigiert mit einem Klick zur richtigen Maske und ist in durchschnittlich 3 Minuten fertig.

Die Anzahl der Benutzer der Webseite wächst in einem Jahr um 100%. Ohne Personenaufwände bleibt das System gleich effizient und robust.

Ein technisches Gebrechen legt Teile der Systemlandschaft lahm (bis 30%). Kunden merken von den Ausfällen nichts.

### Rahmenbedingungen

Neben qualitativen Anforderungen sind Rahmenbedingungen die wichtigsten Treiber von Architekturarbeit. Es ist generell wichtig zu erheben welche Rahmenbedingungen übersehen wurden, verletzt sind oder zukünftig in Gefahr sind verletzt zu werden.

Technische Rahmenbedingungen	Organisatorische Rahmenbedingen	Gesetzliche Vorgaben
<ul style="list-style-type: none"> <li>• Hardware-Vorgaben</li> <li>• Plattform-Vorgaben</li> <li>• Gesetzte Frameworks</li> <li>• Vorgaben des Systembetriebs</li> <li>• Programmiervorgaben</li> <li>• Andere technische Einschränkungen</li> </ul>	<ul style="list-style-type: none"> <li>• Organisationsstruktur</li> <li>• Projektgröße</li> <li>• Vorgehensmodell</li> <li>• Budget</li> <li>• Verfügbarer Entwicklungszeitraum</li> <li>• Organisatorische Standards</li> <li>• Know-How</li> </ul>	<ul style="list-style-type: none"> <li>• Haftungsfragen</li> <li>• Datenschutz</li> <li>• Nachweispflichten</li> <li>• Revisionsicherheit</li> <li>• Regulatorische Vorgaben</li> <li>• Internationale Rechtsfragen</li> </ul>

# In Bewertungsworkshops

Die Bewertung der Angemessenheit der Architektur (AA) wird meist in Workshops durchgeführt. Dabei wird (in Anlehnung an ATAM) Szenario-basiert vorgegangen – Ausgehend von den wichtigsten Treibern werden also Architekturkonzepte besprochen und hinterfragt.

- 1. Generierung von Szenarien**
- Brainstorming mit Treiber-Vertretern
  - Ableitung aus Dokumenten
  - (Einzel-)Interviews mit Stakeholdern

- 2. Priorisierung der Szenarien**
- Punkte kleben (Wahlmechanismus)
  - Einfacher Konsens
  - Bewertung nach Kriterien

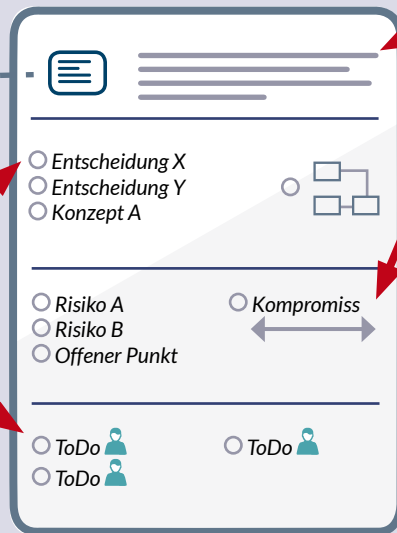
- 3. Durchsprache der Szenarien nach Priorität**

### Architektur vorstellen

- relevante Lösungsdetails
- bekannte Probleme
- Ideen und Lücken

### ToDo's ableiten

- Vor allem aus Risiken und offenen Punkten.
- Verantwortlichen bestimmen.



### Fachliche Wichtigkeit / technische Schwierigkeit

⚡	hoch / hoch Fokus! bei Durchsprache
📁	hoch / niedrig evtl. nutzen um Vertrauen zu stärken
🔍	niedrig / hoch Transparenz! Antwortmaß hinterfragen

### Szenario schärfen

Siehe Schema für Qualitätsszenarien (Seite 3)

### Fragen zum Szenario stellen

- ▶ Sind die Entscheidungen nachvollziehbar und ist Szenario-Zusammenhang klar?  
✓ Verständlichkeit und Fokus
- ▶ Sind alle Aspekte des Szenarios abgedeckt?  
✓ Entscheidungen / offene Punkte / Risiken
- ▶ Welche Architekturansätze behindern hier?  
✓ Kompromisse
- ▶ Werden Qualitätsmerkmale von den genannten Entscheidungen negativ beeinflusst?  
✓ Kompromisse/evtl. Risiken
- ▶ Welche Schwierigkeiten können bei Umsetzung, Auslieferung oder im Betrieb auftreten?  
✓ Risiken/offene Punkte
- ▶ Sind wichtige Rahmenbedingungen verletzt?  
✓ Probleme/Risiken



## Qualitätsmerkmale und Kompromisse

Wirkung auf Qualitätsmerkmal  
positiv oder negativ



Häufiges Ergebnis von Durchsprachen und Analysen sind Kompromisse. Qualitative Eigenschaften von Software sind nicht unabhängig optimierbar. Das folgende Diagramm zeigt wichtige Qualitätsmerkmale angelehnt an ISO 25010 und verbindet sie exemplarisch mit Wechselwirkungen, wie sie auch in Reviews vorkommen können.



Wir freuen uns auf Ihr Feedback: [spicker@embarc.de](mailto:spicker@embarc.de)

<http://architektur-spicker.de>



<http://www.embarc.de>  
[info@embarc.de](mailto:info@embarc.de)



<http://www.sigs-datacom.de>  
[info@sigs-datacom.de](mailto:info@sigs-datacom.de)