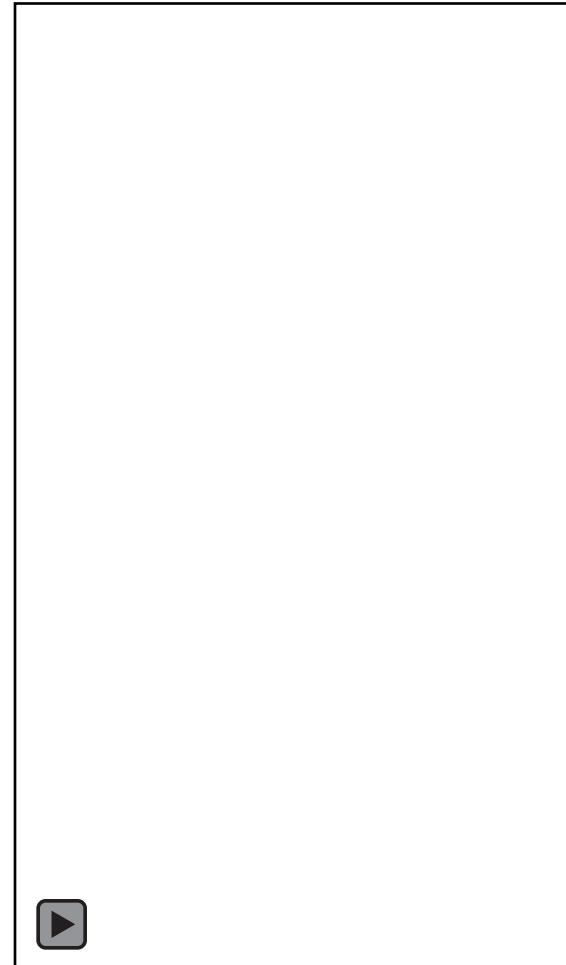


Architektonische Eigenschaften

Bitte sammeln Sie in einem Breakout-Room so viele architektonische Eigenschaften / nicht-funktionale Requirements wie möglich.

Betriebsrelevante Eigenschaften

- Availability
- Performance
- Recoverability
- Reliability
- Scalability
- Robustness



Strukturelle Eigenschaften

- Konfigurierbarkeit
- Erweiterbarkeit
- Benutzbarkeit
- Internationalisierung
- Wartbarkeit
- Portierbarkeit
- Supportability
- Upgradefähigkeit

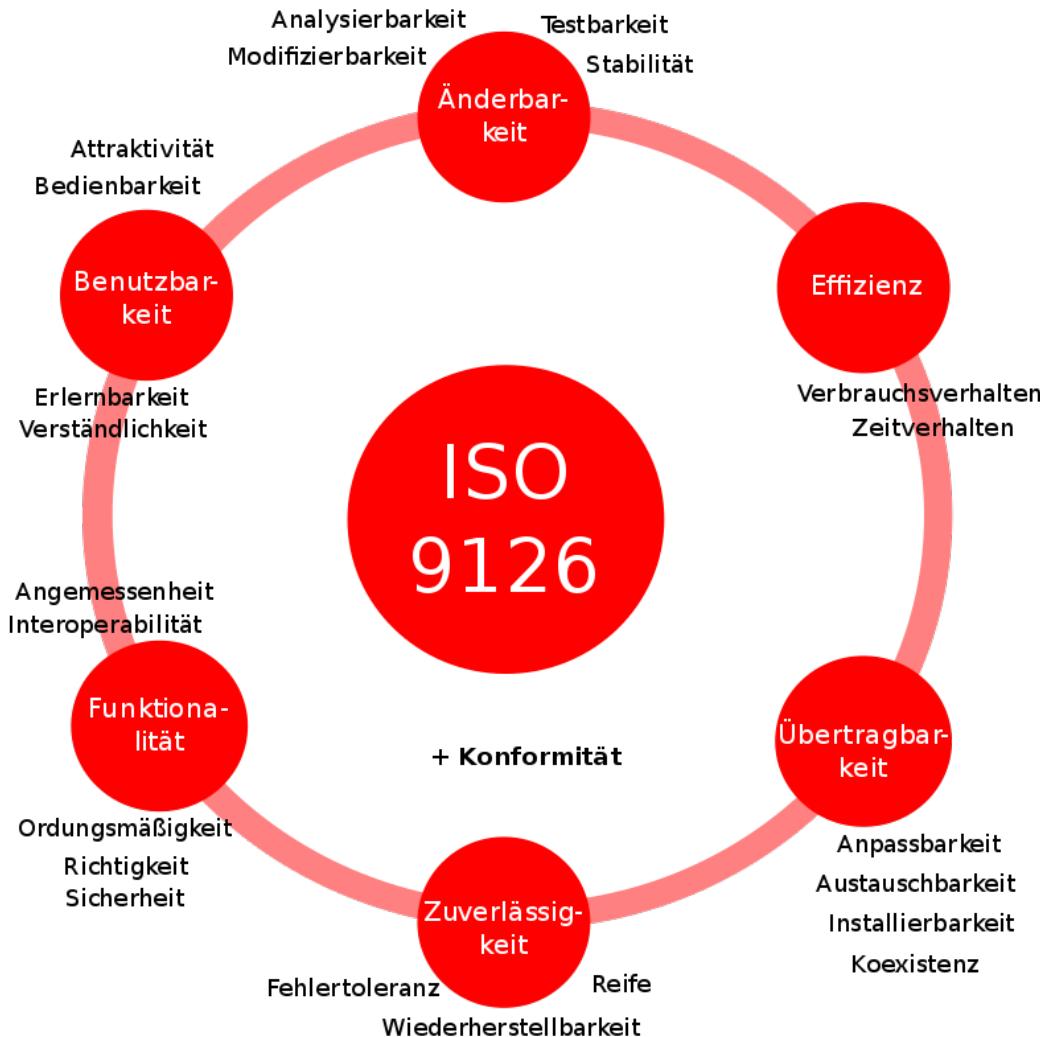


Übergreifende Eigenschaften

- Barrierefreiheit
- Archivierbarkeit
- Authentisierung
- Autorisierung
- Integrity
- Vertraulichkeit
- Legal Compliance



ISO 9126



Quelle: [https://commons.wikimedia.org/wiki/File:ISO_9126_quality_\(de\).svg](https://commons.wikimedia.org/wiki/File:ISO_9126_quality_(de).svg)

Beispiel

R48: Das System muss 100% sicher sein.

R49: Die Antwortzeiten liegen immer unter 10ms.

Hintergrund: Das System soll zentral in London (Load-Balancer, Web-Server, Business-Logik, Datenbank) installiert sein. Benutzer sitzen in ganz Europa.

Gibt es ein Problem?

Beispiel

R49: Die Antwortzeiten liegen immer unter 30ms.

R50: Die Daten müssen immer aktuell sein.

Hintergrund: Das System hat eine zentrale Datenbank in London.

Benutzer sitzen in New York, London, Singapur. Jeder Benutzer kann Daten ändern, die potenziell jeden anderen Benutzer betreffen.

Gibt es ein Problem?

Qualitätsszenarien

Ein Qualitätsszenario

- ist vergleichbar zu einem Use Case
- beschreibt, wie ein System auf einen Stimulus reagieren soll.

Ein Qualitätsszenario besteht aus:

- **Quelle** – Wer hat den Stimulus erzeugt?
- **Stimulus** – Eine Bedingung, die das System beeinflusst
- **Artefakt** – Der Teil des Systems, der stimuliert wird
- **Umgebung** – in der der Stimulus auftritt
- **Antwort** – Eine Aktivität als Reaktion auf den Stimulus
- **Maß** – zur Evaluation der Antwort

Beispiel (P. Clements)

“During peak operation, an unauthorized intruder tries to download prohibited data via the system administrator’s interface. The system detects the attempt, blocks access, and notifies authorities within 15 seconds.”

Quelle – *Wer hat den Stimulus erzeugt?* ?

Stimulus – *Eine Bedingung, die das System beeinflusst* ?

Artefakt – *Der Teil des Systems, der stimuliert wird* ?

Umgebung – *in der der Stimulus auftritt* ?

Antwort – *Eine Aktivität als Reaktion auf den Stimulus* ?

Maß – *zur Evaluation der Antwort* ?

Bestimmen Sie die 6 Punkte dieses Qualitätsszenarios!

Quelle: Clements, P.: Best Practices in Software Architecture, 2006, CMU SEI

Beispiel (P. Clement)

During peak operation, an unauthorized intruder tries to download prohibited data via the system administrator's interface. The system detects the attempt, blocks access, and notifies authorities within 15 seconds.

Quelle	<i>an unauthorized intruder</i>
Stimulus	<i>tries to download prohibited data</i>
Artefakt	<i>system administrator's interface</i>
Umgebung	<i>during peak operation</i>
Antwort	<i>the attempt is detected, blocked, reported</i>
Maß	<i>within 15 seconds</i>

Wer sagt dem Architekten welche Architektur-Eigenschaften wichtig sind?

Stakeholder sind Personen oder Gruppen oder Organisationen, die impacted, interested oder involved in ein Projekt sind.

Übung:

Angenommen, die THM will ein neues System „Facebook für Vorlesungen“ einführen. Zu jeder Vorlesung gibt es Facebook-Seiten mit neuen Postings (z.B. neues Video, neue Übungsaufgaben, ...). Jeder THM-Mitarbeiter, alle Studierenden können subscriben, liken, kommentieren, weiterleiten, ...

- a) Wer sind die Stakeholder?
- b) Wer ist impacted, interested, involved?

Dozenten (Impacted), Studierende,
ITS, Datenschutz, Entwickler, Präsident, Facebook,
Personalrat, Recht, ZeKoll, Studiensekr., Kommunikation, Dekane,



.... wir haben ein Problem!

Dozenten



Einfachere
Pflege der
Daten!

Präsidium



Billiger!

Studierende



Bessere
Performance!

ITS



Verfügbarkeit
Sicherheit
...

Kommunikation



Sexy Look and
Feel!

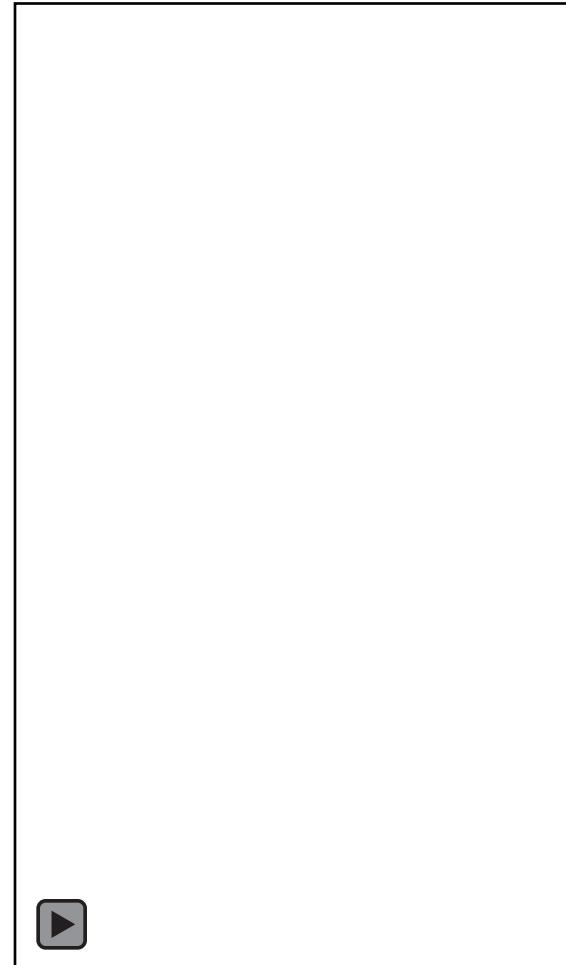


Oh, my ...

Architekt

QAW

- SEI schlägt einen **Quality Attribute Workshop** vor:
 - früh im Projekt – vor der Definition der Softwarearchitektur
 - alle Stakeholder dabei
 - geleitet vom Architekten
 - Ergebnis: Priorisierte Qualitätsszenarien
- Vorteile?
- Nachteile?



Trade-Offs

- Nicht immer sind alle Anforderungen gleichzeitig erfüllbar.
- Der Architekt muss daher abwägen, welche Anforderungen höher zu gewichten sind und welche Anforderungen abgeschwächt werden können.
- Im Beispiel würden die Antwortzeiten bei Verzicht auf Verschlüsselung sicher besser, aber ...

Die richtige Architektur ist die mit den wenigsten Nachteilen.

-> ATAM (Architecture Tradeoff Analysis Method)

Übung / Hausaufgabe

Ordnen Sie die folgenden Requirements den architektonischen Eigenschaften zu:

R51: Das System muss in der Lage sein, die Systeme von Firmen, die wir kaufen oder mit denen wir uns zusammenschließen werden, zu integrieren.

R52: Das System muss für Benutzer in den Filialen einfach zu bedienen sein.

R53: Das System muss Daten verschiedener rechtlich getrennter Tochterfirmen verwalten.