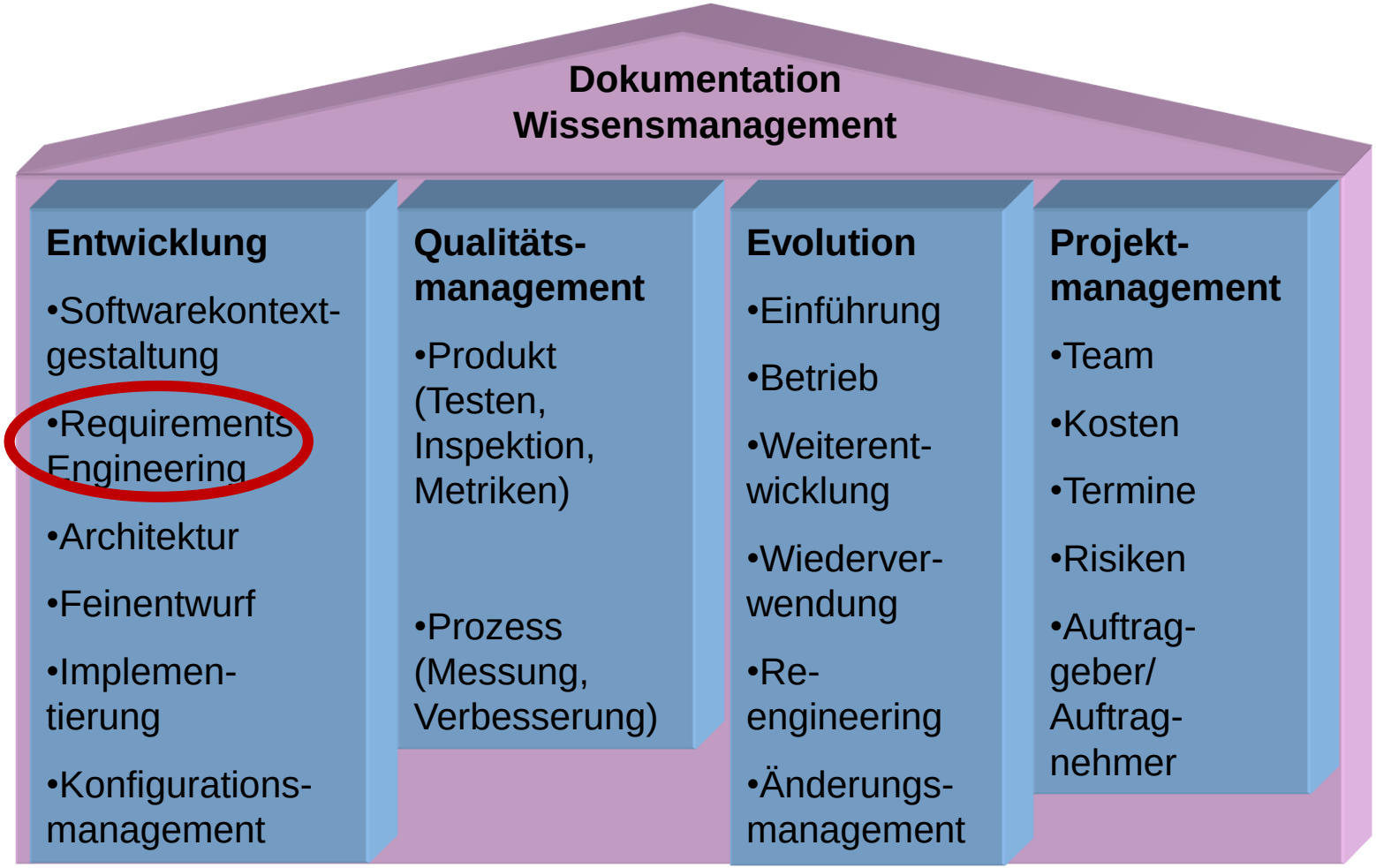
Einführung in Software Engineering

Barbara Paech, Marcus Seiler

Institute of Computer Science
Im Neuenheimer Feld 326
69120 Heidelberg, Germany
http://se.ifi.uni-heidelberg.de
paech@informatik.uni-heidelberg.de

Kapitel 5: SWE im Großen – Ergänzungen zu den Anforderungen

Aufgabenbereiche des Software Engineering



5. Ergänzungen zu Anforderungen (2. Teil)

Bei SWE im Großen ist es wichtig, noch ausführlicher (im Vergleich zu SWE im Kleinen) Anforderungen zu erfassen und deren Qualität zu sichern. Dazu können die folgenden Techniken eingesetzt werden.

- 5.1. Gebrauchstauglichkeit (Usability)
- 5.2. Qualitätssicherung mit KundInnen
- 5.3. Use Cases
- 5.4 Qualitätsanforderungen (inkl. Test)
- 5.5. Anforderungserfassung und - spezifikation

5.4. Qualitäts-anforderungen

- Motivation
- Qualitätsattribute
- QR-Beschreibung
- QR-Test

Motivation Qualitätsanforderungen

- Funktionale Anforderungen beschreiben „WAS“ das System tun soll. Sind gut durch Aufgaben, Use Cases, GUI und Funktionen zu erfassen und zu beschreiben.

- Nicht-funktionale Anforderungen beschreiben „WIE GUT“ das System das tun soll. Sind schwer zu formulieren
  - Subjektiv
  - Oft vage

  Werden oft zu spät konkretisiert

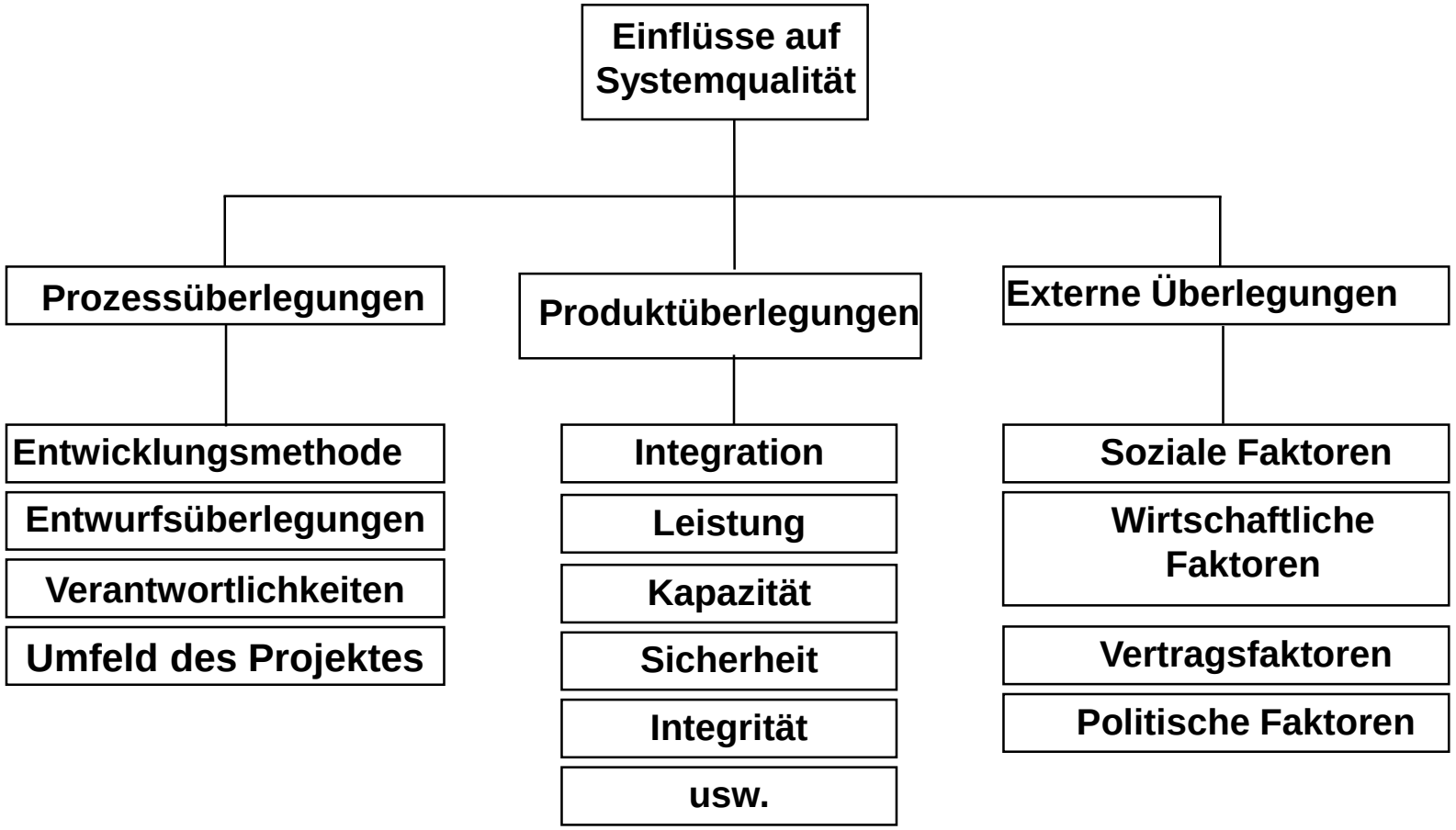
Begriff NFR

- Nicht- funktionale Anforderungen (NFR) sind alle Anforderungen, die die Qualität des Systems beeinflussen und nicht ein bestimmtes Verhalten (Funktion, Daten) beschreiben. Oft auch Qualitätsanforderungen genannt (hier nur Teilmenge, siehe später).

- Abgrenzung zu funktionalen Anforderungen (FR) aber teilweise unklar, z.B.
  - Sicherheit ist ein Qualitätsaspekt
  - „Das System soll keine Angriffe von außen zulassen“ ist ein NFR
  - „Bei Aufruf der Funktion XY muss sich der Nutzer autorisieren“ ist ein FR

[Locucopoulos/Karakostas 95]

Übersicht: Welche NFR gibt es ?



Externe Überlegungen und Prozessüberlegungen werden oft als Randbedingungen bezeichnet

Beispiele für NFR

- Prozessvorgabe
  - Zur Beschreibung des Systems sollen UML-Modelle verwendet werden
- Entwurfsvorgabe
  - Das System soll mit Web-Service-Technologie umgesetzt werden
- Politische Faktoren (gesetzliche Regelungen)
  - Das System soll die folgende TÜV-Regelung erfüllen…
- Produktüberlegungen (Qualität)
  - Die Antwortzeit des Systems soll ….sec bei voller Last nicht überschreiten

Qualitätsanforderungen

- Qualitätsanforderungen (Quality Requirement, QR) beschreiben die Produktüberlegungen.

- Sie sind oft durch Qualitätsattribute kategorisiert. Dazu gibt es Checklisten, die die wichtigsten Qualitätsattribute auflisten.

5.4. Qualitäts-anforderungen

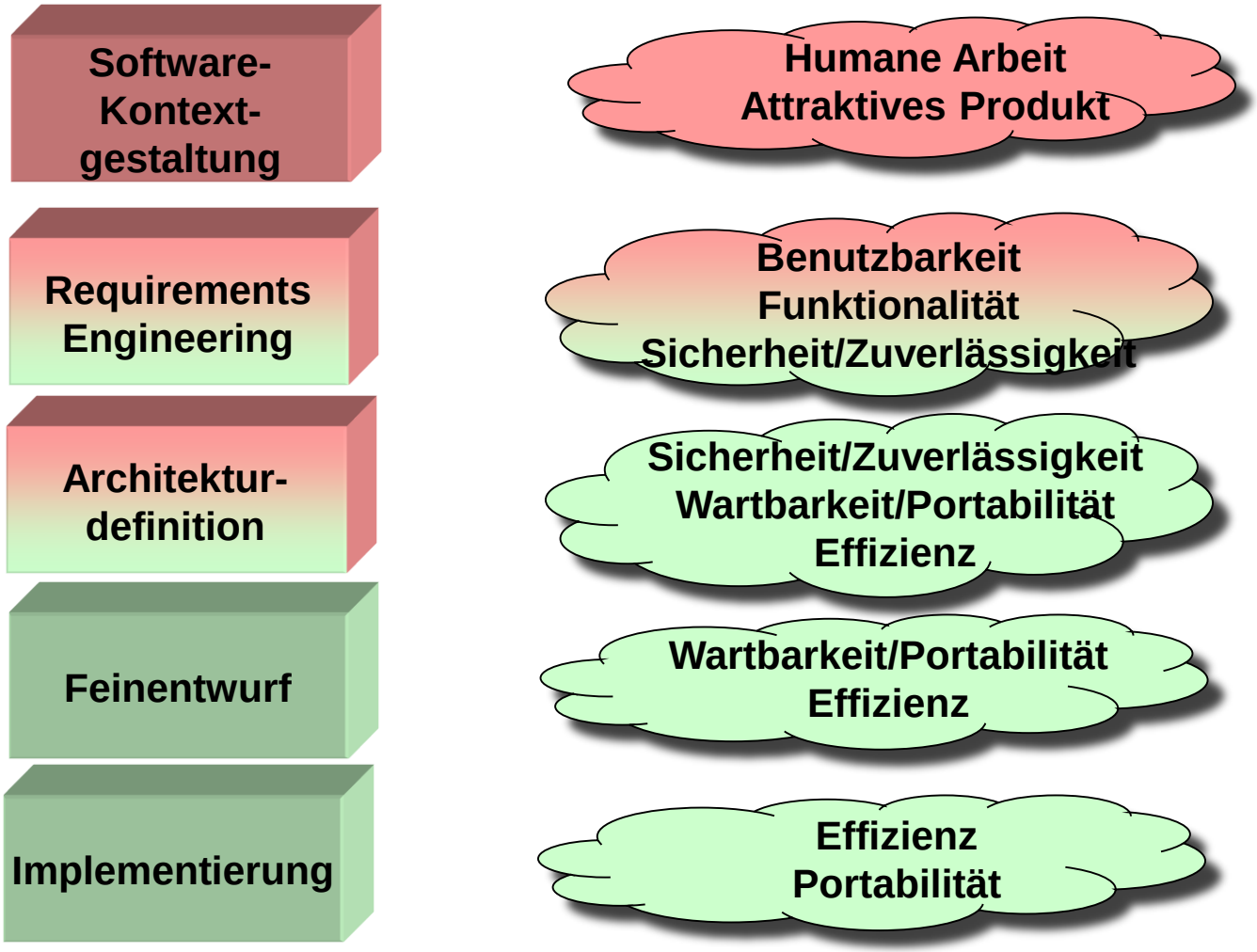
- Motivation
- Qualitätsattribute
- QR-Beschreibung
- QR-Test

Qualitätsattribute

- Qualitätsattribute (QA) strukturieren die Antwort auf die Frage: „Wie gut muss das durch die funktionalen Anforderungen beschriebene Produkt sein?“ Sie geben verschiedene Arten von Softwarequalität vor.

- Oft Konflikte zwischen mehreren QA, z.B. erhöht Kapselung die Wartbarkeit, verringert aber dafür oft die Effizienz.

SW-Produktqualität nach ISO/IEC 25010:2011



Wdh. Software Qualitäten

Standardisierte QA

- ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models

- **Quality in use (Impact of software on users)**
  - Direkte Validierung mit den NutzerInnen
  - Effectiveness, Efficiency, Satisfaction, Freedom from Risk, Context Coverage

- **Software Product Quality**
  - Validierung (external) und Verifikation (internal) während der gesamten Entwicklung
  - Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability, Portability

Quality in Use

- Effectiveness (wie gut kann Aufgabenziel erreicht werden)
- Efficiency (wie effizient kann Aufgabenziel erreicht werden)
- Satisfaction
  - Usefulness (Wahrnehmung der Nützlichkeit), Trust, Pleasure, Comfort
- Freedom from risk
  - Economic risk mitigation, Health and safety risk mitigation, Environmental risk mitigation
- Context coverage
  - Context completeness (in Bezug auf spezifizierte Kontexte)
  - Flexibility (Anpassung an weitere Kontexte)

Software Product Quality

- Functional suitability
  - Functional completeness (vollständig), Functional correctness (korrekt), Functional appropriateness (hilfreich)
- Performance efficiency
  - Time behaviour, Resource utilization, Capacity
- Compatibility (Kompatibilität)
  - Co-existence (mit anderer SW/HW in gleicher Umgebung), Interoperability
- Usability
  - Appropriateness recognizability (Verständnis der Features, z.B. Dokumentation), Learnability, Operability (steuerbar), User error protection, User interface aesthetics, Accessibility (auch spezielle NutzerInnen)
- Reliability (Zuverlässigkeit)
  - Maturity (Reife, Verhalten im Normalbetrieb), Availability (verfügbar), Fault tolerance, Recoverability (wiederherstellbar)
- Security (Sicherheit, Schutz von Information und Daten)
  - Confidentiality (nur authorisierter Zugang), Integrity (nur authorisierte Änderungen), Non-repudiation (Ereignisse nachweisbar), Accountability (Ereignisse zuweisbar), Authenticity (Identität zuweisbar)
- Maintainability (Wartbarkeit)
  - Modularity, Reusability, Analysability, Modifiability, Testability
- Portability
  - Adaptability, Installability, Replaceability

Konkretisierung von QA

- QA werden durch Anforderungen konkretisiert
  - Qualitätsanforderung (QR) oder Verhaltensanforderung (FR) oder Entwurfsanforderung

- Konkrete QR muss messbar sein
  - Hilfreich sind konkrete Szenarien (z.B. Lastszenarien, Sicherheitsangriffe)
  - Konkrete Werte müssen aufgrund von Kosten/ Nutzenabwägungen festgelegt werden
    - Sind oft stark beeinflusst von externen Faktoren
      z.B. wie schnell muss das System reagieren, damit es für die NutzerInnen noch akzeptabel ist, oder wie wichtig ist welche Effizienz für den Markterfolg

[nach Lauesen 2002]

Konkrete QR zu QA Functional Correctness

- Messbare Aspekte von Functional Correctness sind z.B. Genauigkeit von Intervallgrenzen und Berechnungs-genauigkeit

- Konkrete Beispiele
  - Namensfeld soll 150 Zeichen umfassen
  - Buchungsdatum darf bis zu 2 Jahre das aktuelle Datum übersteigen
  - Sensordaten sollen mit 14 Bit Genauigkeit abgespeichert werden
  - Spracherkennung auch bei Hintergrundgeräuschen der Lautstärke...

[nach Lauesen 2002]

Konkrete QR zu QA Performance Efficiency

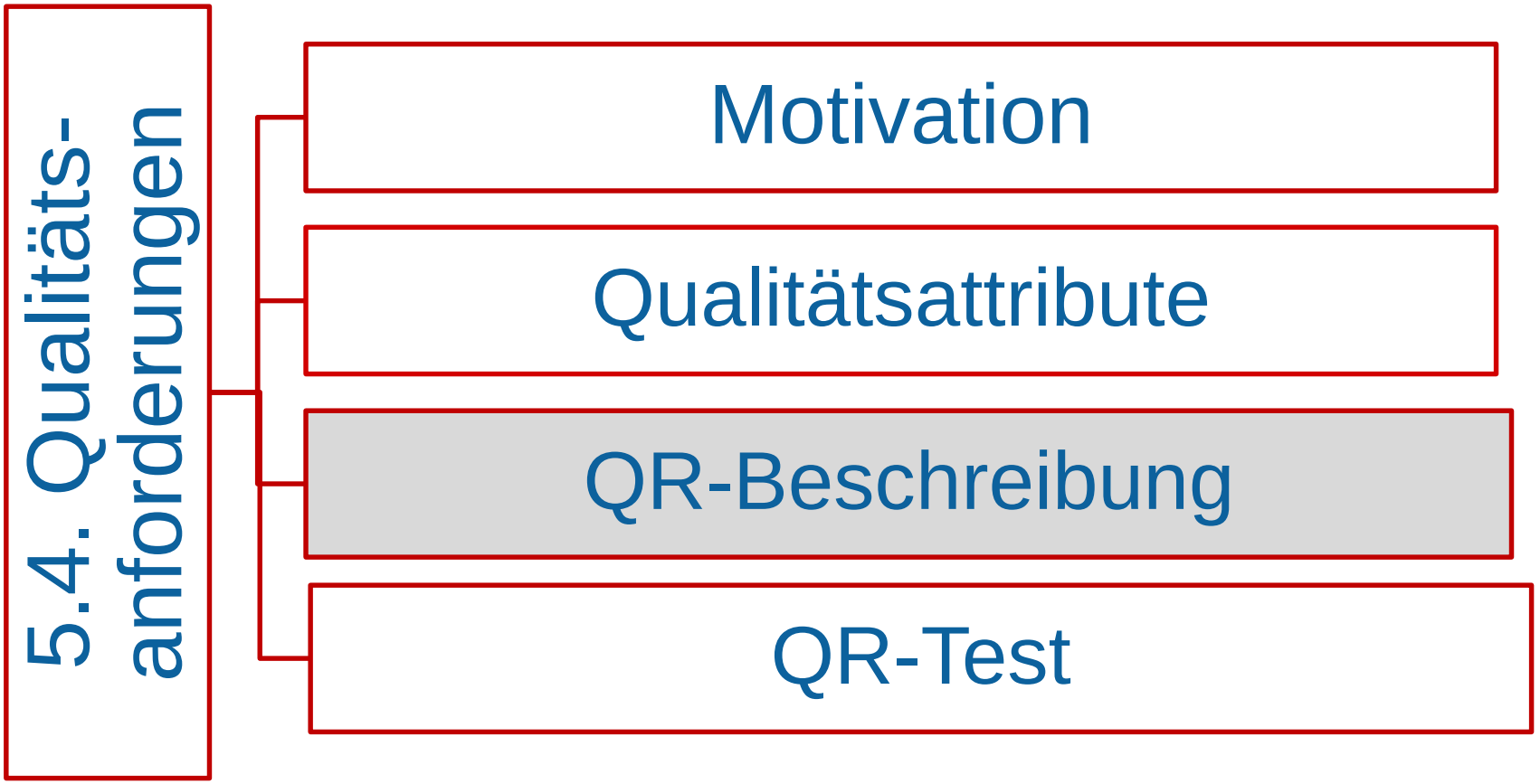
- Verbrauchsverhalten
  - Vor allem Volumenangaben, z.B.
    - Max 16 MB Speicher
    - Bis zu 2000 gleichzeitige Nutzer
    - Bis zu 10.000 Kundeneinträge, mit 20% Zuwachs pro Jahr

- Zeitverhalten
  - Angaben zu Systemreaktionszeit (Aufgabenreaktionszeit in Bezug zu Benutzungsqualität)
  - Angabe von Durchschnittszeit und Maximum, oft unter Angaben bestimmter Wahrscheinlichkeiten
  - Setzt Annahmen über Häufigkeit der Nutzereingaben (Systemlast) und Zeitverhalten der Hardware voraus
  - Konkrete Beispiele:
    - 100 Zahlungstransaktionen pro Sekunde unter höchster Systemlast
    - CPU Verbrauch unter Standardlast bis zu 50% (Rest für Batch-Jobs)
    - Keyword-Suche soll max. 5 Sekunden dauern
    - Blättern in 200 Seiten Dokumente soll max. 1 Sekunde dauern

Übung: Beispiele für QR

1. Das System muss auf Mac, Windows, Linux-derivaten laufen.
2. Der Quelltext ist kommentiert (50% ohne Leertexten, java-doc)
3. Das System soll n Stunden, k Benutzer, ohne schwerwiegende Fehler (kein Absturz von Y) laufen
4. Bei der Änderung vom Rückgabedatum soll beim Nutzer nachgefragt werden
5. Das Ergebnis jeder mathematischen Berechnung soll mindestens auf 3.Nachkommastelle genau sein.
6. „Entwurfsmuster sind zu benutzen“

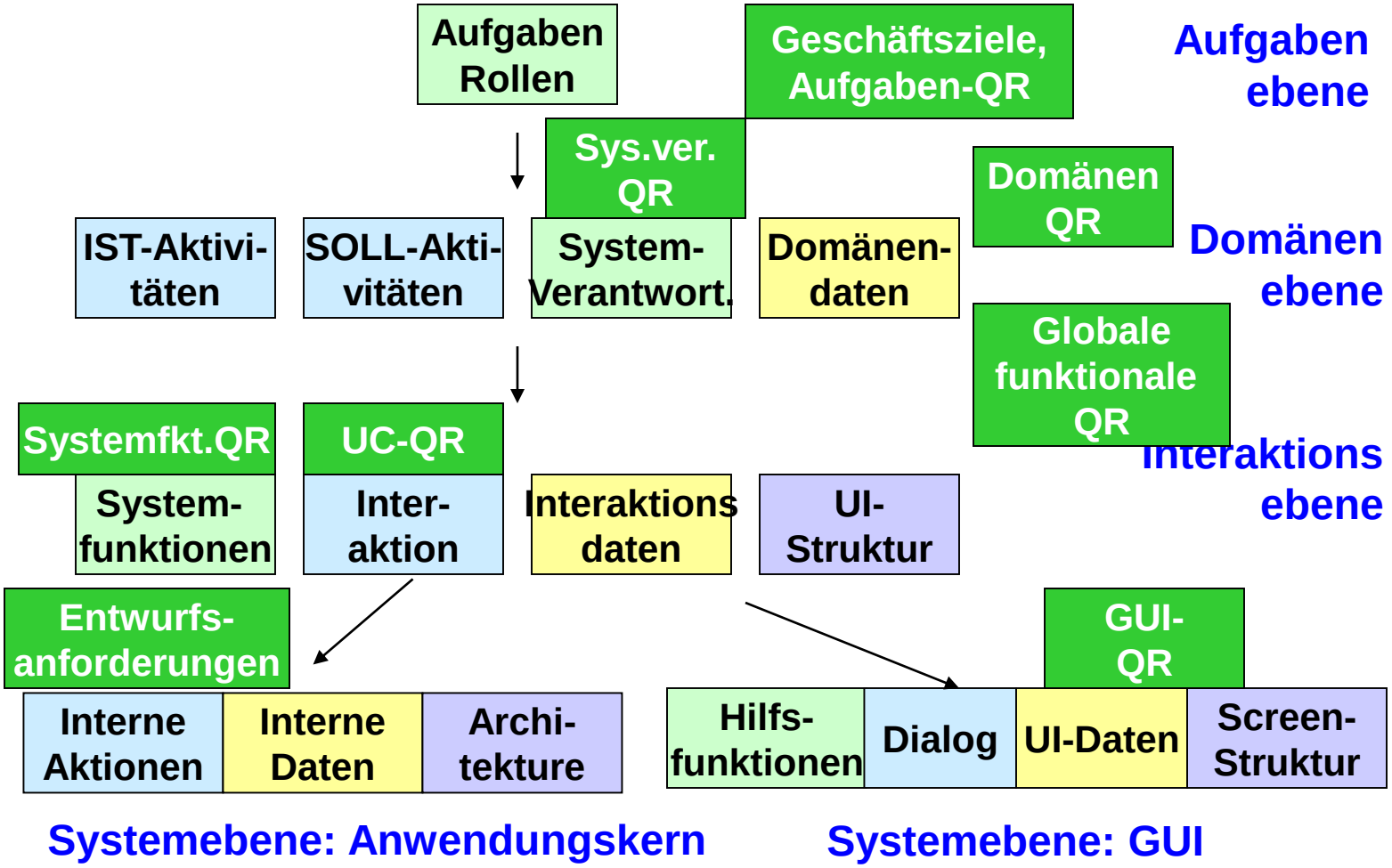
Messbar?
Welche QA?



Wie beschreibe ich QR allgemein?

- QR können oft funktionalen Elementen zugeordnet werden, z.B. QR für eine konkrete Systemfunktionen
- Andere QR sind übergreifend (gelten für das ganze System)
- QR entstehen oft aus geschäftlichen Überlegungen

Beschreibungsebenen für QR



Zuordnung zu den Ebenen (1)

- Geschäftsziele (Business Goal)
  - *Welchen Nutzen soll das System für die KundInnen bringen?*
  - Z.B. Filmverwaltung: mindestens 50% der Filme sollen ausgeliehen sein

- Qualitätsanforderungen an die Aufgabe
  - *Welche Qualität soll die Durchführung einer bestimmten Aufgabe haben?*
  - Ergeben sich aus dem Aufgabenkontext
  - z.B. Unteraufgabe Bewertung: Bewertung für 50 Filme darf max. 5 Minuten dauern

Zuordnung zu den Ebenen (2)

- Domänenfaktoren
  - *Welche Faktoren sind in der Domäne geregelt (in Bezug auf die Aufgaben)?*
  - Ergeben sich aus der Anwendungsdomäne
  - z.B. Filmverwaltung: einzuhaltende Rechte an bestimmten Bilder von Filmen

- Globale funktionale Anforderungen
  - *Übergreifende funktionale Anforderungen, die sich auf das System beziehen und die noch in einzelne FR (oder QR zu bestimmten UC oder Funktionen) zerlegt werden müssen*
  - z.B. Filmverwaltung: Ausleihdaten dürfen nicht von Unbefugten einsehbar sein

Zuordnung zu den Ebenen (3)

- Qualitätsanforderungen an die Systemunterstützung (Systemverantwortlichkeiten, Use Cases und Systemfunktionen)
  - *Anforderung an bestimmte SysVer, UC, SysFkt*
  - Z.B. Filmverwaltung: Beim UC “Rückgabe” dürfen keine Daten verloren gehen
- Qualitätsanforderungen an das GUI
  - *Anforderung an Dialog und Layout*
  - Z.B. Filmverwaltung: In einem Dialog dürfen bei einer Löschaktion Daten nicht aus Versehen gelöscht werden.
- Entwurfsanforderungen
  - *Anforderung an die interne Systemstruktur*
  - Z.B. Filmverwaltung: es darf keine zusätzliche Hardware verwendet werden

Beispiel Sicherheit (1)

QR zu dem Thema Sicherheit auf allen Ebenen

- Aufgaben-QR: Aufgabe X ist sicherheitskritisch, d.h. Informationen über Ergebnisse dürfen nicht nach außen gelangen und sie darf nur von besonders geschulten Personen durchgeführt werden (*diese Aussage vererbt sich auf die Systemunterstützung!)*

Beispiel Sicherheit (2)

QR zu dem Thema Sicherheit auf allen Ebenen

- Domänen-QR: An Aufgabe X sind immer mind. 5 Rollen beteiligt, von denen nur Rolle1 und Rolle2 Aktion A ausführen dürfen.

- Domänen-QR oder Globale funktionale QR (je nachdem, ob eher auf Domäne oder System bezogen):
  Datenschutzrichtlinien sind einzuhalten

Beispiel Sicherheit (3)

QR zu dem Thema Sicherheit auf allen Ebenen

- UC-QR: UC Y ist sicherheitskritisch, d.h. Informationen über Ergebnisse dürfen nicht nach außen gelangen
- Funktions-QR: Funktion Z darf keinen Zugang zu den Daten ABC erhalten
- GUI-QR: Auf dem Bildschirm dürfen Benutzerdaten nur anonymisiert dargestellt werden (z.B. Passwort durch ***)
- Entwurf-QR: Die Verbindung zwischen Komponente X und Y muss abhörsicher sein.

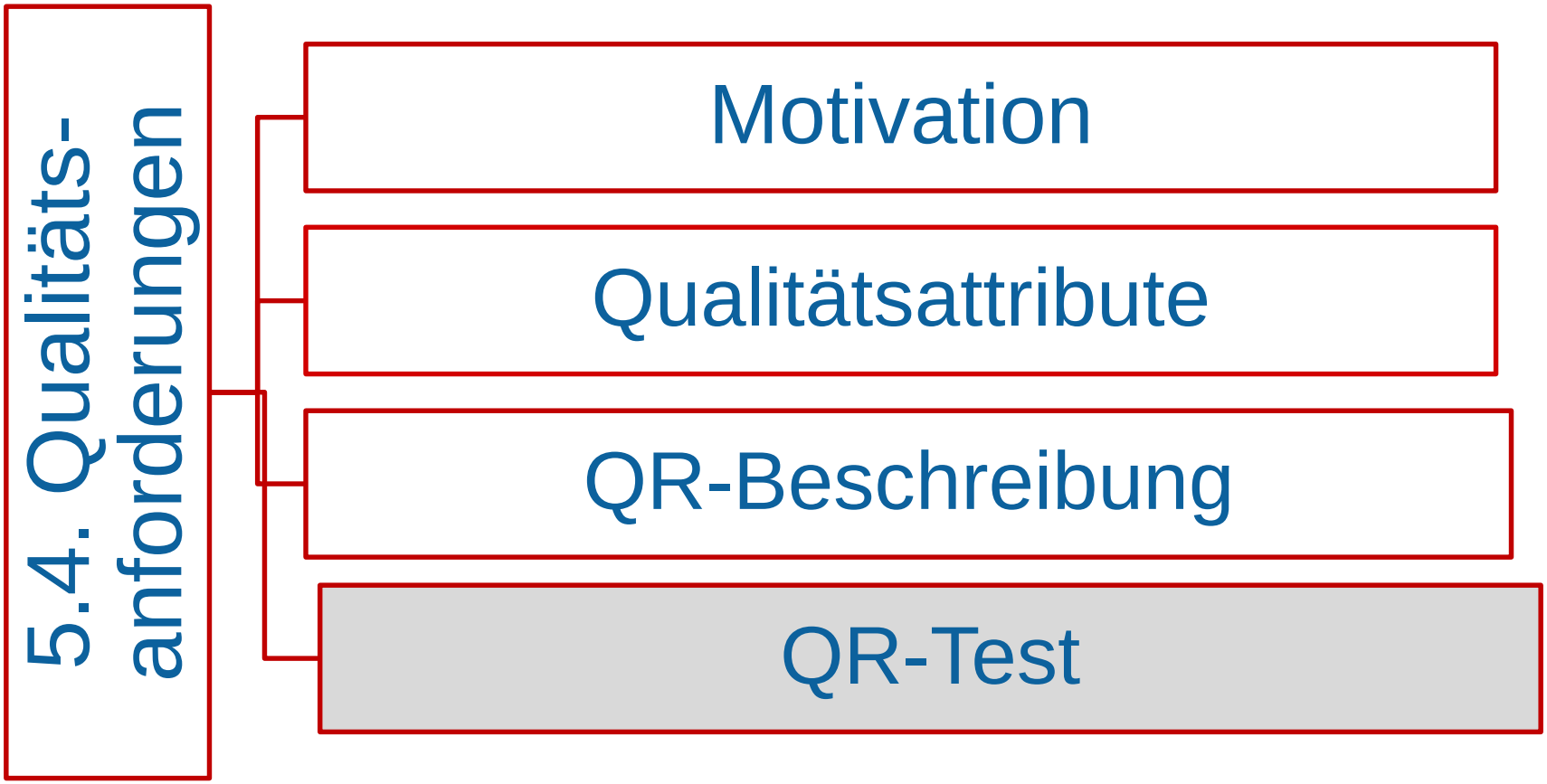
QR und Rationale

- QR sind oft die Kriterien für die Gestaltung der FR

QR schränkt FR ein	Auf-gaben/ Sys.-ver-QR	Domä-nen-QR	Glo-bale QR	UC-QR	Funk-tions-QR	GUI-QR	Entwurfs-QR
Aufgaben	x	x					x(Sys.ver)
UC	x	x	x	x			x
Funk-tionen	x	x	x	x	x		x
GUI	x	x	x	x	x	x	x
Architek-tur	x	x	x	x	x		x

Vorgehen Beschreibung QR

- Die QR sollten gleichzeitig mit den FR so detailliert wie möglich beschrieben werden.
- Z.B. ausgehend von den Geschäftszielen systematisch alle Arten von QA durchgehen und für jede Ebene entsprechende QR formulieren
- Es existieren viele weitere systematische Wege, z.B. über Risikobetrachtungen (definiere wichtige Elemente (Assets) auf allen Ebenen und überlege, was deren wichtigste Qualität ist und durch welches Risiko sie beeinträchtigt sein kann). Dann werden QR als Gegenmaßnahmen zu den Risiken definiert.



Test von QR

- Auch die Erfüllung von Qualitätsanforderungen muss getestet werden.
- Je nach Qualitätsattribut sind unterschiedliche Testmethoden nötig
- Meist erst auf Ebene des Systemtest möglich, insbesondere auch Abnahmetest (wegen der Produktivumgebung)
- Meistens Durchspielen von Szenarien

Szenarien für Normalbetrieb

- Interoperabilität (Koexistenz)
  - => Kompatibilitätstest (Datenaustausch mit vorhandenen Systemen)
- Wartbarkeit
  - => Änderungstest, Dokumentationstest (Durchspielen von Änderungen)
- Übertragbarkeit
  - => Anpassbarkeit: Durchspielen von Anpassungen
  - => Installierbarkeit
    - => Konfigurationstest (z.B. verschiedene. Sprachen, Betriebssysteme)
  - => Austauschbarkeit: Durchführen des Austauschs von Komponenten
- Wiederherstellbarkeit (Zuverlässigkeit)
  - => Durchspielen der Wiederherstellung
- Usability
  - => Siehe Nutzungstest

Szenarien für besondere Situationen

- Sicherheit
  - => Sicherheitstest (z.B. Durchspielen von Hackerangriffen)
- Zuverlässigkeit
  - Reife und Verfügbarkeit
    - =>Stabilität/Zuverlässigkeitstest (Dauerbetrieb)
  - Fehlertoleranz
    - => Stresstest (Überlastungssituation herbeiführen), Robustheitstest (gegenüber Fehlbedienung, HW-Ausfall...)
- Performanz
  - => Lasttest (Szenarien mit hoher Anzahl der Anwender, Transaktionen) und Volumen/Massentest (Vorgabe von großen Datenmengen)

Zusammenfassung NFR

- NFR umfassen Prozessanforderungen, Qualitätsanforderungen (QR) und externe Faktoren
- QR werden durch Qualitätsattribute (QA) kategorisiert
- QR sind messbar und auf allen Ebenen zu beschreiben
- QR auf hoher Ebene liefern oft eine Begründung für QR auf niedriger Ebene

- Test von QR unterschiedlich ja nach QA
- Oft sehr aufwändig, weil reale Bedingungen herzustellen sind, um Qualität zu überprüfen

Literatur NFR

- L Chung, BA Nixon, E Yu, J Mylopoulos (2000) “Non-Functional Requirements in Software Engineering”, Kluwer Academic Publishers
- S. Lauesen (2002) Requirements Engineering, Addison-Wesley
- P. Loucopoulos, V. Karakostas (1995) System Requirements Engineering, Mc-Graw.Hill

5.5. Anforderungs-erhebung und -spezifikation

- Einleitung
- Anforderungserhebung
- Anforderungsspezifikation

Anforderungserhebung- und spezifikation

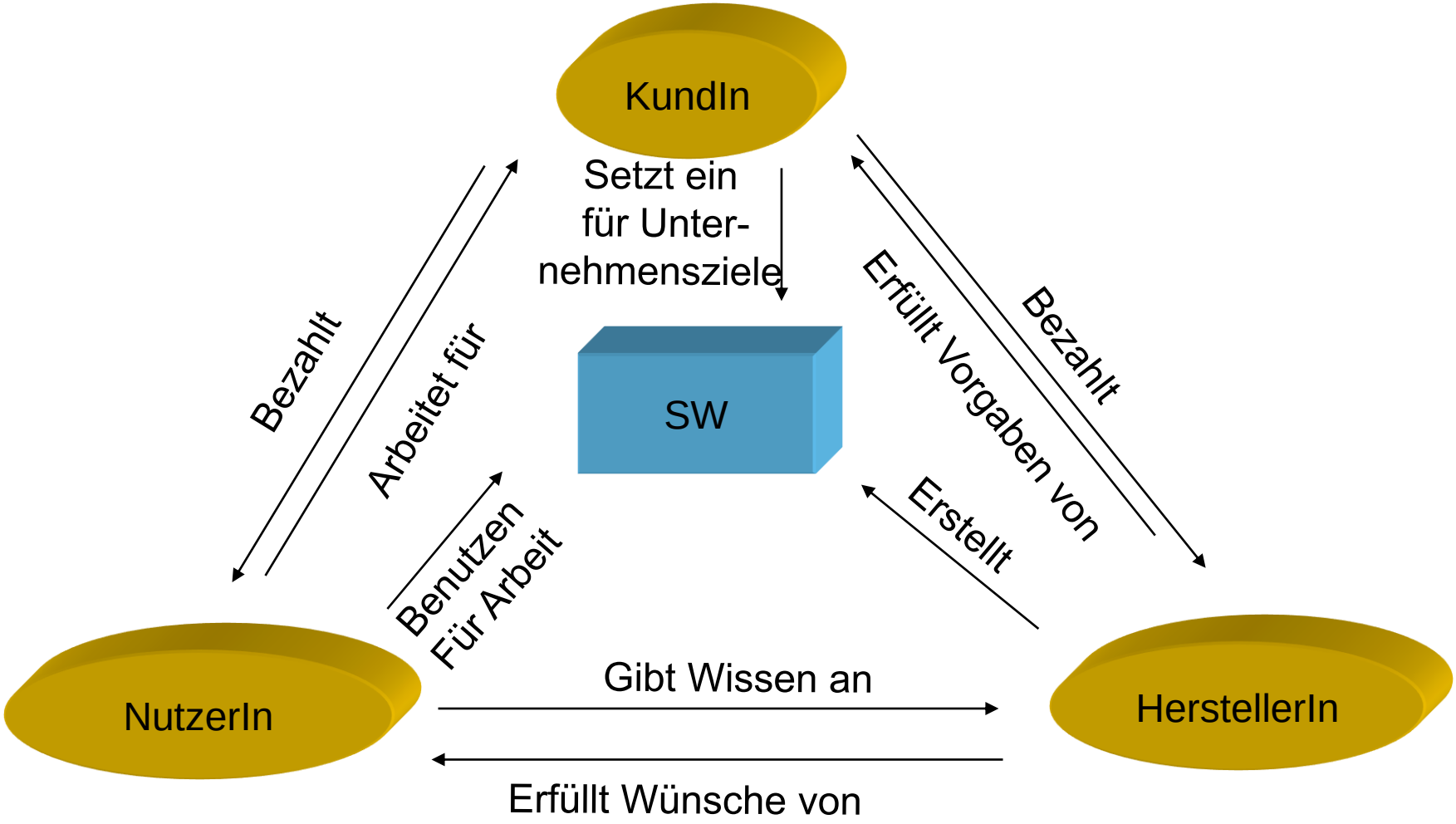
- Die Beschreibung der Anforderungen (Anforderungsspezifikation) reicht nicht aus.
- Anforderungen müssen zunächst erhoben werden und dann über die Zeit gepflegt werden (Management der Dokumente)
- => Gesamtprozess Requirements Engineering nötig

[Pohl,Rupp 2010]

Was ist Requirements Engineering?

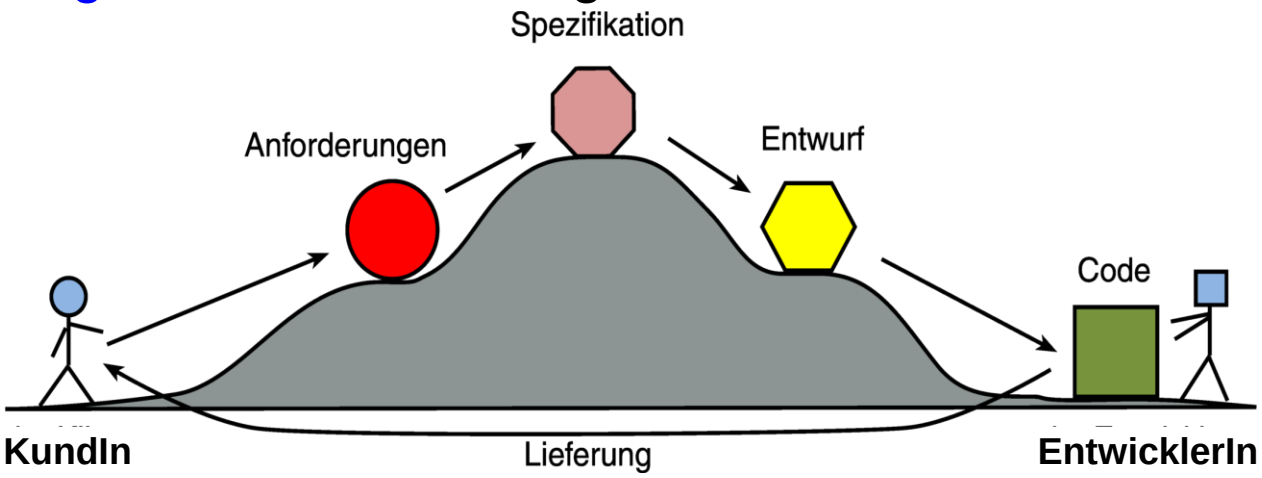
- Das Requirements Engineering ist ein kooperativer, inkrementeller Prozess, dessen Ziel es ist zu gewährleisten, dass
  - Alle relevanten Anforderungen bekannt sind und in dem erforderlichen Detaillierungsgrad verstanden sind
  - Die involvierten Stakeholder eine ausreichende Übereinstimmung über die bekannten Anforderungen erzielen
  - Alle Anforderungen konform zu den Dokumentationsvorschriften bzw. konform zu den Spezifikationsvorschriften spezifiziert sind.

Wdh. Folien 04 Beteiligte beim SWE



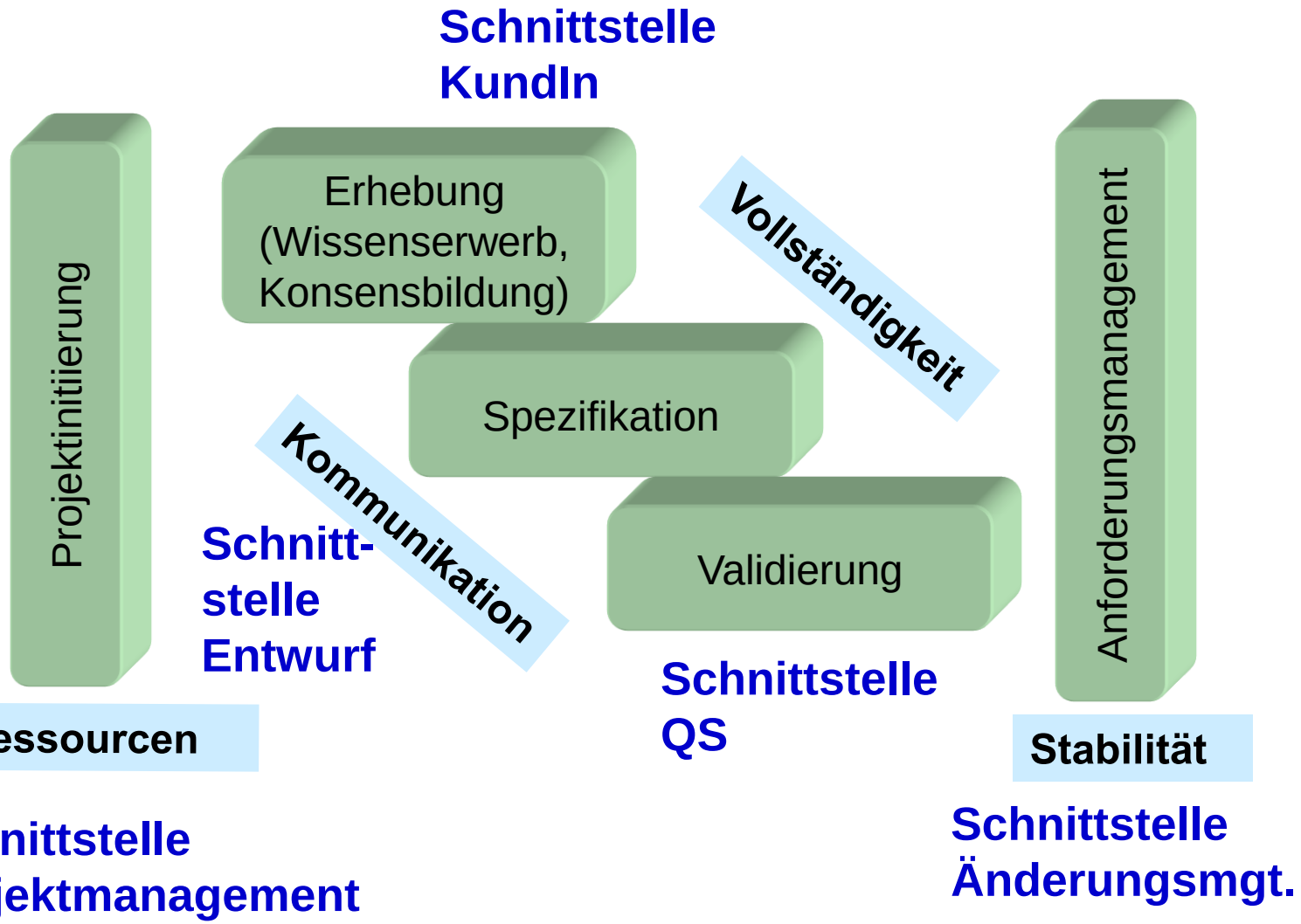
Wdh Folie 05: Anforderungsaktivitäten im Überblick

- Die Anforderungen werden erhoben, in der Spezifikation formuliert, geprüft und anschließend in den Entwurf umgesetzt. Schließlich wird implementiert, auf verschiedenen Ebenen geprüft und korrigiert. Das Resultat geht zurück an die KundInnen.



- KundInnen bekommen nur dann, was sie haben wollen, wenn ihre Anforderungen sorgfältig erhoben und unterwegs nicht verfälscht wurden.

Was ist im RE zu tun?



5.5. Anforderungs-erhebung und -spezifikation

- Einleitung
- Anforderungserhebung
- Anforderungsspezifikation

Ziele der Anforderungserhebung

- Wissenserwerb
  - wer wird die Software wofür nutzen
  - welchen Vorteil soll die Software bringen
  - Randbedingungen der Entwicklung

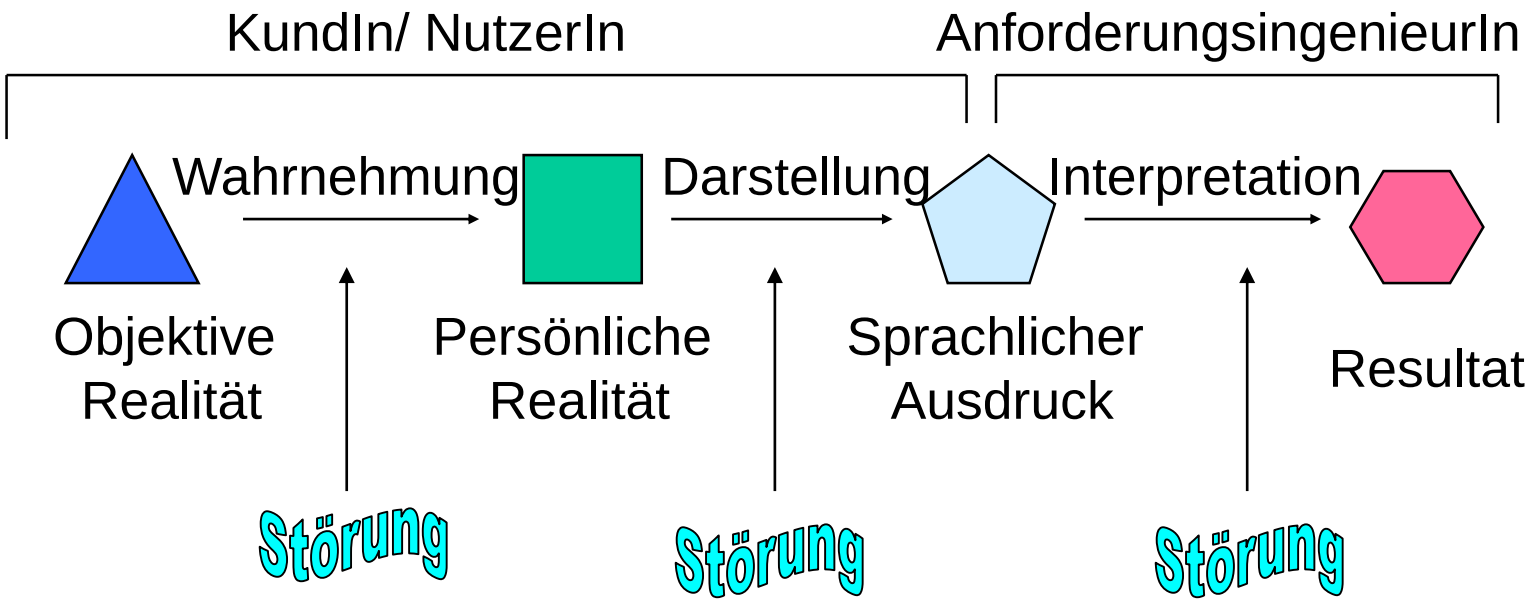
- Konsensbildung
  - zwischen verschiedenen Beteiligten
  - zwischen KundIn und NutzerIn
  - zwischen KundIn und HerstellerIn

Warum ist der Kontakt mit den NutzerInnen schwierig?

- Berücksichtigung **aller Betroffenen** (Stakeholder)
- Kommunikation
  - NutzerInnen können oft **nicht abstrakt** beschreiben, was sie tun und warum oder was sie brauchen
  - NutzerInnen haben **zu allgemeine** Wünsche
- Vorstellung neuer Möglichkeiten und ihrer Konsequenzen
  - NutzerInnen hängen an **alten** Verfahrensweisen
- Konflikte
  - aufgrund von **Machtkämpfen**
  - aufgrund von **Widerstand gegenüber Änderungen**
- Prioritäten
  - NutzerInnen wünschen sich **zu viel**
- Änderungen
  - NutzerInnen kommen **immer wieder** mit neuen Wünschen

Kommunikation

- Kommunikation erfolgt über Sprache
  - Repräsentation von Erfahrungen (Wahrnehmung)
  - Mitteilung der persönlichen Realität (Darstellung)



Welches Wissen ist für Anforderungen zu erfassen?

- **Bisherige Arbeitsweisen**
  - z.B. Notenverwaltung: Durchführung der Bewertung
- **Probleme damit**
  - z.B. Notenverwaltung: In excel-sheets müssen immer alle TN und LV neu eingegeben werden
- **Ziele für neue Arbeitsweise / Systeme**
  - z.B. Notenverwaltung: Alle Lehredaten zentral zur Verfügung stellen
- **Erfolgsfaktoren dafür**
  - z.B. Notenverwaltung: Aufwand für Eingabe verringert
- **Grobe Systemarchitektur (wie viele Komponenten, Verteiltheit)**
  - z.B. Datenbank mit Zugriff über das Web und lokal
- **Realistische Lösungen**
  - z.B. Notenverwaltung: erstmal alle LV zentral eingeben (nicht auch TN)
- **Konsequenzen und Risiken**
  - z.B. Sicherheit der zentralen DB problematisch

[Macaulay]

Kenntnisse von NutzerIn und EntwicklerIn

Wissensgebiete	***Abstraktes Wissen***	***Konkrete Erfahrung***
Derzeitige Arbeit des/der NutzerIn	**1** Sachbezogene Struktur der derzeitigen Arbeit des/der NutzerIn *Brauchen NutzerInnen und EntwicklerInnen*	**4** Konkrete Erfahrung mit der derzeitigen Arbeit des/der NutzerIn *Haben NutzerInnen; brauchen EntwicklerInnen*
Neue Systeme	**2** Ideen- und Entwurfvorstellungen *Brauchen NutzerInnen und EntwicklerInnen*	**5** Konkrete Erfahrung mit einem neuen System *Brauchen NenutzerInnen*
Technische Auswahl-möglichkeiten	**3** Übersicht technologischer Möglichkeiten *Brauchen EntwicklerInnen*	**6** Konkrete Erfahrung mit technologischen Möglichkeiten *Haben EntwicklerInnen; brauchen NutzerInnen*

[Macaulay]

Wie erhebe ich Anforderungen?

Werkzeuge und Techniken für die Entwicklung von Wissen	Abstrakt			Konkret		
	1	2	3	4	5	6
Beobachtungen				●		
Benutzerbefragung	●			●		
Entwickler verrichten Benutzerarbeit				●		
Videoaufnahmen				●		
Bildschirmskizzen				●		
Laut-Denken Experimente				●	●	
Rich-Pictures	●			●	●	
Ethnographische Studien	●			●		
Use Cases	●	●		●		
Zukunftswerkstätten	●	●			●	
Objektorientierte Analyse	●	●		●		
Ereignislisten	●	●				
Entity-Relationship-Diagramme	●	●				
Konzeptuelle Modellierung	●					
Datenfluss-Diagramm	●	●				
Formale Spezifikation	●					
Prototypen		●			●	●
Besuche anderer Anlagen			●			●
Literaturstudium			●			
Studium von Standard Software			●			●

1 = IST abstrakt
2 = SOLL abstrakt
3 = Technik abstrakt
4 = IST konkret
5 = SOLL konkret
6 = Technik konkret

5.5. Anforderungs-erhebung und -spezifikation

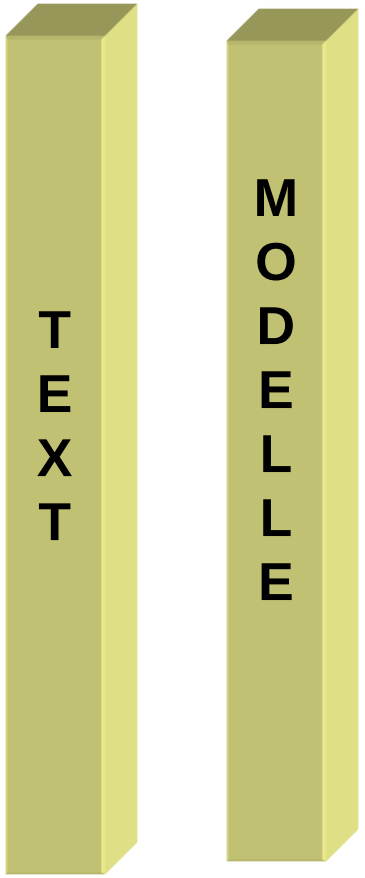
- Einleitung
- Anforderungserhebung
- Anforderungsspezifikation

Anforderungsspezifikation

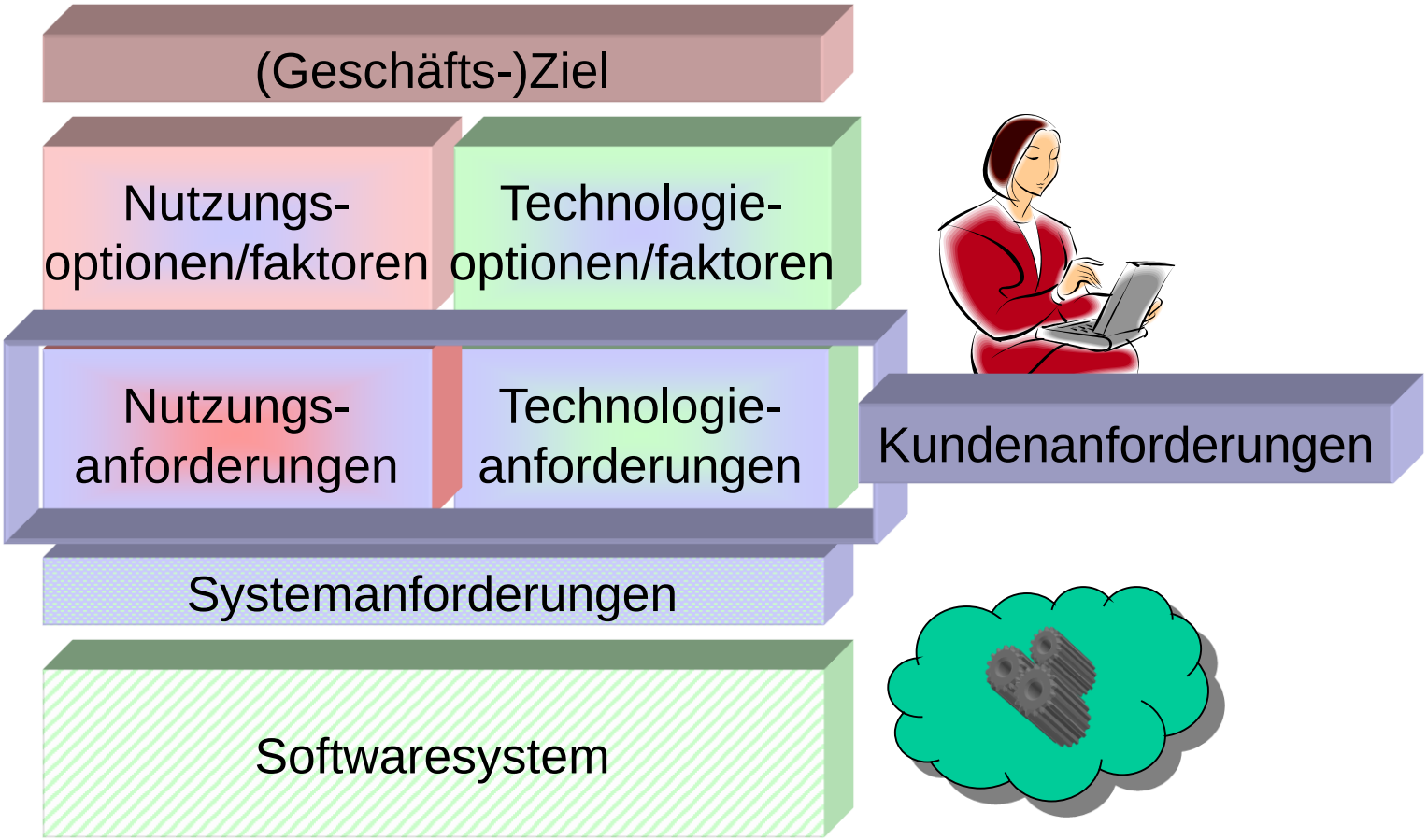
Funktionale Anforderungen:
Wie soll die Software die NutzerInnen unterstützen?

Nicht-funktionale Anforderungen:
1.Qualitätsanforderungen:
Wie gut soll die Unterstützung sein?
2. Randbedingungen

Rationale:
Wie sind die getroffenen Gestaltungsentscheidungen **begründet**?



Welche Inhalte umfassen Anforderungsdokumente?



Definition: Anforderung

ISO/IEC 29148-2011:

- **Stakeholder Requirements (Kundenanforderung):** Anforderung, die einE KundIn stellt bzw. benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen
  - z.B. Messung mit Elektroden kann für PatientInnen unangenehm sein. ÄrztInnen sollten deshalb immer beide Hände während der Messung an den Elektroden haben.
- **System Requirements (Systemanforderung):** Anforderung, die ein System erfüllen bzw. besitzen muss
  - z.B. ÄrztInnen müssen die Messung mit beiden Händen an den Elektroden starten bzw. beenden können.
- Typischerweise 2 unterschiedliche Dokumente!

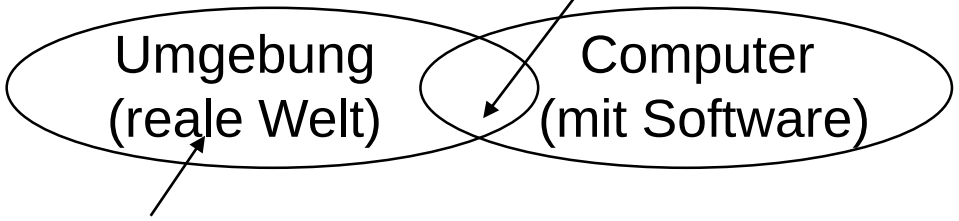
Klärung: Kundenanforderungen vs. Systemanforderungen

Annahmen, z.B.:

“Impulse durch Raddrehung, wenn Flugzeug am Boden.”

“Der Umkehrschub darf nur dann aktivierbar sein, wenn Radimpulse vorliegen.”

Systemanforderung (auch oft Entwickler-anforderung genannt)



Kundenanforderung (auch oft Nutzungsanforderung genannt), z.B.:

“Der Umkehrschub darf nur dann aktivierbar sein, wenn sich das Flugzeug am Boden befindet”

Klärung: Lastenheft

- Wird auch Anforderungsdefinition oder Fachkonzept genannt
- Zusammenstellung aller Anforderungen der/des Kundin/en hinsichtlich Liefer- und Leistungsumfang (VDI/VDE 3694)

- beschreibt die Kundenanforderungen
- beschreibt WAS und WOFÜR etwas zu erstellen ist

- wird oft von dem/der KundIn erstellt

- dient als Ausschreibungs-, Angebots- und Vertragsgrundlage

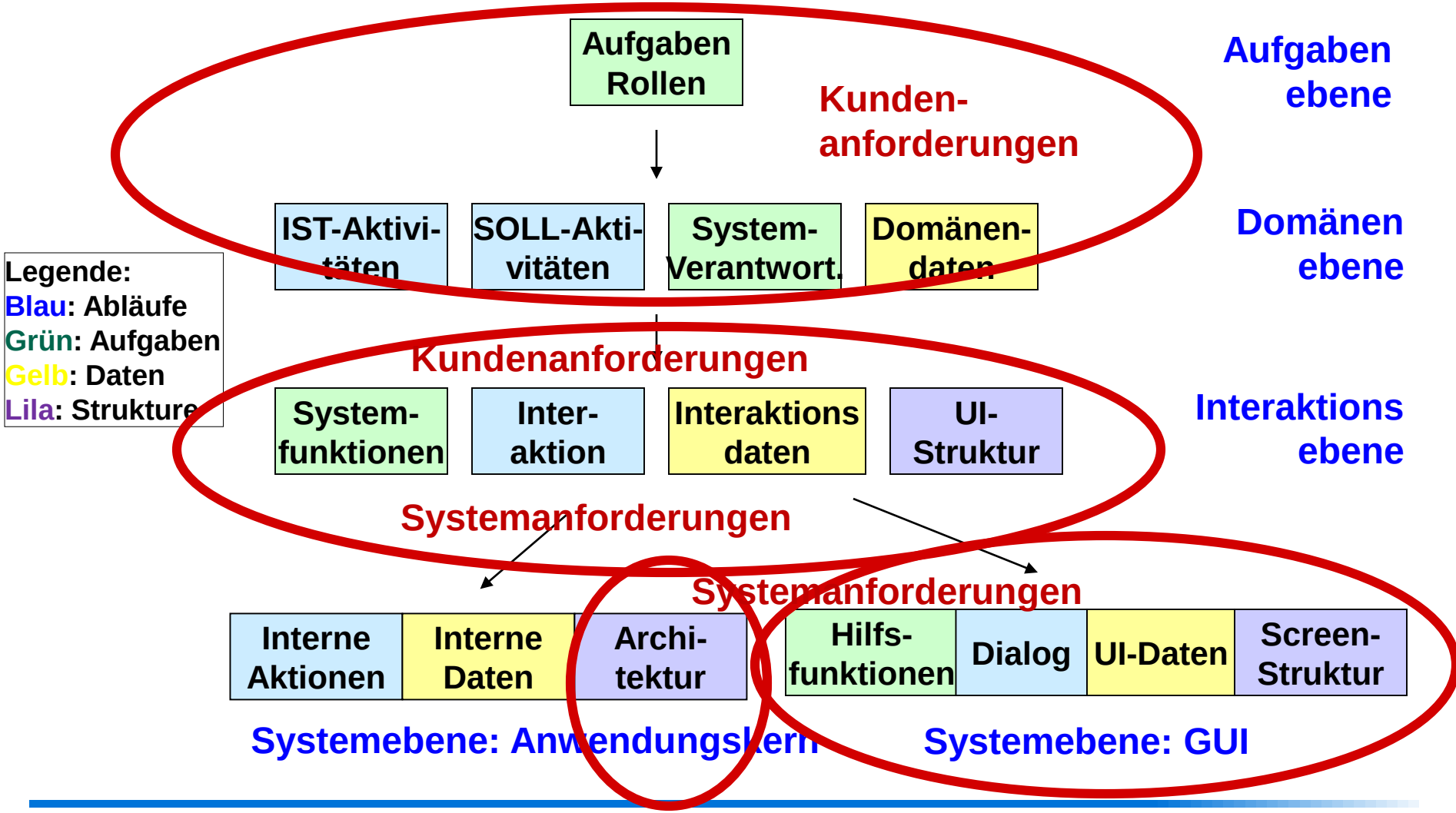
Klärung: Pflichtenheft

- Wird auch Softwarespezifikation oder DV-Konzept genannt
- Beschreibung der Realisierung aller Anforderungen des Lastenhefts (VDI/VDE 3694)

- enthält und detailliert das Lastenheft
- Beschreibt die Systemanforderungen
- beschreibt WIE und WOMIT Anforderungen zu realisieren sind

- wird i.d.R. nach Auftragsteilung von der/dem KundIn erstellt
- wird von der/dem KundIn genehmigt und dient dann als verbindliche Vereinbarung für die Realisierung

Zuordnung zu Beschreibungsebenen



Verwendung Anforderungsdokumente

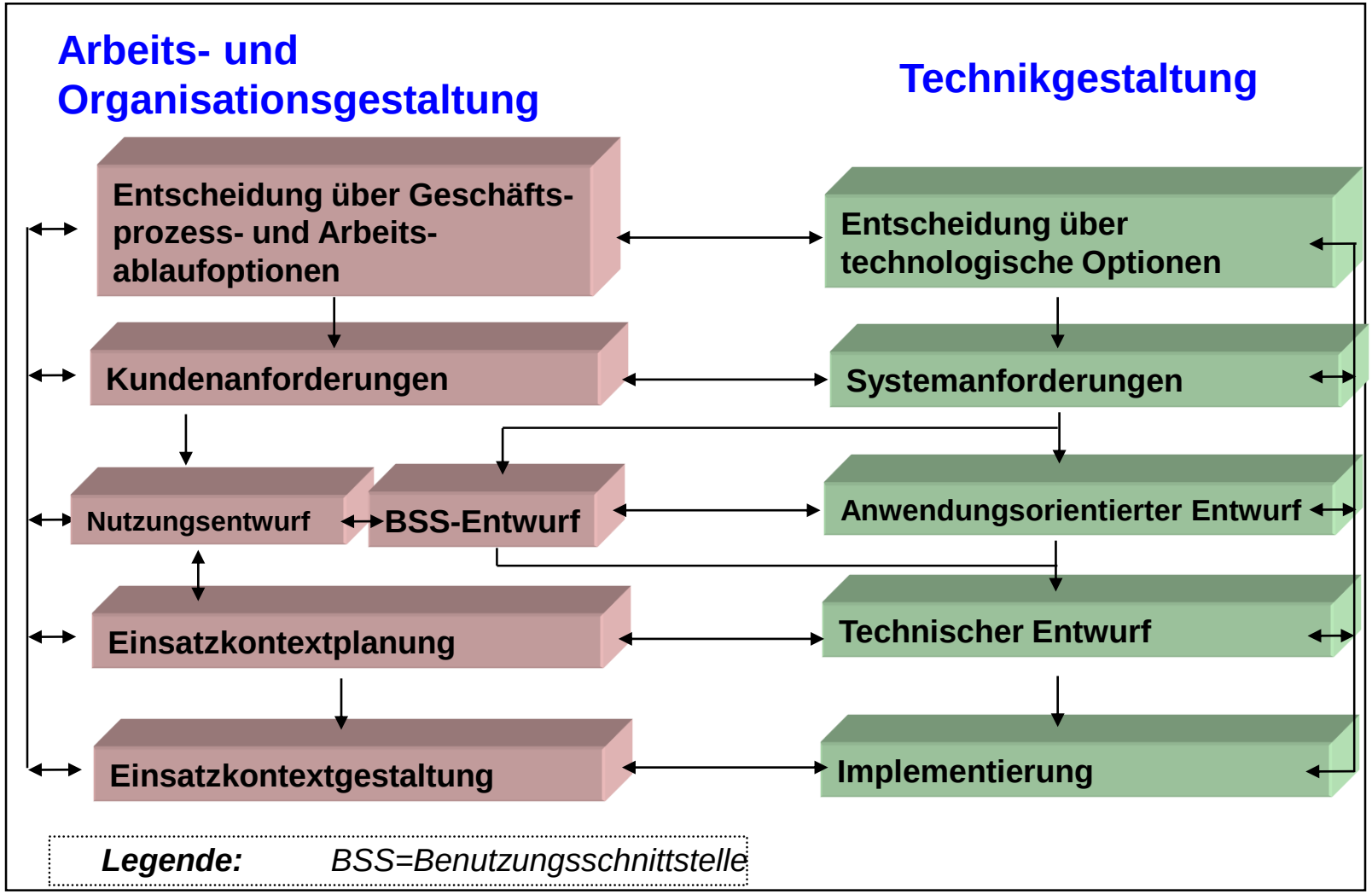
- um **Konsens** unter Beteiligten zu erzielen
- als **vertragliche Basis** zwischen KundInnen und EntwicklerInnen
- als Basis für **Meilensteine** im Projektmanagement
- als primäre **Vorgabe für Entwurf/Implementierung**
- als **Referenz für Verifikation und Validation**
- als **Grundlage für Change-/Releasemanagement und Handbucherstellung**

Lastenheft

Pflichtenheft

[Paech 2000]

SW-Entwicklung als Arbeits- und Technikgestaltung



Zusammenfassung

- Der Entwicklungsprozess und insbesondere die Anforderungsdokumente sind durch die **Interessen der Beteiligten** geprägt.

- Dies geht umso besser, je mehr sich die Beteiligten dieser Interessen **bewusst sind und diese Interessen explizit beschrieben sind** (z.B. durch Geschäftsprozessmodell, Geschäftszielvorgaben, Ergonomische Standards, Entwicklungs-Standards).

- Anforderungen müssen deshalb systematisch erhoben und spezifiziert werden. Systemanforderungen lassen sich systematisch aus Kundenanforderungen gewinnen.

Literatur

- K Pohl, R Rupp (2010) Basiswissen Requirements Engineering, dpunkt Verlag
- L Macaulay (1993) Requirements Engineering, Springer
- S. Lauesen (2002) Requirements Engineering, Addison-Wesley
- B. Paech, Aufgabenorientierte Softwareentwicklung, Integrierte Gestaltung von Unternehmen, Arbeit und Software, Springer Verlag 2000


Frohe Weihnachtszeit
und
Alles Gute für 2016