Anforderungen klären

Stand: April 2020

www.arc42.de

Ziele und Inhalt

Sie lernen:

- was Architekten als Vorgabe erwarten dürfen
- was Sie tun sollten, wenn die Vorgabe nicht OK ist
- wie Sie explizite Qualitätsziele für Architekturen setzt

Sie üben:

- Qualitätsanforderungen Ihres Systems explizit zu machen

Lernziele gemäß iSAQB CPSA-F:

- **LZ 1-6: Zusammenhang zwischen Entwicklungsvorgehen und Softwarearchitektur erläutern können (R1)**
- **LZ 1-7: Kurz- und langfristige Ziele differenzieren (R1)**
- **LZ 1-8: Explizite von impliziten Aussagen unterscheiden (R1)**
- LZ 1-9: Zuständigkeit von Softwarearchitekten in organisatorischen Kontext einordnen (R3)
- **LZ 2-3: Einflussfaktoren auf Softwarearchitektur erheben und berücksichtigen können (R1)**
- **LZ 3-5: Kontextabgrenzung von Systemen erläutern und anwenden (R1)**
- **LZ 4-1: Qualitätsmodelle und Qualitätsmerkmale diskutieren (R1)**
- **LZ 4-2: Qualitätsanforderungen an Softwarearchitekturen klären (R1)**

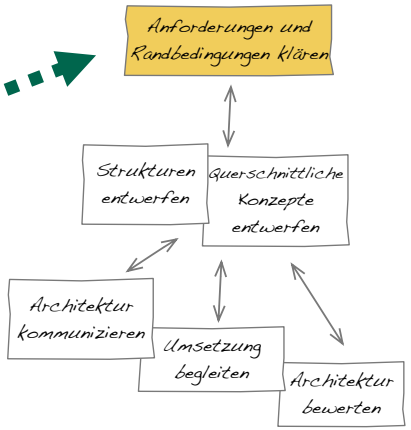
Die Vorgaben: Analyseergebnisse

- Annahme:
  Jemand legt (Teile der) Anforderungen fest.
  (z.B. Kunden, Marketing, Product Owner, Business Analysts, Requirements Engineers)

- Als Architekt: insbesondere **architekturrelevante Anforderungen** verstehen und bezüglich Vollständigkeit, Relevanz und Risiken einschätzen.

- (Theoretisch:) Alles nicht Geforderte ist Freiheitsgrad für Ihre Entwurfsentscheidungen!
  - Aktiv nachhaken!

Architektur-relevante Anforderungen



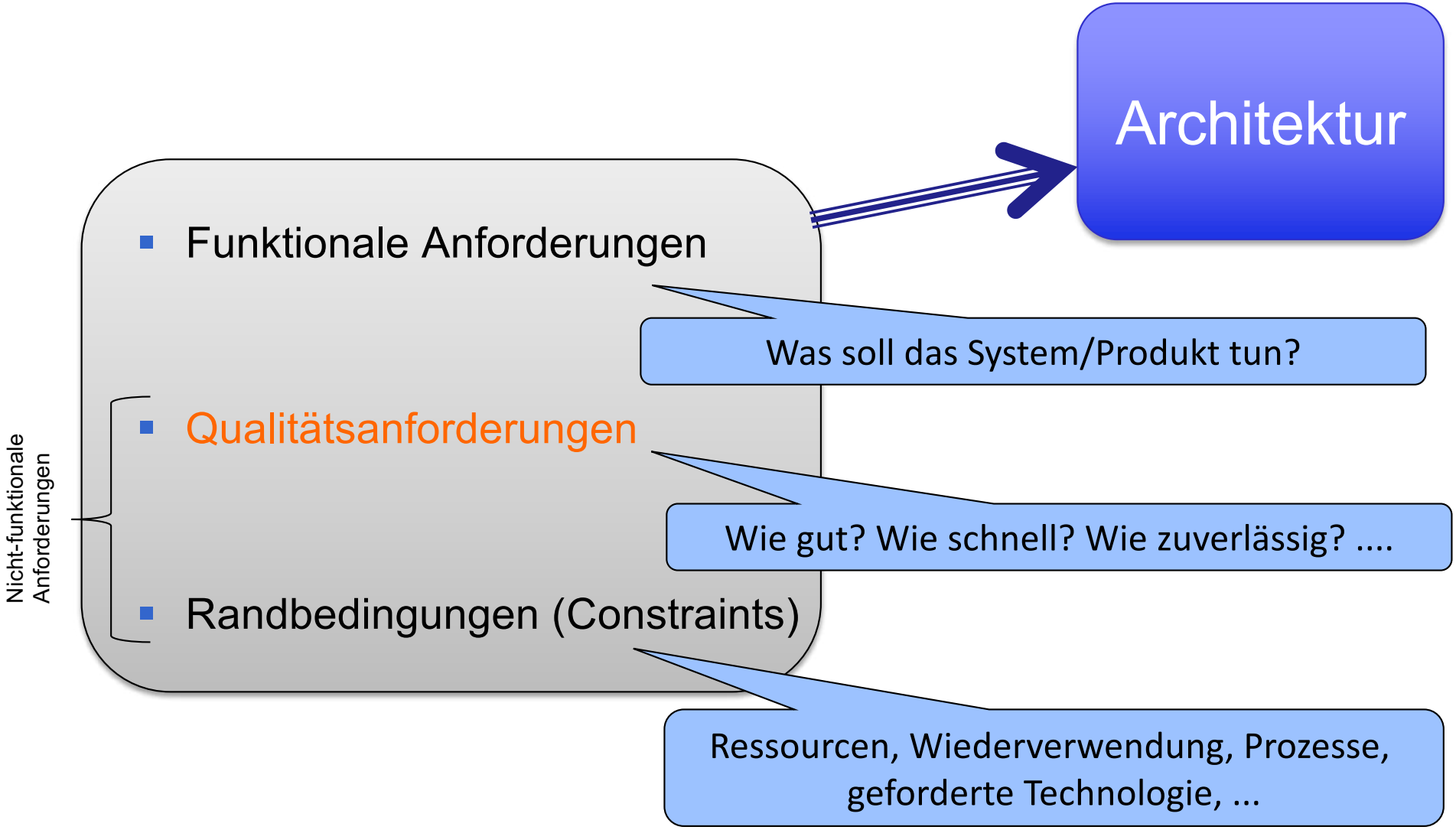
ArA: Anforderungen, die besondere Auswirkung auf die Architektur haben (können).

SoftwarearchitektInnen müssen diese Anforderungen:

- identifizieren oder finden und
- kommunizieren.

Andere Anforderungen können – genau wie die Architektur – iterativ ermittelt und kommuniziert werden.

Drei Arten von Anforderungen...



www.**VOLERE**.co.uk

PROJEKTTREIBER

1. Ziele des Projekts
2. Stakeholder

RANDBEDINGUNGEN FÜR DAS PROJEKT

3. Geforderte Randbedingungen
4. Namenskonventionen und Terminologie
5. Relevante Fakten und Annahmen

FUNKTIONALE ANFORDERUNGEN

6. Einbettung in das Arbeitsgebiet
7. Datenmodell
8. Umfang des Produkts
9. Funktionale Anforderungen

NICHT-FUNKTIONALE ANFORDERUNGEN

10. Anforderungen an die Benutzungsschnittstelle
11. Ergonomieanforderungen
12. Performanzanforderungen
13. Operative und Umwelt-Anforderungen
14. Anforderungen an Wartbarkeit und Support
15. Sicherheitsanforderungen
16. Kulturelle Anforderungen
17. Rechtliche Anforderungen

PROJEKTASPEKTE

18. Offene Punkte
19. Kauf und Wiederverwendung
20. Neue Probleme
21. Projektplanung
22. Anforderungen an Migration und Inbetriebnahme
23. Risiken
24. Kosten
25. Anwenderdokumentation und -schulung
26. Warteraum
27. Lösungsideen

Die wichtigsten Inhalte *):

- Projektziele
- Stakeholder
- Scopeabgrenzung
- Funktionale Anforderungen
- Qualitätsanforderungen
- Randbedingungen

*) Mehr dazu siehe www.req42.de

James & Suzanne Robertson:
Mastering the Requirements Process, 3rd Edition
Addison Wesley, 2012

Erfolgreicher Projektstart =

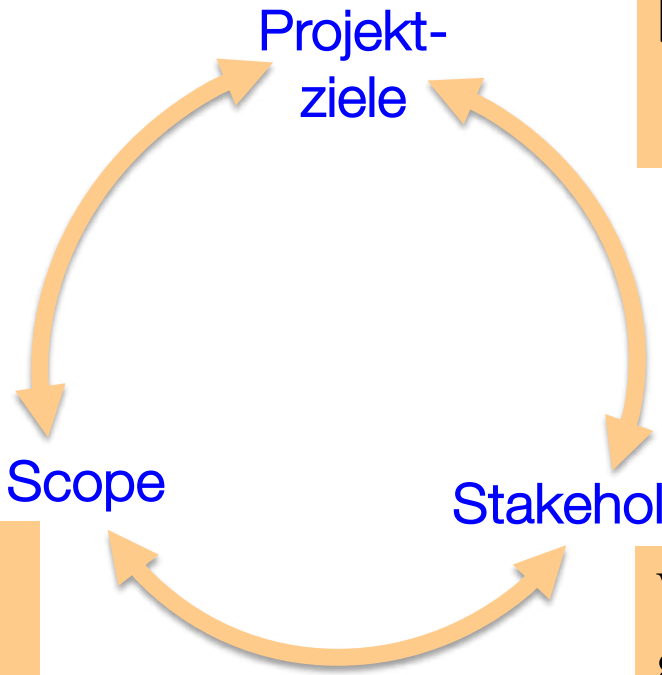
Ausbalancieren von drei Zutaten:

Was soll in diesem Projekt erreicht werden?

P (Purpose)
A (Advantage)
M (Measure)

P (Purpose)
A (Advantage)
M (Measure)

P (Purpose)
A (Advantage)
M (Measure)



Was gehört zu diesem Projekt und was liegt außerhalb?

(Abgrenzung der Schnittstellen zu Nachbarsystemen und Umwelt)

Wer soll in diesem Projekt gehört werden?

(Alle Projektbeteiligten und Projektbetroffenen)

Funktionale Anforderungen

Geforderte Abläufe (Funktionen) + Begriffsdefinitonen (Daten)

Viele alternative Notationen:

- Umgangssprachliche Beschreibung
  - „das System soll .....“
  - oder als Epics, Features, Storys
- Modelle:
  - Use-Cases
  - Fluss-/Aktivitätsdiagramme
  - BPMN
  - ....

Glossar

aller fachlich relevanten Begriffe
(Data Dictionary, Begriffslexikon, ...)

Begriff	Synonym	Definition

Nicht-funktionale Anforderungen

Qualitätsanforderungen + **Randbedingungen (Constraints)**

- Benutzbarkeitsanforderungen
- Effizienzanforderungen
- Zuverlässigkeit und Verfügbarkeitsanforderungen
- Änderbarkeitsanforderungen
- Portabilitätsanforderungen
- Sicherheitsanforderungen
- Gesetzliche Anforderungen

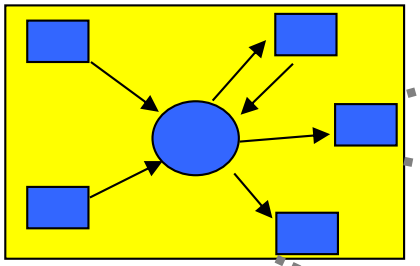
- Randbedingungen für das System/Produkt
  - Vorgaben für Einbettung und Verteilung
  - Vorgeschriebene Technologien
  - Physikalische Anforderungen
  - Umweltanforderungen
- Randbedingungen an den Prozess
  - Anforderungen an das Vorgehensmodell
  - Anforderungen bezüglich Systemsupport
- Managementrandbedingungen
  - Vorgaben über Zeit, Budget, Personal, ...

vgl.
ISO/IEC 9126
oder
ISO 25010

Architekten dürfen als Ausgangspunkt Ihrer Arbeit folgendes erwarten:

Projektziele

Scope/Kontextabgrenzung



Funktionale Anforderungen

z.B. als Epics/Features/Storys

oder

als Use-Cases/Aktivitätsdiagramme mit zugehörigen Beschreibungen

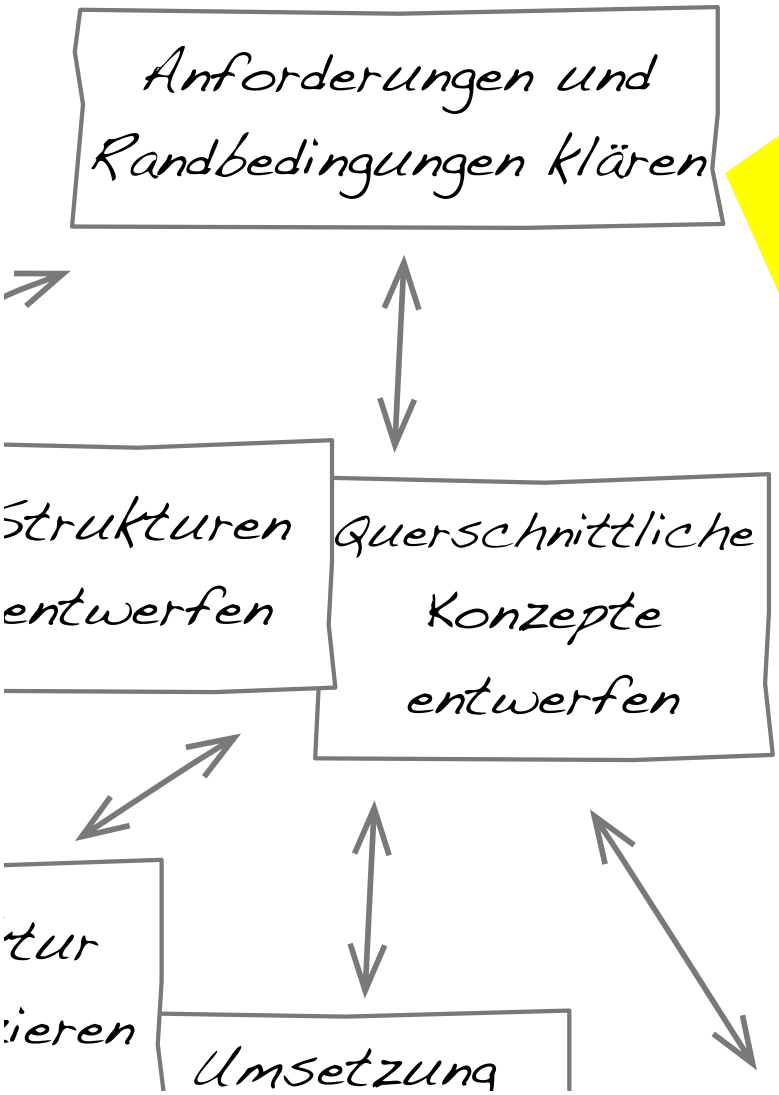
Glossar/ fachliche Begriffs-definitionen

Qualitäts-anforderungen & Randbedingungen

Der (traurige) Stand der Praxis

benötigt	**(praktisch vorhanden)**
Projektziele	meist implizit bekannt 
Stakeholder	oft unvollständig, nicht schriftlich 
Scope/Kontextabgrenzung	gar nicht oder unvollständig 
Funktionale Anforderungen	
Funktionen & Abläufe	am ehesten vorgegeben, meist umgangssprachlich 
Daten	oft implizit, ohne Glossare 
Qualitätsanforderungen	fehlt oft 
Randbedingungen	oft bekannt, aber nicht schriftlich erfasst 

Was müssen Architekten tun?



- Projektziele und funktionale Anforderungen verstehen
- Stakeholder vervollständigen
- Scope/Kontext präzisieren (oder festlegen)
- Qualitätsanforderungen und Randbedingungen präzisieren (oder festlegen)

Kenne Deine Stakeholder!

Kap. 1.3

Rolle	Beschreibung	Ziele / Erwartungen	Kontakt	Bemerkung

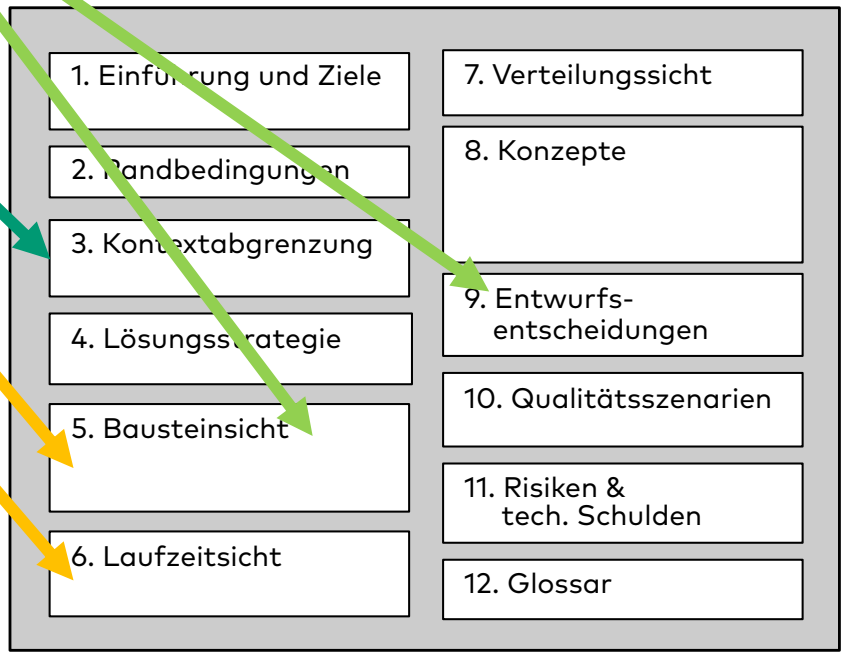
Beispiele:

Management, Auftraggeber, Projekt-Steuerkreis, sonstige Projekt-Gremien, PMO, Projektmanager, Produktmanager, Fachbereich, Unternehmens-/Enterprise-Architekt, Architektur-Abteilung, Methoden-Abteilung, QS-Abteilung, IT-Strategie, Software-Architekt, Software-Designer, Entwickler, Tester, Konfigurationsmanager, Build-Manager, Release-Manager, Wartungsteam, externe Dienstleister, Zulieferer, Hardware-Designer, Rollout-Manager, Infrastruktur-Planer, Sicherheitsbeauftragter, Behörde, Aufsichtsgremium, Auditor, Mitbewerber/Konkurrent, Endanwender, Fachpresse, Fachadministrator, Systemadministrator, Operator, Hotline, Betriebsrat, Lieferant von Standardsoftware, verbundene Projekte, Normierungsgremium, Gesetzgeber...

Erwartungshaltung...

Was muss Architektur denen liefern?
Oder umgekehrt!

Rolle	Ziel / Intention	**Erwartete Liefergegenstände**	Kontakt	P… Abn… …levanz	…g



(fachliche) Kontextabgrenzung

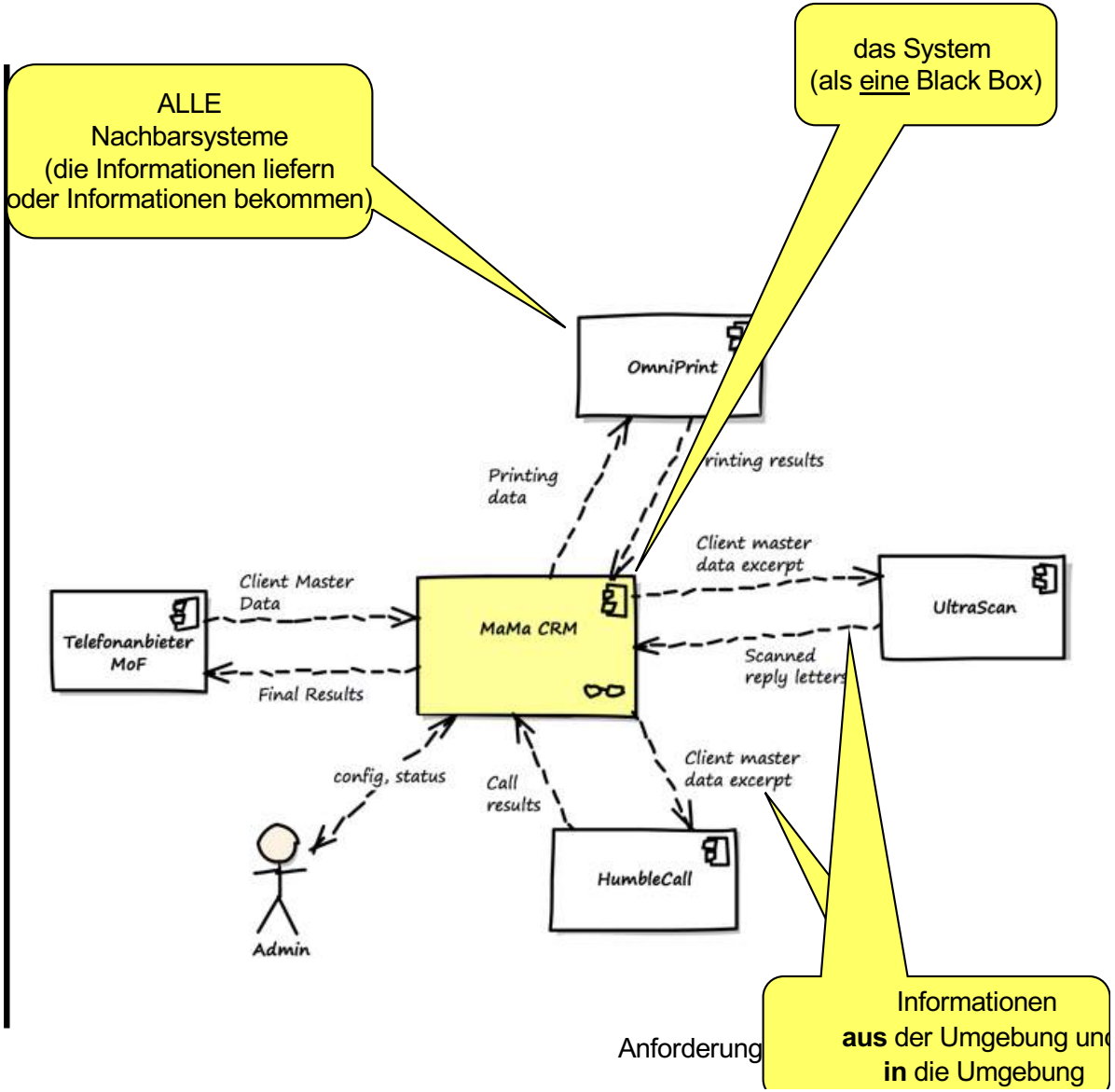
Kap. 3.1

- Legt Schnittstellen zwischen zu erstellendem System und Umwelt fest.

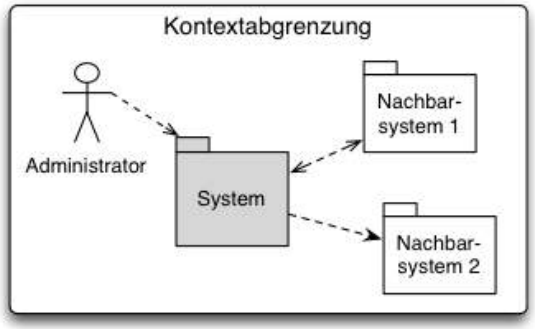
Tabellarische Kontextabgrenzung:

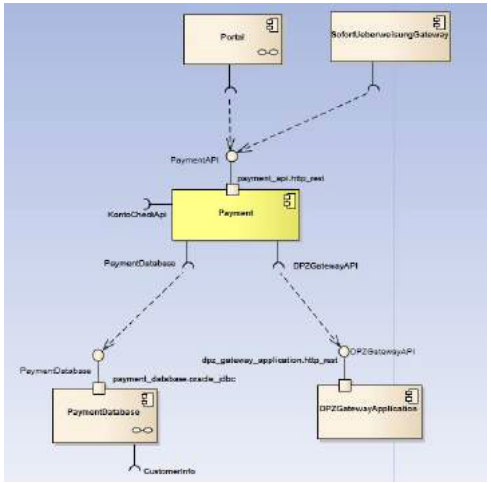
Buchbestellsystem

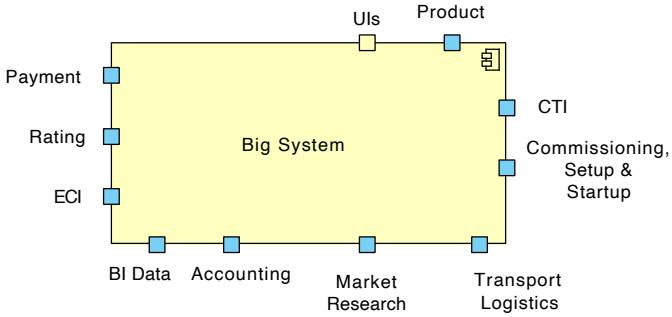
	IN	OUT
Kunde	Bestellung	Bücher, Rechnung, Lieferschein
Druckerei	Bücher	Druckauftrag
Management	Kreditstatus	Umsatzbericht
Buchhaltung	---	Rechnung



Notationen für Kontextabgrenzung





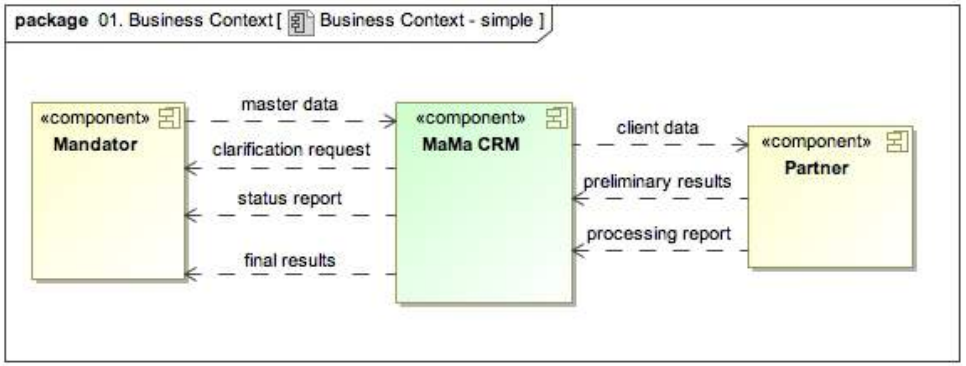


Nachbar (-system)	Ausgetauschte Daten	Erklärung / Bedeutung
Administrator	-> Admin-Befehle	User- und Rechtemanagement, DB-Konfiguration ...
Nachbarsystem-1	-> x-Status <- y-Kommando	....
Nachbarsystem-2	<- Z-Daten	.....

Tabelle präzisiert Ein-/Ausgaben

Praxistipps Kontextabgrenzung

- Beginne Kontext am Tag-0!
  - Starte grob
  - Detailliere später

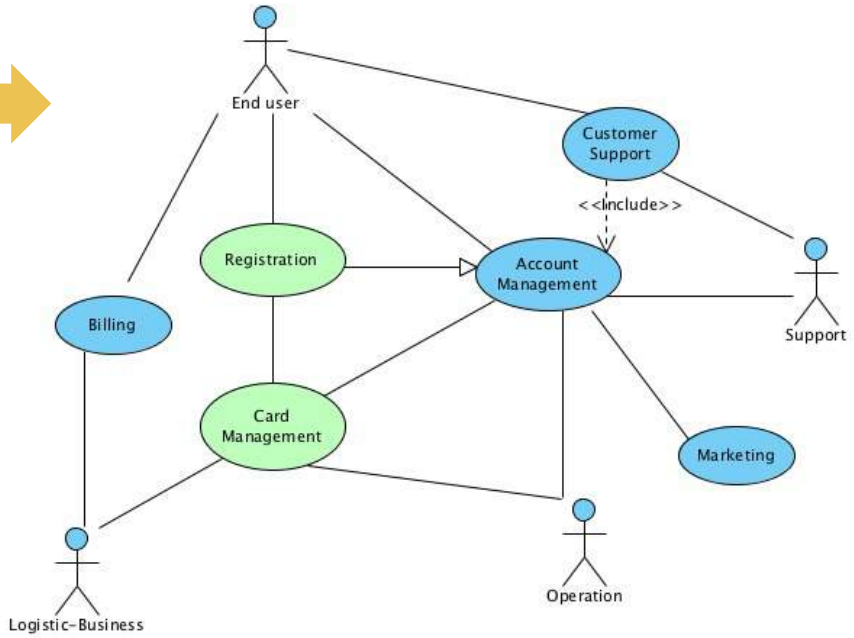


- Unterscheide fachlichen und technischen Kontext
- UML-relaxed: Datenflüsse statt Aufrufe/Abhängigkeiten
- Hole Feedback zum Kontext von **allen** Stakeholder

- Viele Nachbarn?
  Bilde (fachliche) Cluster
- Volatile Nachbarn?
  Explizit darauf hinweisen
- Bewerte die Risiken aller externen Schnittstellen
- Starte KEIN Projekt ohne Kontextabgrenzung.

Praxistipps funktionale Anforderungen

- Bilde Use-Case Cluster
  - Grobe Struktur der Funktionen
  - Abstrahiere: „Black-Box Funktionen“
  - Detailliere später



- Suche Fachbegriffe („domain stuff“) in Funktionen/Abläufen
- Kläre Funktionen an konkreten Beispielen („Szenarien“)

- Implementiere Akzeptanztests (acceptance test driven development) für komplexe Funktionen

Qualitätsziele beantworten die Frage: Wann ist die Architektur „gut“?

- Es gibt kein absolutes Maß für „gut“ oder „schlecht“...
  - ... nur Architekturen, die festgelegte Ziele und Qualitätsmerkmale mehr oder weniger erreichen
- Man kann nicht alle Qualitätsmerkmale gleichzeitig optimieren, sondern muss immer Kompromisse eingehen

Beispiele:

- Verteilte 3-Ebenen-Architektur ist
  - ideal für ein Finanz-Reporting-System einer Firma,
  - Wahnsinn für ein Avionic-System eines Flugzeuges.
- Eine Architektur mit einem Höchstmaß an Flexibilität ist sinnlos für eine Einmal-Wegwerf-Anwendung.

Hitparade der Qualitätsanforderungen!!

- Tabelle der Top-3 architekturrelevanten Qualitäts-anforderungen:
  - Erfrage Prioritäten bei maßgeblichen Stakeholdern

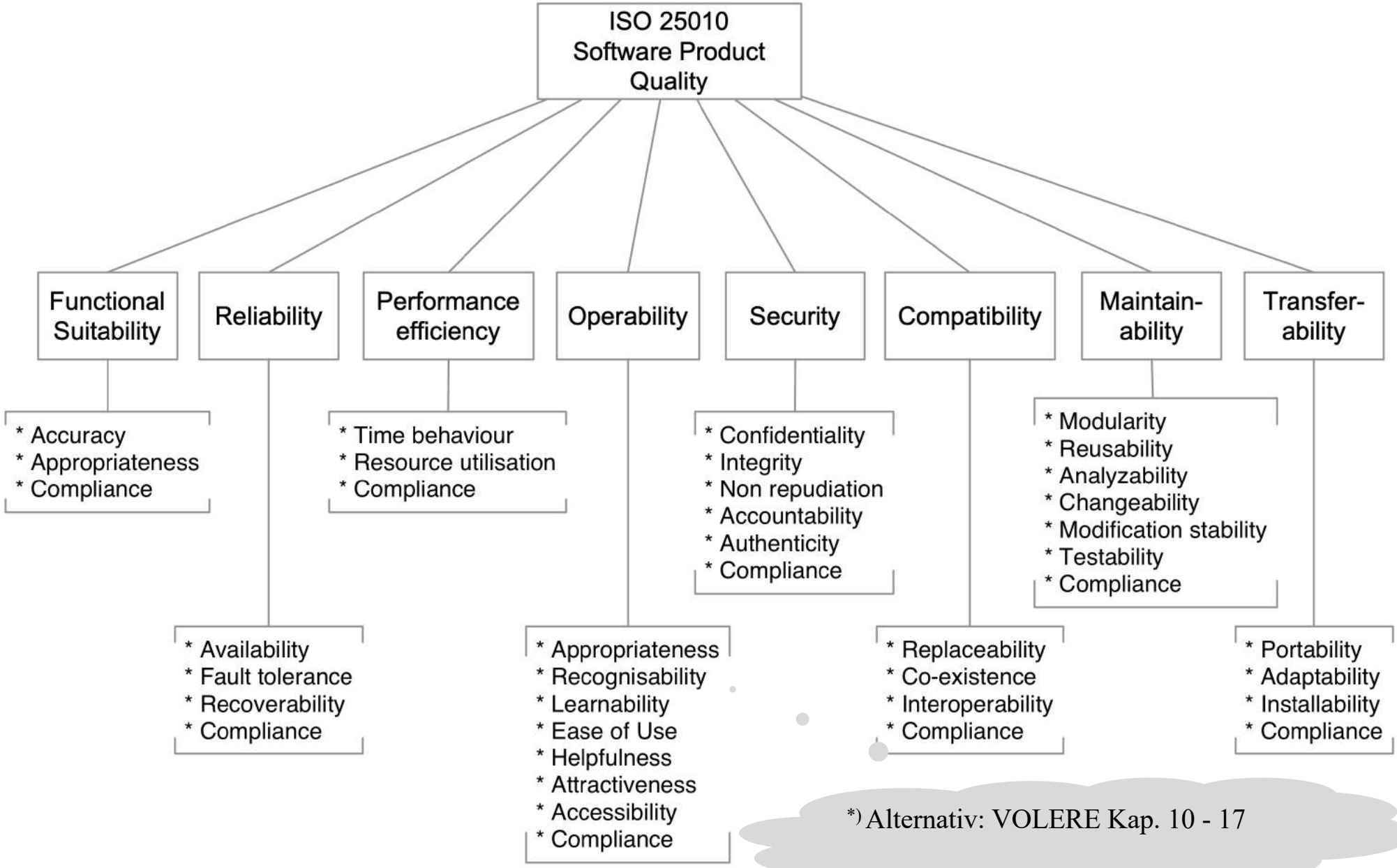
Kap. 1.2

Prio	Q-Anforderung	Bedeutung / Szenarien
1	Verfügbarkeit	System ist Werktags 7-19h für alle Sales-Use-Cases zu mind. 99% verfügbar
2	Performance	Suche in Artikel-Katalog liefert nach spätestens 2sek die ersten Treffer
3	...	.....

- Keine Q-Anforderungen bekommen?
  1. Triff Annahmen **(„educated guess“)**
  2. Dokumentiere Annahmen
  3. Hole Feedback dazu ein

- Konkretisiere Q-Anforderungen gemeinsam mit Stakeholdern

Nutzen Sie ein Standardmodell für Q-Anforderungen*) als Ausgangspunkt



*) Alternativ: VOLERE Kap. 10 - 17

Volere: Quality Requirements

10. Look and Feel Requirements
- Appearance Requirements
- Style Requirements

11. Usability Requirements
- Ease of Use
- Personalization & Internationalization
- Ease of Learning
- Accessibility Requirements

12. Performance & Safety Requirements
- Speed and Latency
- Safety Critical Req.
- Precision & Accuracy
- Reliability and Availability
- Robustness & Fault Tolerance
- Capacity
- Scalability and Extensibility
- Longevity

13. Operability Requirements
- Expected Physical Environment
- Interfacing with Adjacent Systems
- Productization Requirements
- Release Requirements

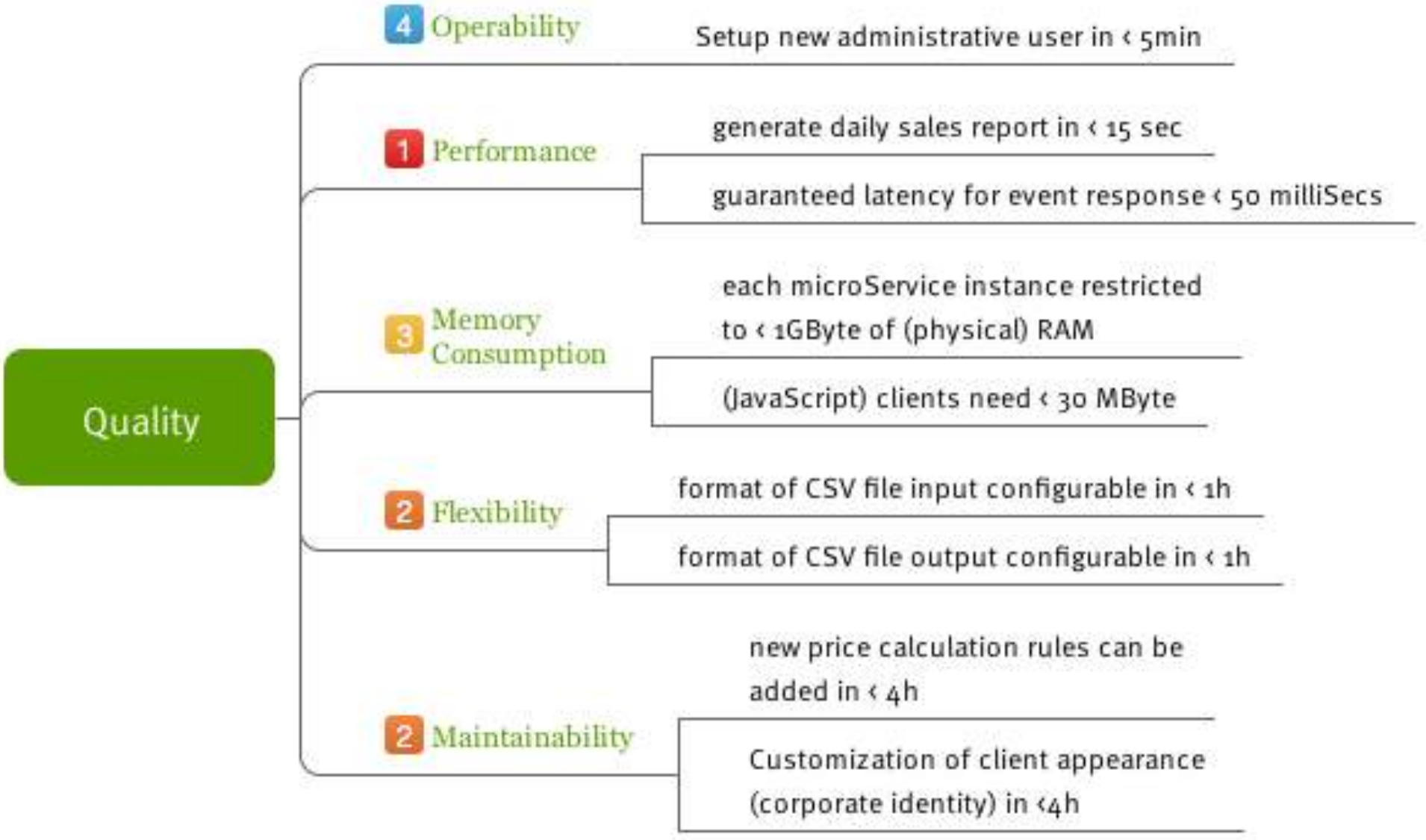
14. Maintainability & Support Requirements
- Maintenance Requirements
- Supportability
- Adaptability

15. Security Requirements
- Access Requirements
- Integrity Requirements
- Privacy Requirements
- Audit Requirements
- Immunity Requirements

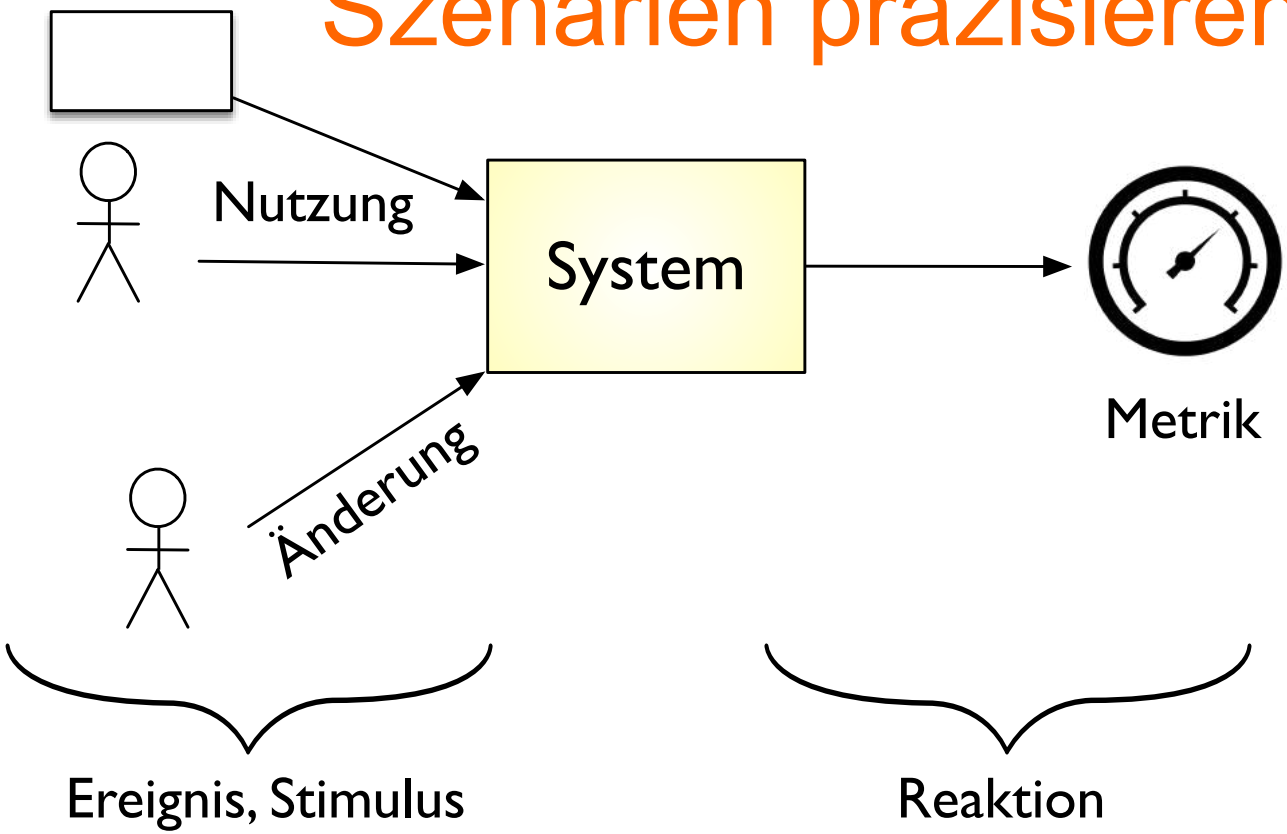
16. Cultural Requirements

17. Legal Requirements
- Compliance Requirements
- Standard Requirements

Qualitätsbaum mit Szenarien



Szenarien präzisieren Qualitätsziele



Interaktionen von Stakeholder und System

Wie reagiert das System auf Stimulus?

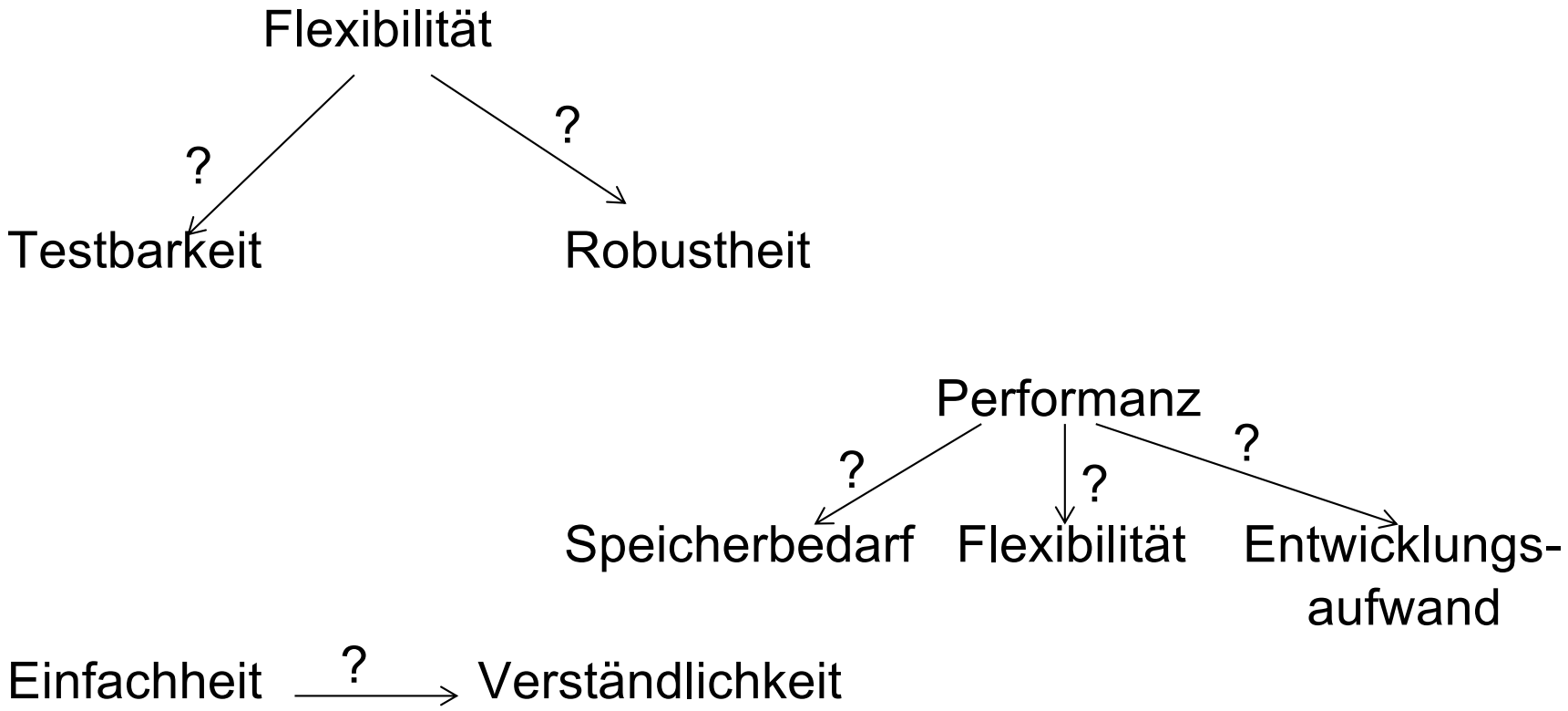
- Anwendungs-/Nutzungsszenarien (Use-Case Scenario)
  - Stakeholder wenden System an
- Änderungsszenarien (Change Scenario)
  - Stakeholder modifiziert Teil des Systems oder dessen unmittelbare Umgebung

Exemplarische Qualitätsszenarien

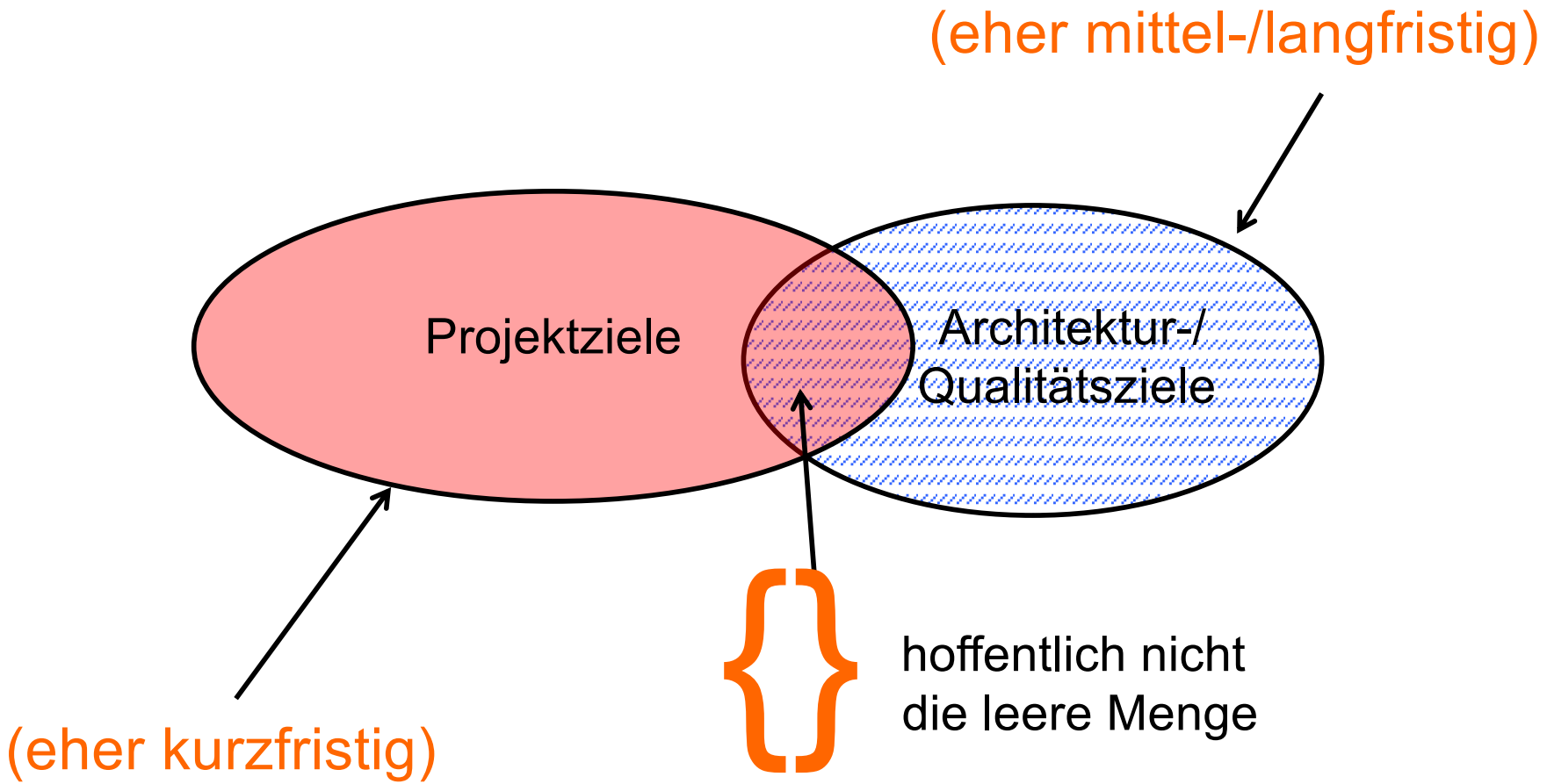
- X-Report in bei mittlerer Systemlast (< 100 parallele Benutzer) weniger als 15 Sekunden
- Y-Aufgabe an der GUI im Mittel
  - <1 Sekunde für 1- 20 concurrent-user
  - <5 Sekunden für 50 – 2000 concurrent user
- Bei CPU-Ausfall ist Hot-Spare in <5 Minuten online (ohne Datenverlust)
- Operator kann System in < 10 Minuten neu starten
- Erweiterung von Z um xy-GUI in weniger als 5 PT
- Neuer xy-Prozess in weniger als 10 PT in Produktion
- Neuer yz-Tarif in weniger als 6h produktiv

Ausführliche Beispiele für Q-Anforderungen:
https://github.com/arc42/quality-requirements

Abhängigkeiten / Korrelation von Qualitätseigenschaften



Projekt- und Architekturziele



- Beide müssen unter den Beteiligten verhandelt und abgestimmt werden.

Wohin mit den Ergebnissen?

1. Einführung und Ziele
1.1 Aufgabenstellung
1.2 Qualitätsziele
1.3 Stakeholder

2. Randbedingungen
2.1 Technische Randbedingungen
2.2 Organisatorische Randbedingungen
2.3 Konventionen

3. Kontextabgrenzung
3.1 Fachlicher Kontext
3.2 Technischer- oder Verteilungskontext

4. Lösungsstrategie

5. Bausteinsicht
5.1 Ebene 1
5.2 Ebene 2
….

6. Laufzeitsicht
6.1 Laufzeitszenario 1
6.2 Laufzeitszenario 2
….

7.Verteilungssicht
7.1 Infrastruktur Ebene 1
7.2 Infrastruktur Ebene 2
….

8. Konzepte
8.1 Fachliche Struktur und Modelle
8.2 Typische Muster und Strukturen
8.3 Persistenz
8.4 Benutzeroberfläche
….

9. Entwurfsentscheidungen
9.1 Entwurfsentscheidung 1
9.2 Entwurfsentscheidung 2
….

10. Qualitätsszenarien
10.1 Qualitätsbaum
10.2 Qualitäts-/Bewertungsszenarien

11. Risiken

12. Glossar

Zusammenfassung

- Stellen Sie sicher, dass Sie als Ausgangspunkt für Architekturentscheidungen die architekturrelevanten Anforderungen kennen

- Dazu zählen
  - Scope/Kontextabgrenzung:
    - Was ist Ihr Thema?
    - Welche externen Schnittstellen müssen Sie berücksichtigen?
  - Stakeholder
    - Insbesonders um deren Erwartungshaltung bezüglich Architekturdokumentation zu kennen
  - Die wichtigen Qualitätsanforderungen/-ziele an die Lösung

- Vorsicht: Viele Qualitätsmerkmale beeinflussen sich gegenseitig.