ITECH PROGRESS

for your business success

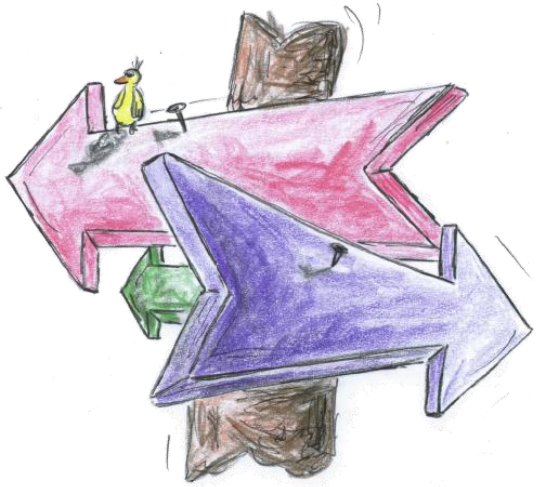
iSAQB

Bewertung von Software-Architekturen

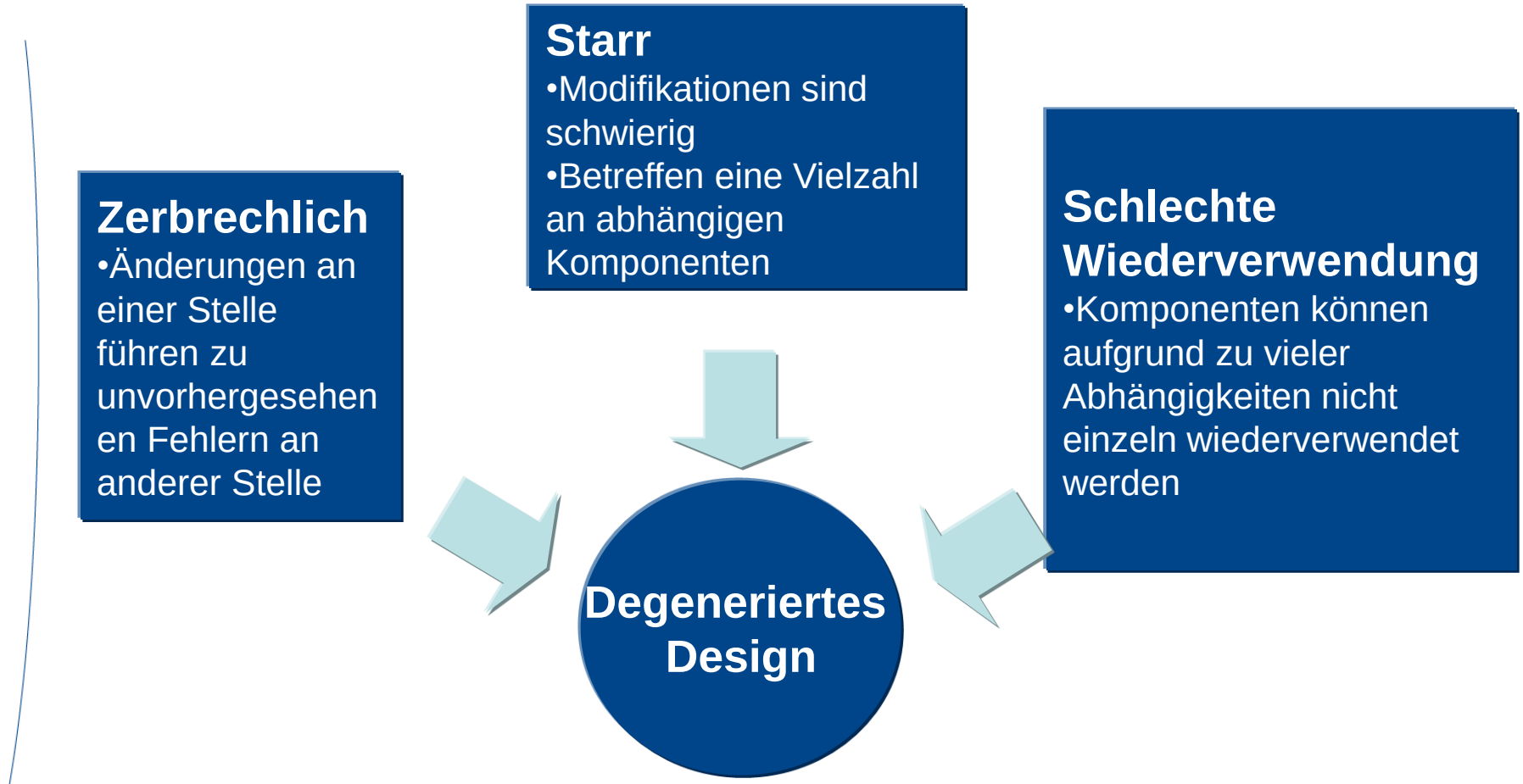
Dipl.-Ing. Mahbouba Gharbi
@email: m.gharbi@itech-progress.com

Agenda

- Motivation
- Bewertung von Software-Architekturen
  - Qualitative Bewertung
  - Quantitative Bewertung
- Wie tief, wie gut, wie sinnvoll?

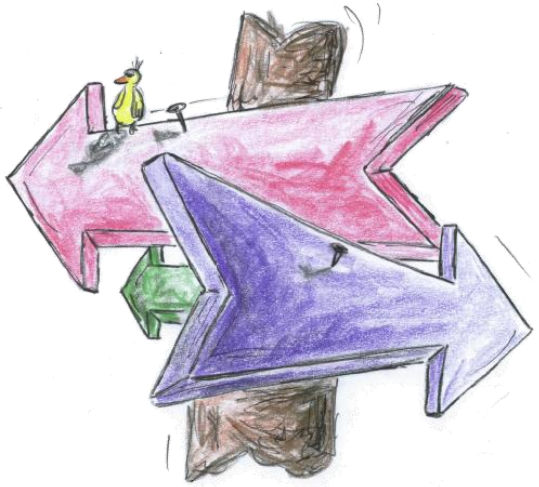


Symptome von degeneriertem Design



Agenda

- Motivation
- Bewertung von Software-Architekturen
  - Qualitative Bewertung
  - Quantitative Bewertung
- Wie tief, wie gut, wie sinnvoll?



Warum wird eine Architektur bewertet?

Wissen über das zu entwickelnde System

- Machbarkeit
- Kosten
- Qualität

> You cannot control what you cannot measure
>
> *Tom DeMarco*

Bewertung in Software-Projekten

Bewertung von Prozessen:
- Entwicklungs- oder Betriebsprozesse
- organisatorische Aspekte
- Einsatz von Ressourcen
- kaum Aussagen über Qualität

Bewertung

Bewertung von Artefakten:
- Anforderungen
- Entwürfe
- Quellcode
- Dokumente

Bewertung von Software-Architekturen

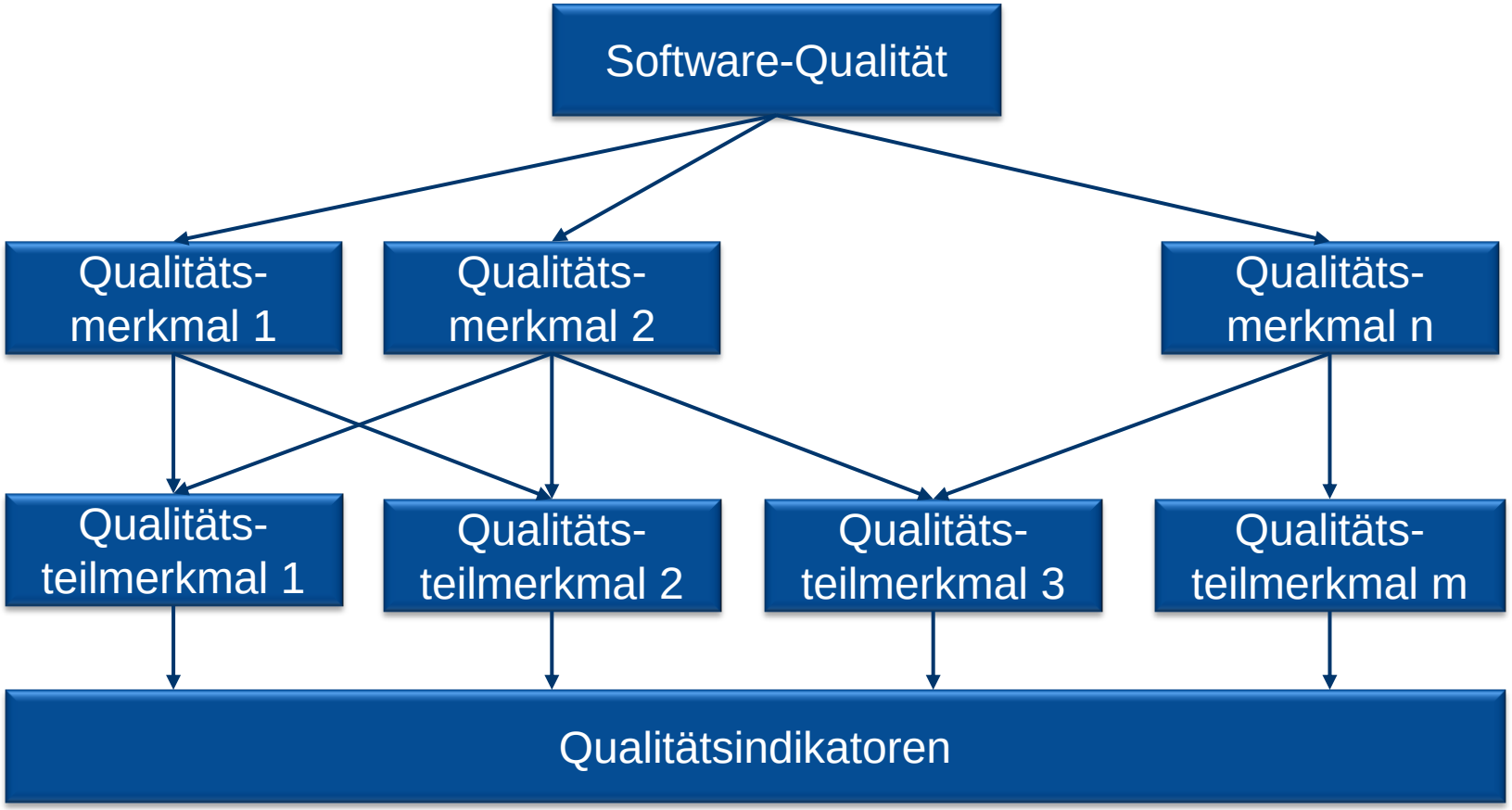
Qualitativ

- Bewertung nach Beschaffenheit oder Güte
- Bewertung von Qualitätsmerkmalen
- Hilft Risiken zu identifizieren
- Bewertung sollte regelmäßig und so früh wie möglich stattfinden

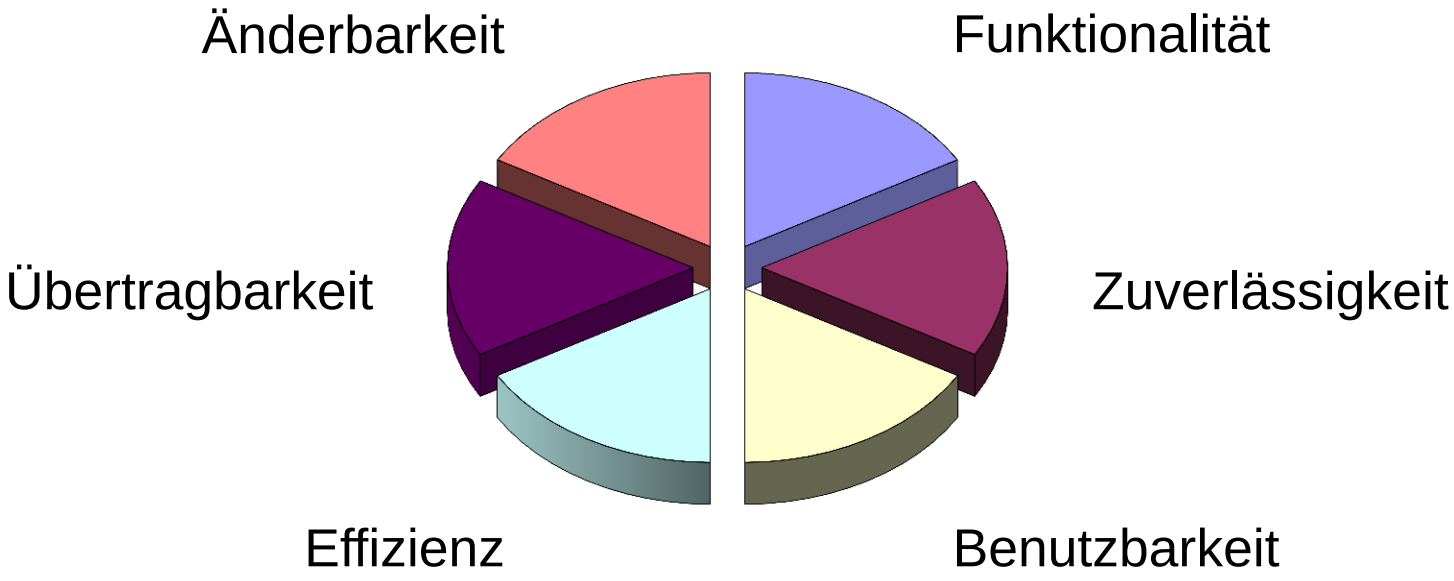
Quantitativ

- Bewertung der Artefakte in Zahlen
- Geben gute Hinweise für strukturelle Veränderungen
- Keine Aussage über Funktionsfähigkeit und Qualität zur Laufzeit
- Benötigen fachlichen und technischen Kontext um vergleichbar zu sein
  - Tipp: Daten sammeln aus Vergleichsprojekten

Methoden zur Bewertung – qualitativ



Qualitätsmerkmale nach ISO/IEC 9126 (ab 2005 25000)



Qualitative Bewertung von Software-Architekturen

- ATAM
  - Architecture Tradeoff Analysis Method
  - Methodisches Vorgehen zur Architekturbewertung
- Ziel der Architekturbewertung
  - Definition der von den maßgeblichen Stakeholdern geforderten Qualitätsmerkmale in möglichst konkreter Form
- Hilfsmittel
  - Szenarien
  - Qualitätsbäume
  - Liste möglicher Risiken der Architektur

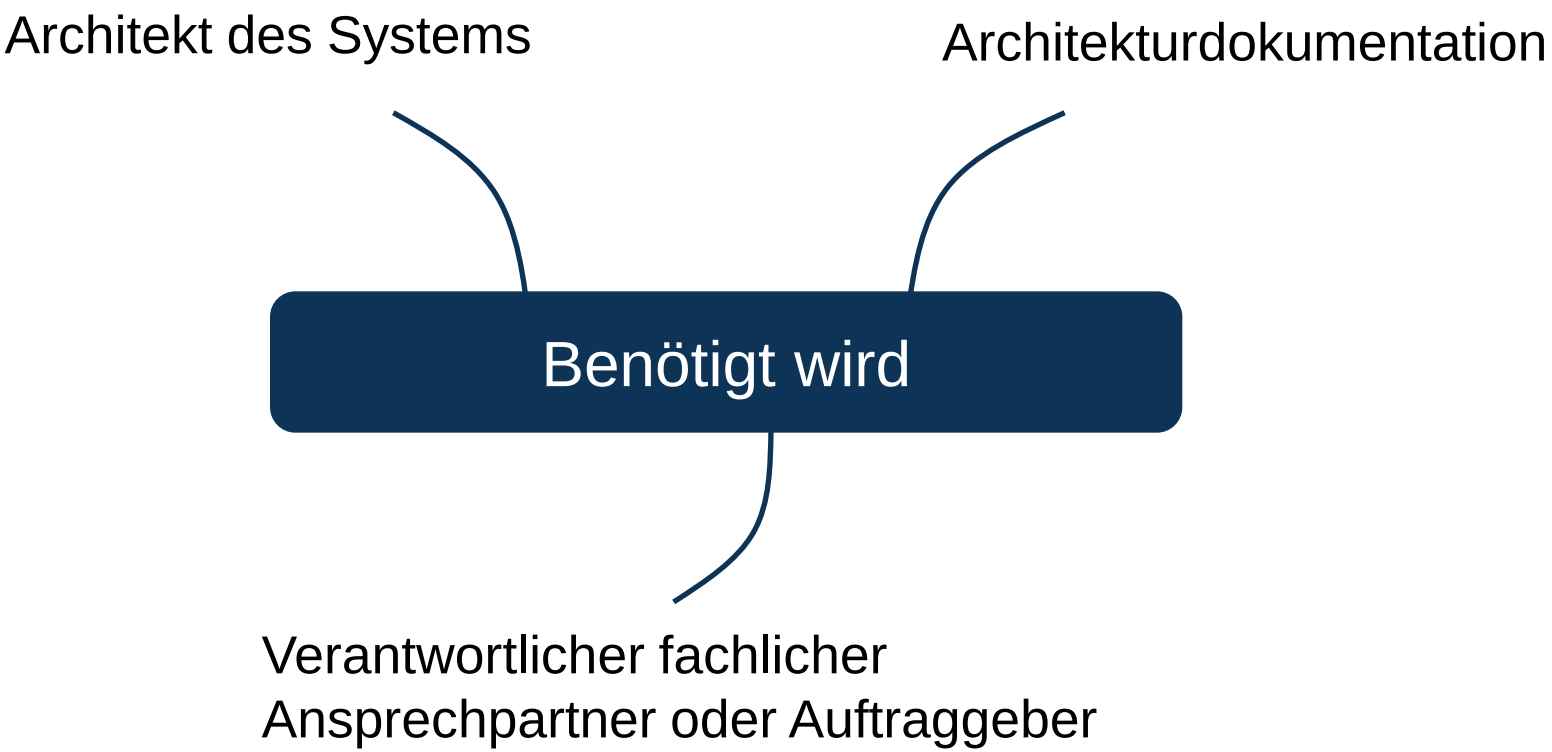
ATAM

- Entwickelt am ‚Software Engineering Institute' der Carnegie Mellon Universität
- Dient dazu, eine passende Software Architektur für ein Software System auszuwählen
- Führende Methode im Bereich der Architektur-Software-Bewertung
- Eine Bewertung mit der ATAM dauert normalerweise drei bis vier Tage

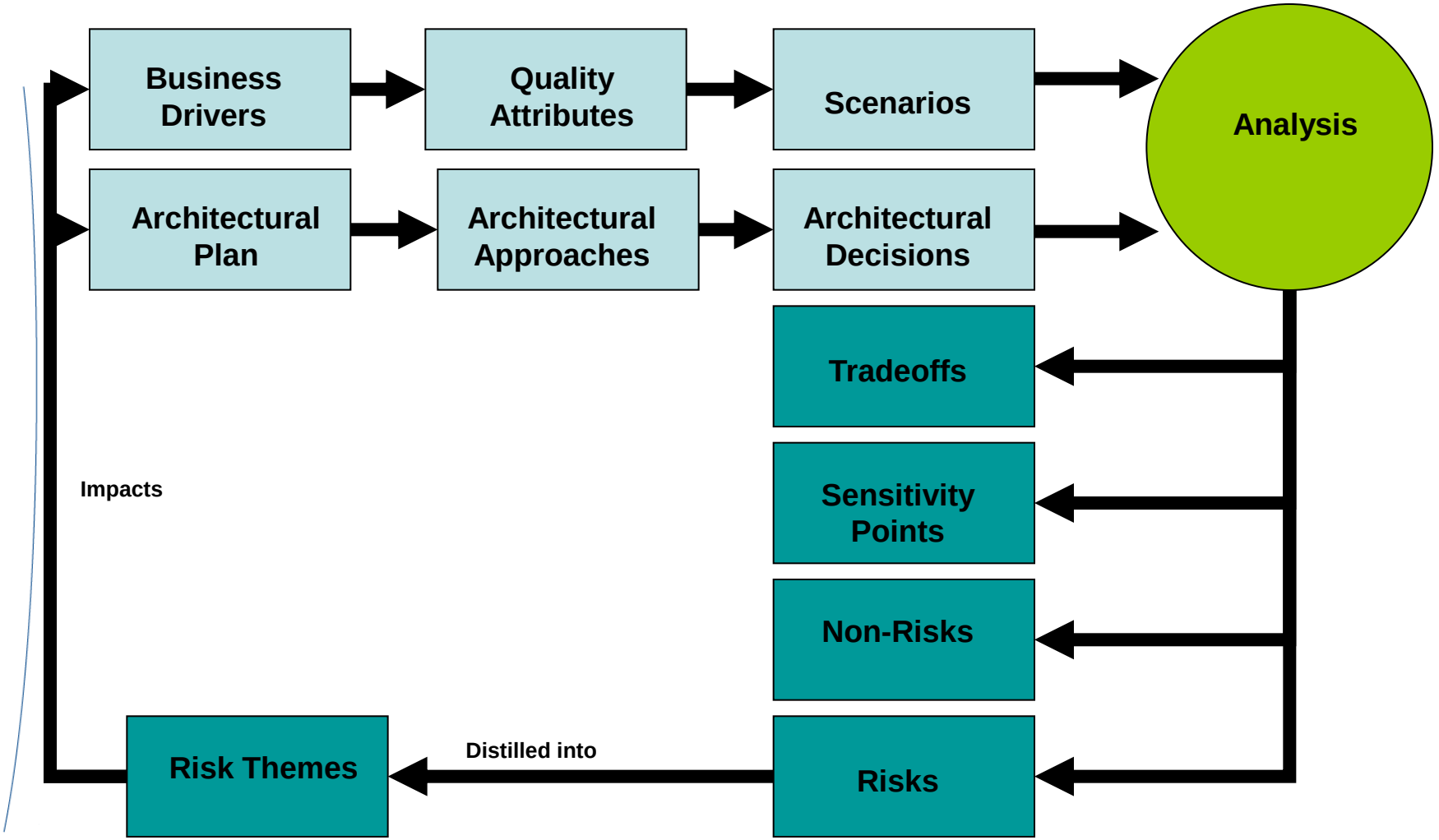
Bewährte Vorteile von ATAM

- eindeutige Attributs-Qualitätsanforderungen
- verbesserte Architektur-Dokumentation
- dokumentierte Grundlage für architektonische Entscheidungen
- identifiziert Risiken frühzeitig im Lebenszyklus
- verbesserte Kommunikation zwischen den Beteiligten

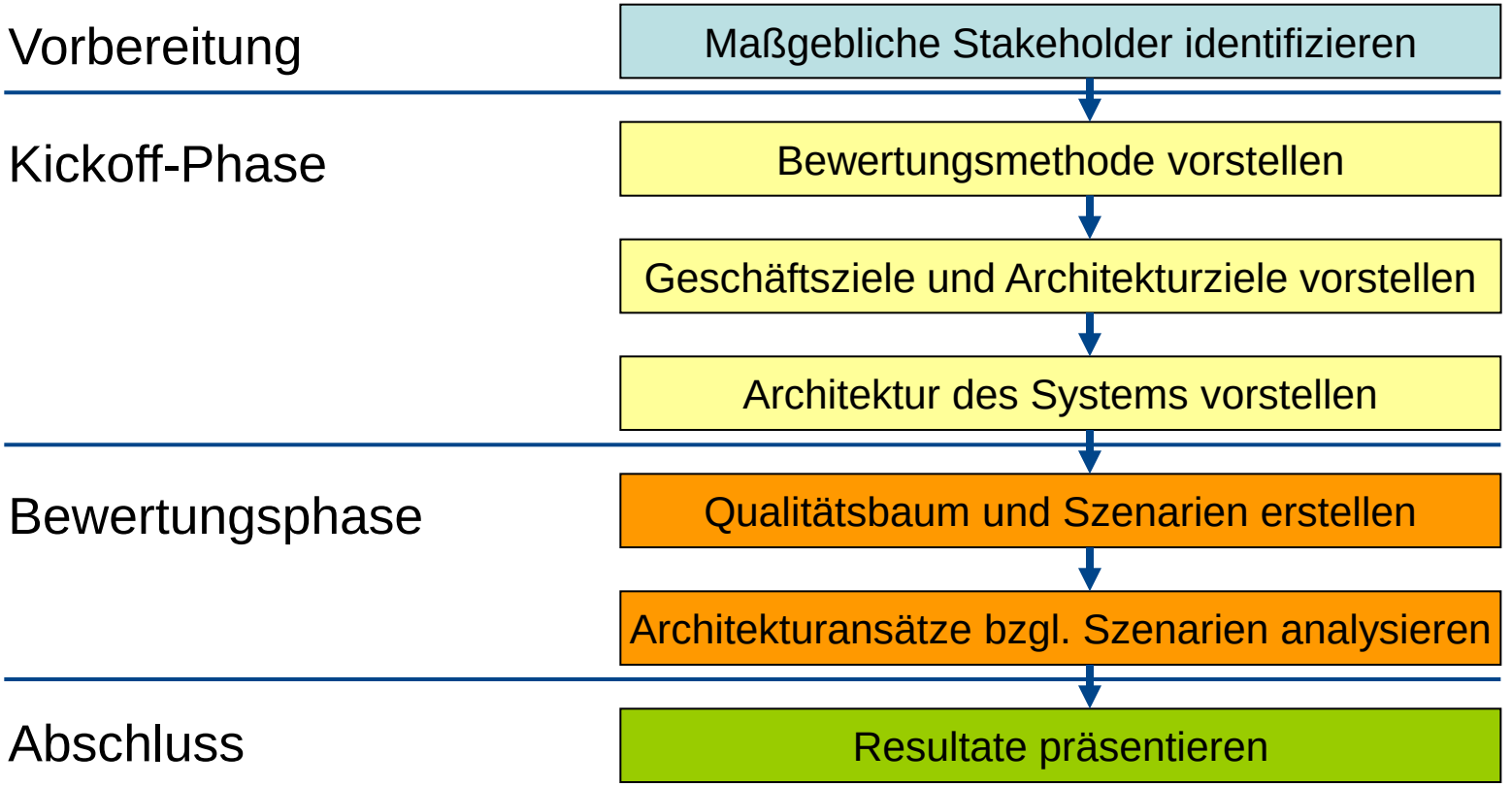
Voraussetzungen für ATAM



ATAM - Konzeptionelles Schema

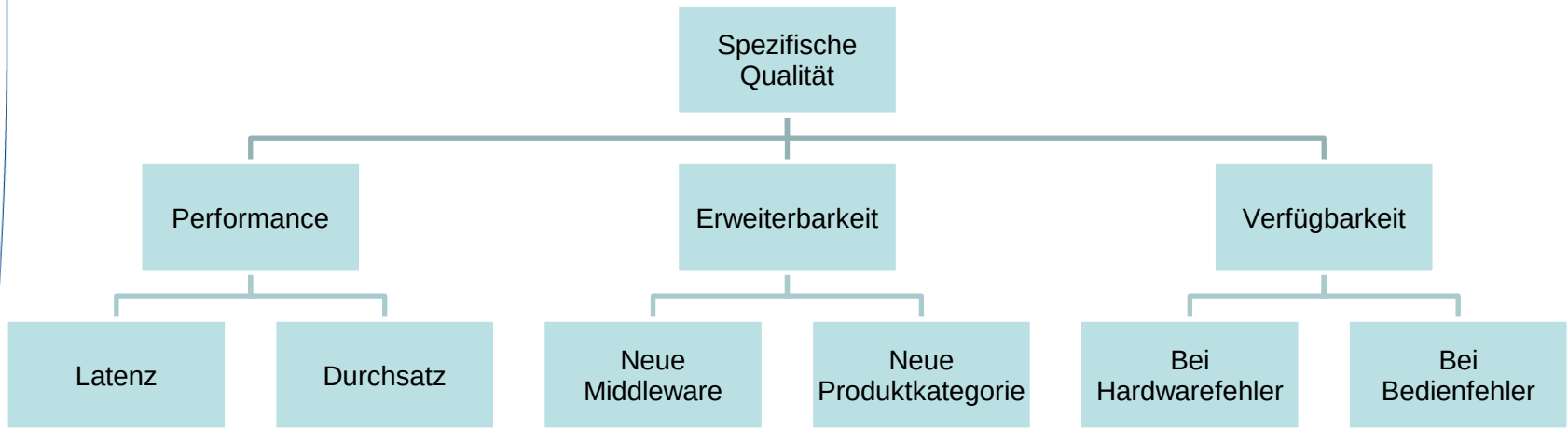


Vorgehen bei der Bewertung



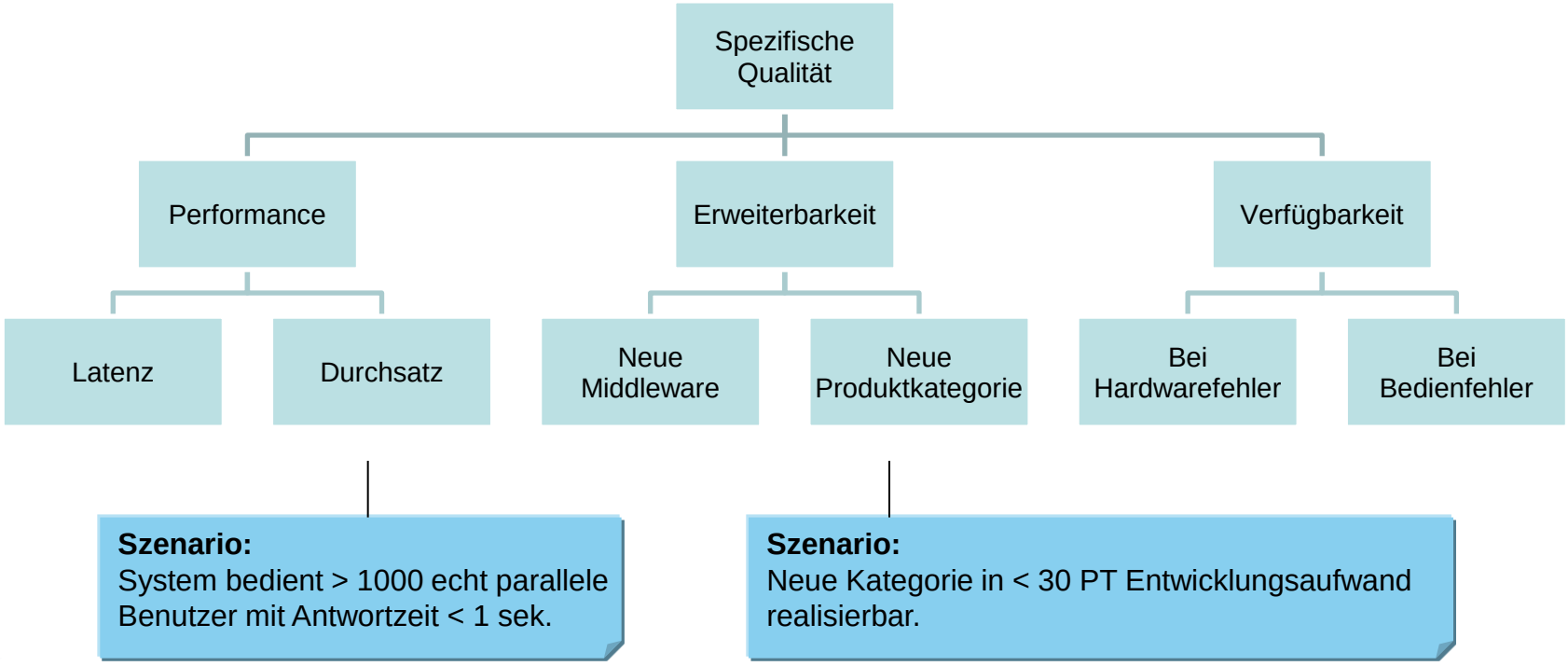
Qualitätsbaum erstellen

- Hierarchische Form
  - Allgemeine Merkmale stehen links (bzw. oben)
  - Speziellere Anforderungen stehen rechts (bzw. unten)



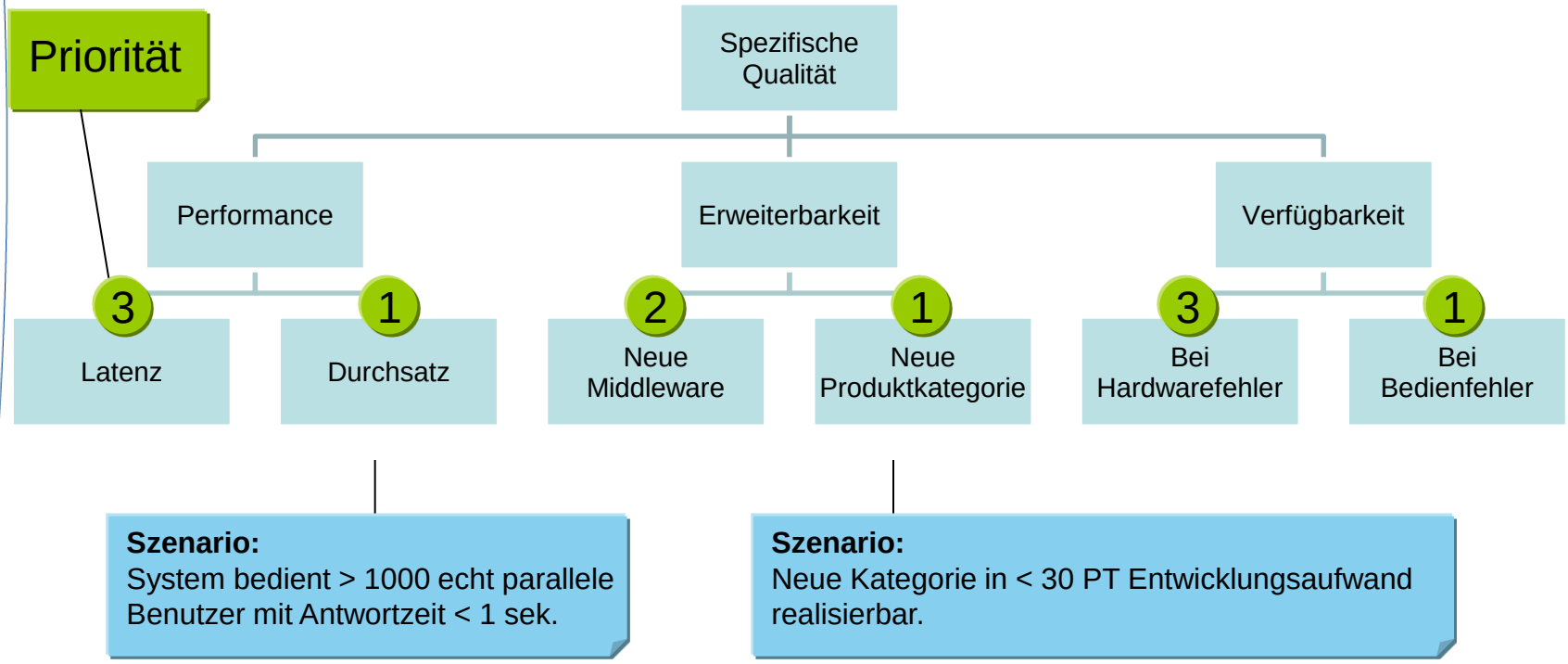
Qualitätsbaum erstellen

- Wichtigste Qualitätsziele durch Szenarien beschreiben



Qualitätsbaum erstellen

- Qualitätsmerkmale und Szenarien priorisieren



Bewertung hinsichtlich der Qualitätsmerkmale

- In kleinen Gruppen gemeinsam mit dem Architekten
- Reihenfolge gemäß Prioritäten

⇒ Beantwortung verschiedener Fragen

Bewertung hinsichtlich der Qualitätsmerkmale

- Welche Architekturentscheidungen wurden zur Erreichung eines Szenarios getroffen?
- Welcher Architekturansatz unterstützt die Erreichung des Szenarios?
- Welche Analysen, Prototypen oder Untersuchungen stützen diese Entscheidung?
- Wurden andere Qualitätsmerkmale oder Architekturziele beeinflusst?
- Welche Kompromisse wurden eingegangen?
- Welche Risiken bestehen?

Ergebnis

- Überblick über
  - Die Güte hinsichtlich konkreter Szenarien und der spezifischen Architekturziele
  - Risiken bei der Umsetzung der wichtigsten Szenarien
  - Maßnahmen zur Verhinderung der Risiken
  - Szenarien die ohne Risiken erreicht werden können

Bewertung von Software-Architekturen

Qualitativ

- Bewertung nach Beschaffenheit oder Güte
- Bewertung von Qualitätsmerkmalen
- Hilft Risiken zu identifizieren
- Bewertung sollte regelmäßig und so früh wie möglich stattfinden

Quantitativ

- Bewertung der Artefakte in Zahlen
- Geben gute Hinweise für strukturelle Veränderungen
- Keine Aussage über Funktionsfähigkeit und Qualität zur Laufzeit
- Benötigen fachlichen und technischen Kontext um vergleichbar zu sein
  - Tipp: Daten sammeln aus Vergleichsprojekten

Quantitative Metriken

- Anforderungen
  - Anzahl der geänderten Anforderungen pro Zeiteinheit
- Tests und Testfälle
  - Anzahl der Testfälle
  - Anzahl der Testfälle pro Klasse/Paket
  - Anzahl der Testfälle pro Anforderung
  - Testabdeckung
- Fehler
  - Mittlere Zeit bis zur Behebung eines Fehlers
  - Anzahl der gefundenen Fehler pro Paket

Quantitative Metriken

- Prozesse
  - Anzahl der implementierten/getesteten Features pro Zeiteinheit
  - Anzahl der neuen Codezeilen pro Zeiteinheit
  - Zeit für Meetings in Relation zur gesamten Arbeitszeit
  - Verhältnis der geschätzten zu den benötigten Arbeitstagen (pro Artefakt)

Quantitative Metriken

- Quellcode
  - Anzahl der Codezeilen (Lines of Code, LOC)
  - Abhängigkeitsmaße (Kopplung)
  - Anzahl der Kommentare in Relation zur Anzahl der Programmzeilen
  - Anzahl statischer Methoden
  - Komplexität (der möglichen Ablaufpfade, zyklomatische Komplexität)
  - Anzahl der Methoden pro Klasse
  - Vererbungstiefe

Beispiel: Zyklomatische Komplexität

- McCabe-Metrik, 1976 durch Thomas J. McCabe eingeführt
- Zeigt die Anzahl voneinander unabhängiger linearer Pfade eines Softwaremoduls

- Zyklomatische Komplexität = e − n + 2p
  - e: Anzahl Kanten im Graphen
  - n: Anzahl Knoten im Graphen
  - p: Anzahl der einzelnen Kontrollflussgraphen (ein Graph pro Funktion/Prozedur)

Niedrige zyklomatische Komplexität

- Modul ist einfach zu verstehen, zu testen und zu pflegen

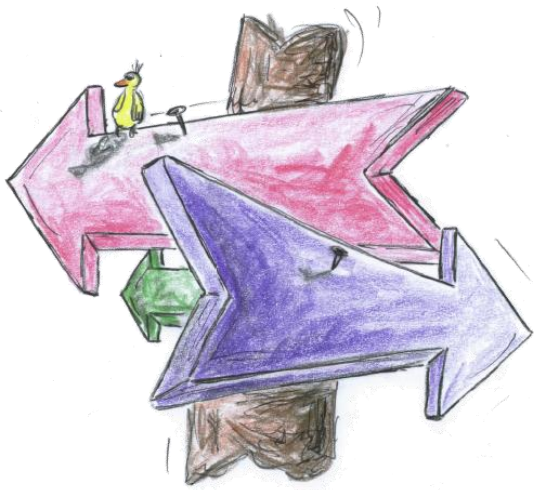
Hohe zyklomatische Komplexität

- Modul ist sehr komplex und zu schwer zu testen

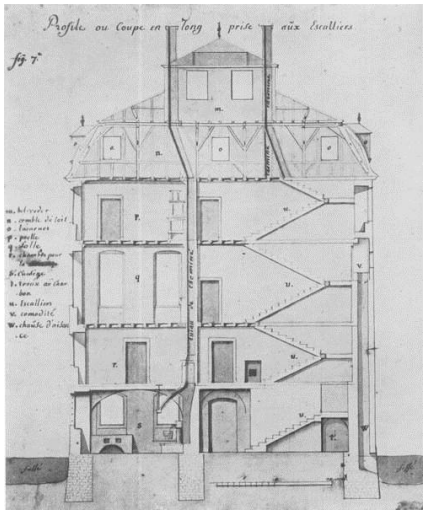
- Warnung kann unterdrückt werden, wenn
  - sich die Komplexität nur schwerlich verringern lässt
  - die Methode einfach zu verstehen, zu testen und zu warten ist
  - Beispiel: Methoden mit umfangreichen switch-Anweisungen

Agenda

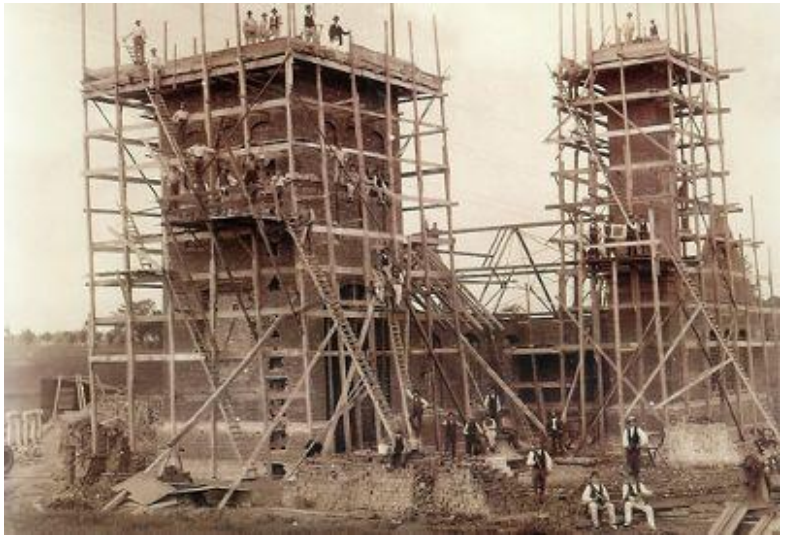
- Motivation
- Bewertung von Software-Architekturen
  - Qualitative Bewertung
  - Quantitative Bewertung
- Wie tief, wie gut, wie sinnvoll?



Architekturanalyse begleitet den Entwicklungsprozess

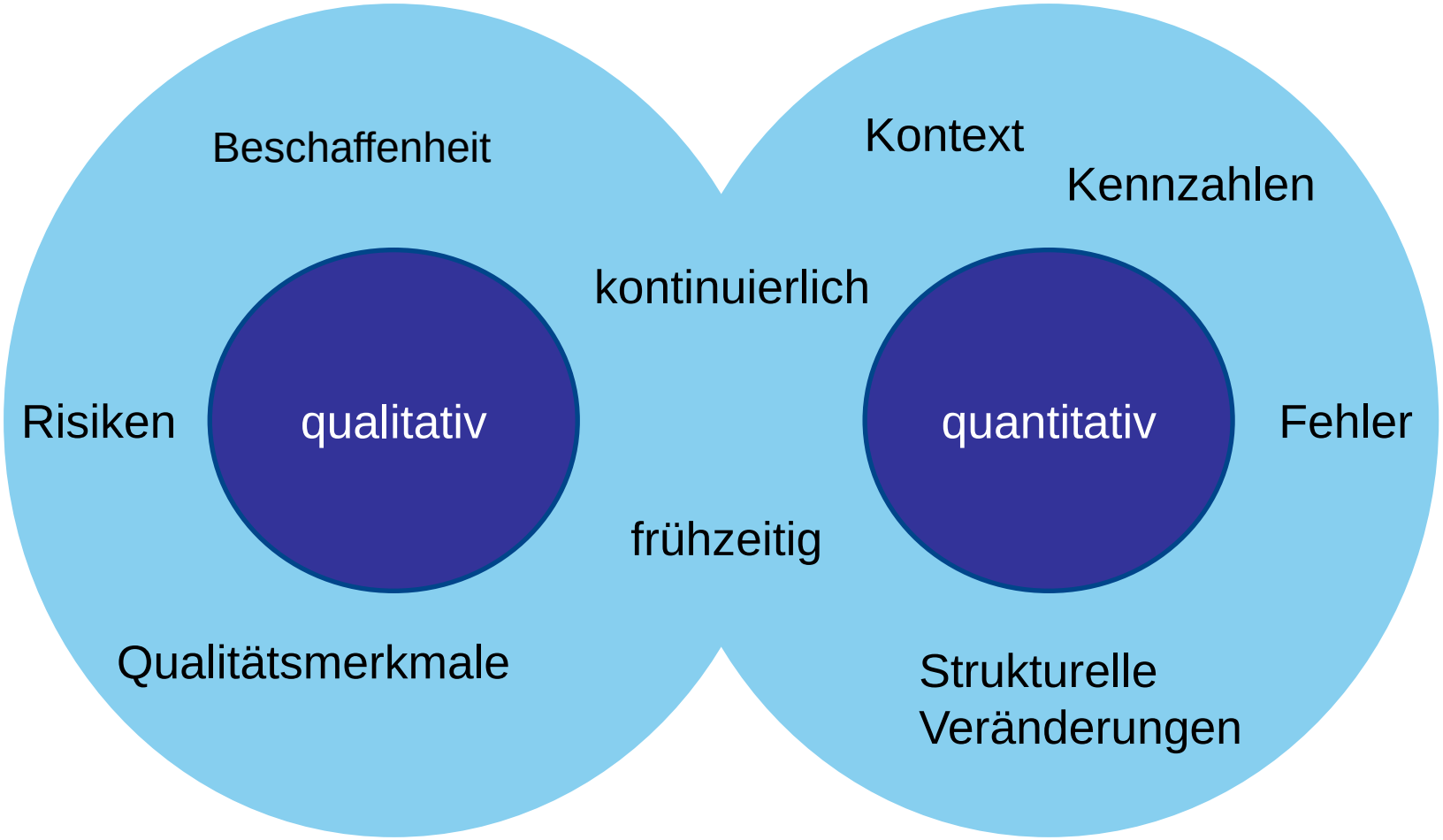


Nicht nur die Architektur muss bewertet werden,

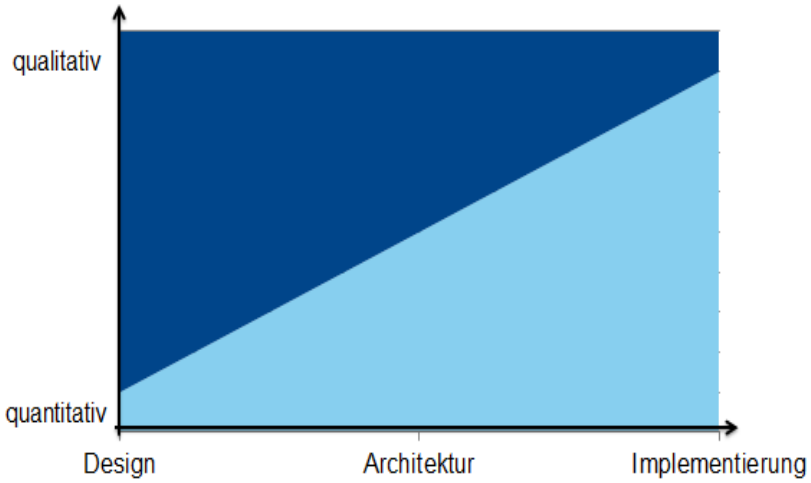


sondern auch die Umsetzung!

Wie tief?



Wie tief?

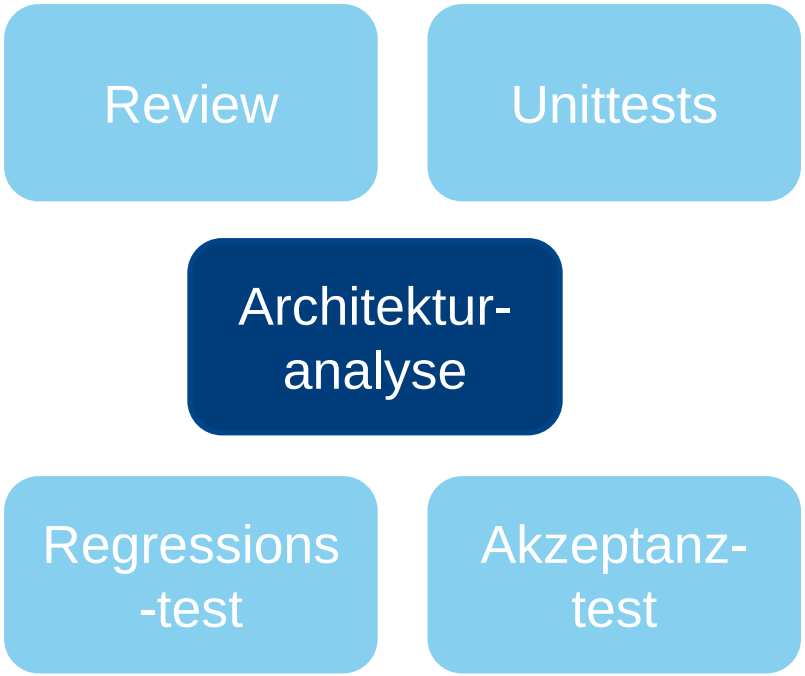


- Die eingesetzten Methoden und Werkzeuge orientieren sich an der Projektphase.
- Die Werkzeugauswahl orientiert sich an den zu prüfenden Metriken.
- Anzahl der Metriken sollte minimiert werden
- Analyse sollte auf überschaubare Ausschnitte konzentriert werden

Wie gut?

- Codemetriken können gute Hinweise für strukturelle Veränderungen von Software geben. Über Funktionsfähigkeit und Qualität zur Laufzeit sagen sie hingegen nichts aus.
- Effizienz und Effektivität der Architekturanalyse werden durch die Werkzeugauswahl mitbestimmt.
- Über Kennzahlen können Qualitätsmerkmale messbar und verfolgbar und Architekturen vergleichbar gemacht werden.
- Die Ergebnisse der Architekturanalyse müssen kommuniziert werden.

Wie sinnvoll?



- Die Architekturanalyse begleitet den Software-entwicklungsprozess wie z.B. Reviews und Tests.
- Die Architekturanalyse schafft einen gemeinsamen Blick auf die Architektur.
- Die Architekturanalyse macht Architekturen vergleichbar.
- Die Architekturanalyse deckt Fehler und Risiken frühzeitig auf und liefert Ansatzpunkte für Gegenmaßnahmen.

Fazit:

Architekturbewertung: Keine Noten aber mehr Durchblick!

Referenzen

- VL Software-Qualitätsmanagement, Universität Leipzig
- Effektive Software-Architekturen – Ein praktischer Leitfaden
  - Gernot Starke, Hanser Verlag
- DIN ISO/IEC 9126
- ATAM: Method for Architecture Evaluation**,** R.Kazman, M. Klein, P.Clements