



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Agiles Requirements Engineering

Wiederholung / Zusammenfassung wichtiger Punkte

Master Technische Informatik – Embedded Systems –
Prof. Dr.-Ing. Hartmut Schirmacher

Inhalt

- Kontext, Stakeholder, Anforderungen und Ziele
- Kategorisierung und Dokumentation



Kontext, Stakeholder, Anforderungen und Ziele



Systemkontext

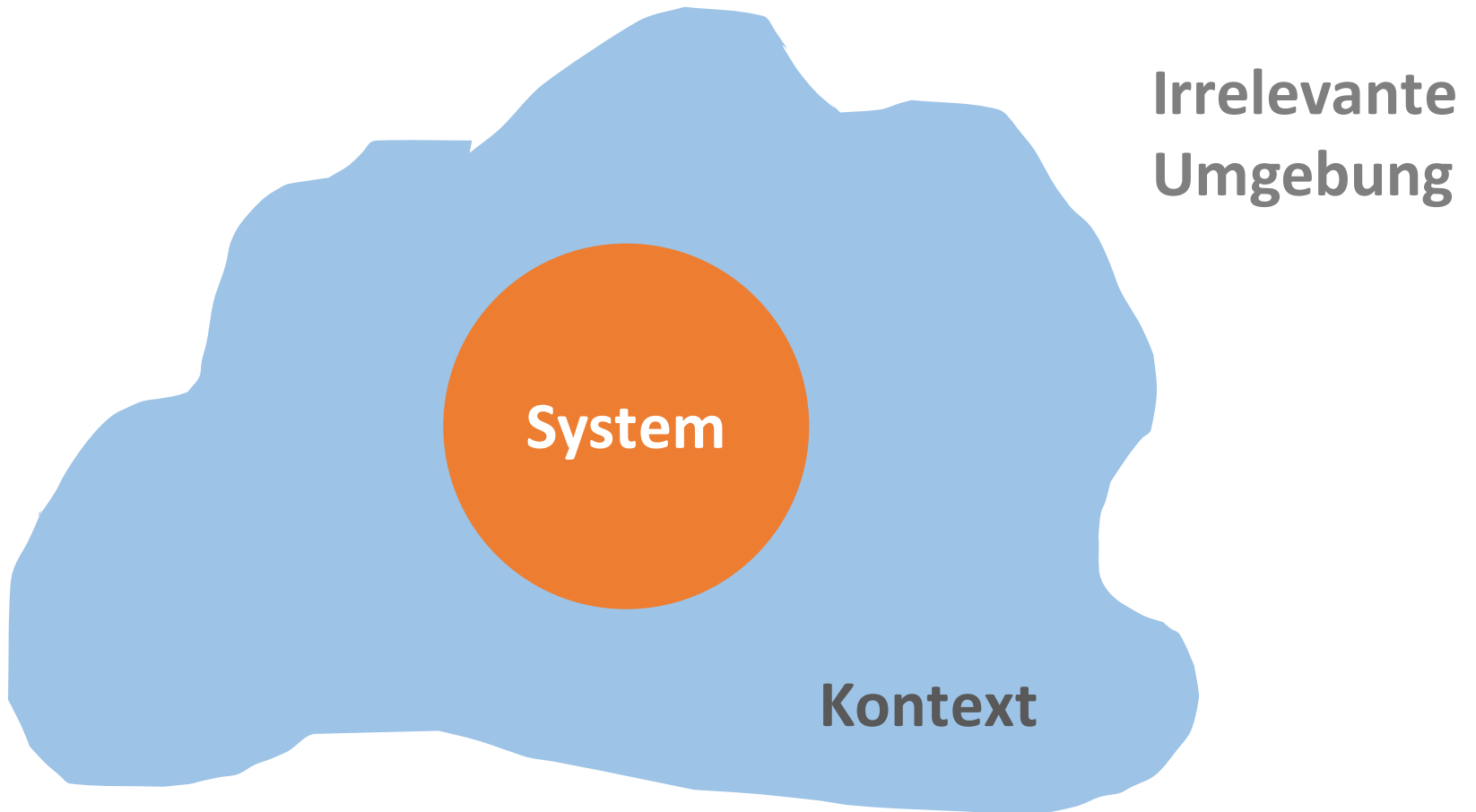
Definition *: Systemkontext

[...] der Teil der Umgebung eines Systems, der für die Definition und das Verständnis der Anforderungen des betrachteten Systems relevant ist.

- Wie integriert sich das System in die Realität?
- Betrachtetes System von Umgebung abgrenzen
- Beziehungen System – Umgebung identifizieren
- Materielles (Systeme, Dokumente) und immaterielles (z.B. Prozesse) beachten



System, Kontext, irrelevante Umgebung



Stakeholder

Definition *: Stakeholder

Ein Stakeholder eines Systems ist eine Person oder Organisation, die (direkt oder indirekt) **Einfluss auf die Anforderungen** des betrachteten Systems hat.

Zum Beispiel

- Angedachte Nutzer des Systems
- Kunden / Käufer, Betreiber
- Internes Management
- Dienstleister, Zulieferer
- Entwickler und Tester



Identifikation der Stakeholder

Stakeholder sind i.d.R. die wichtigste Quelle für Anforderungen

- Aufdecken von Widersprüchen (Zielkonflikte)
- Alle Stakeholder müssen in den Anforderungen adäquat vertreten sein (auch die Entwicklung!)

Wie finde ich sie?

- Fachexperten fragen, Stakeholder fragen
- Informationen zu vergleichbaren Produkte anschauen
- Sich den Einsatz des Produktes konkret vorstellen
- Sich das Entwickeln und Testen des Produkts konkret vorstellen



Anforderung nach Pohl/Rupp

Anforderung:

- (1) Eine **Bedingung oder Fähigkeit**, die **von einem Benutzer** (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels **benötigt** wird.
- (2) Eine Bedingung oder Fähigkeit, die ein System oder Teilsystem erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation, oder andere, **formell vorgegebene Dokumente** zu erfüllen.
- (3) Eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft gemäß (1) oder (2).

Diese Definition ist eine direkte Übersetzung aus IEEE 610.



Ziele

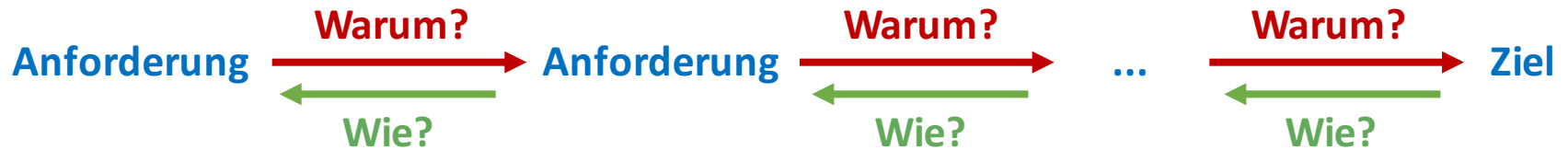
- Anforderungen beziehen sich (bereits) auf das Produkt / System
- Doch am Anfang: volle Konzentration auf den Stakeholder
- Was will Stakeholder X eigentlich (wirklich) erreichen?
(unabhängig davon, wie die Lösung aussieht)
- **Ziele = Schlüssel zu erwartungskonformer Lösung**

Aber Vorsicht, nicht alles ist ein brauchbares Ziel:

- *Ich möchte damit die Weltherrschaft erringen...*
- *Es muss einfach besser funktionieren als das jetzige System!*



Anforderung vs. Ziel



Ziel:

- In der **Problemdomäne des Stakeholders** formuliert
- **Unabhängig von der Realisierung des Produkts** formuliert
- Was will der **Stakeholder erreichen**?
- Nicht: was soll das Produkt genau tun?
(nach Meinung des Stakeholders)
- Anhand der Ziele kann man den Wert der Umsetzung späterer Anforderungen einschätzen
 - Bringt Feature X dem Stakeholder etwas in Bezug auf Ziel Y ?



Anforderung vs. Ziel: Beispiel

Beispiel: Produkt = Firmenwebseite

Firma soll von potentiellen Neukunden als „High Tech“ wahrgenommen werden.

Kein Produktmerkmal, sondern ein „abstrakteres“ Unternehmensziel

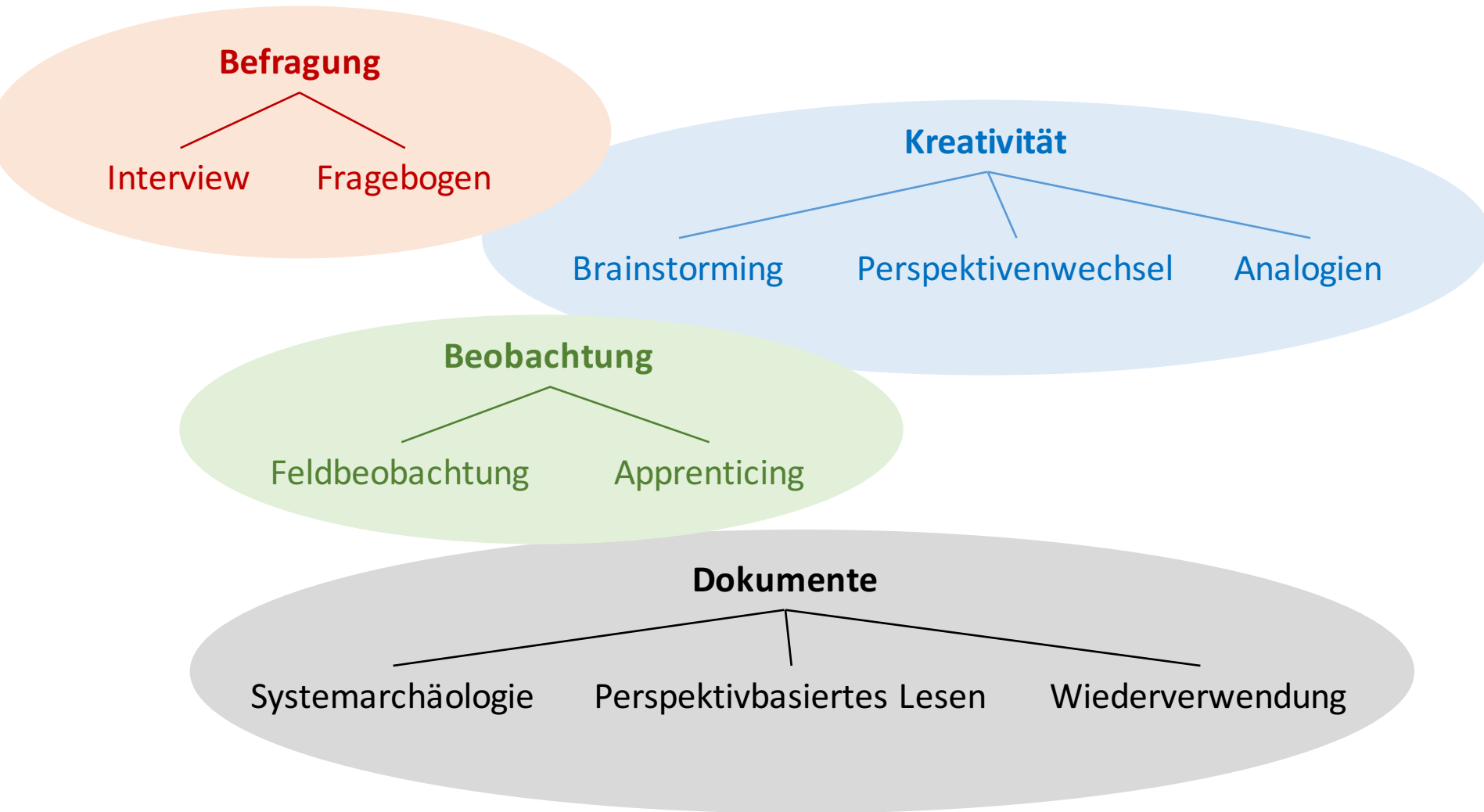
Startseite soll auf oberster Ebene (sofort sichtbar, nicht verborgen) Verweise auf Architektur und Technologie der angebotenen Lösungen anbieten.

Ein kleiner Personenkreis um den CTO des Unternehmens soll alle 3 Monate automatisch zu einem elektronischen Review der Technologie– und Architekturinhalte eingeladen werden.

Produktanforderungen vor dem Hintergrund des Unternehmensziels



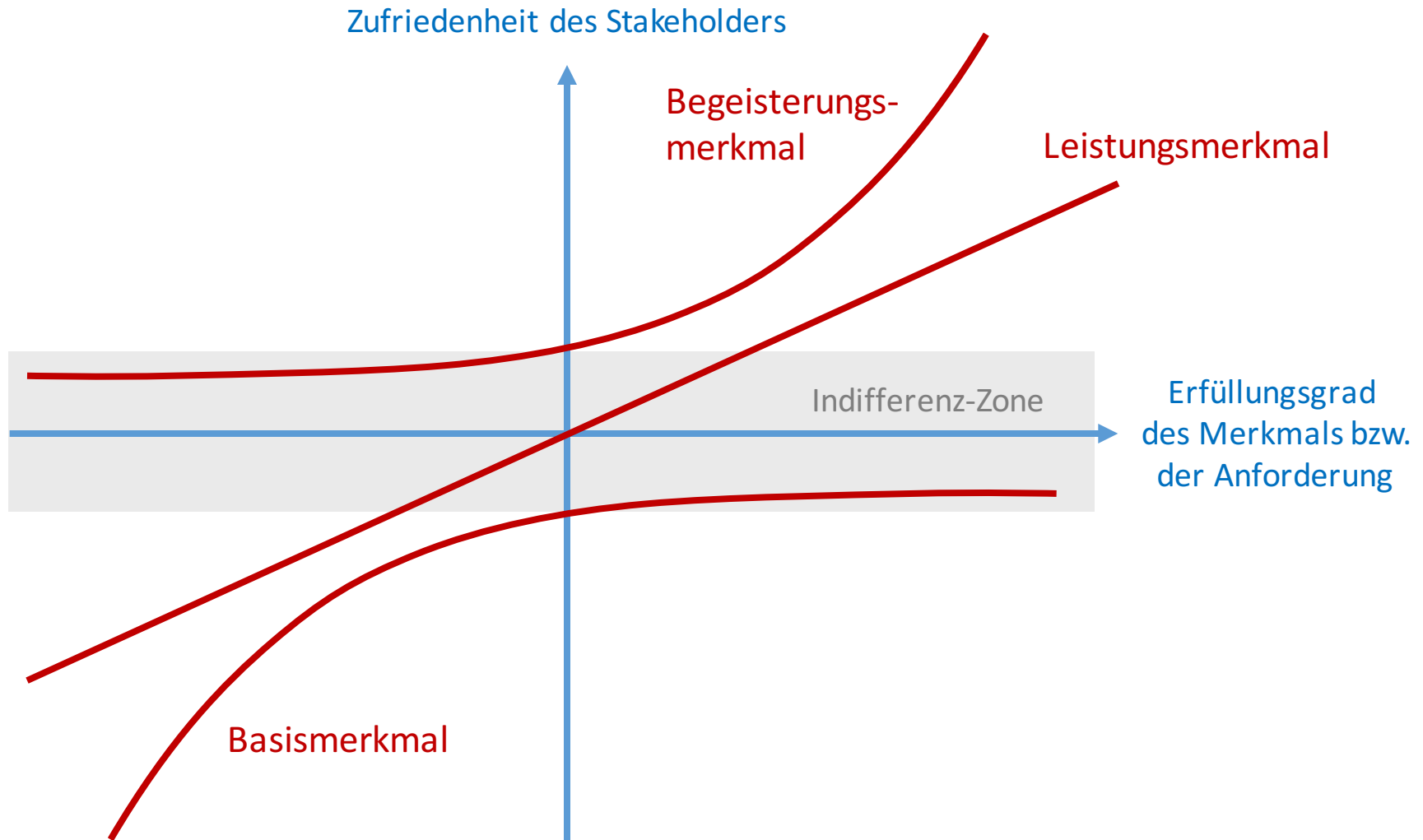
Ermittlungstechniken



Anforderungen
kategorisieren und
dokumentieren



Kano-Modell



Anforderungsarten: Funktionale Anforderung

Definition * : Funktionale Anforderung

[...] eine Anforderung bezüglich des **Ergebnisses eines Verhaltens**, das von einer Funktion des Systems bereitgestellt werden soll.

- Funktionalität, die das System zu Verfügung stellen soll
- Auch Funktionen der Schnittstellen, die bedient werden müssen



Anforderungsarten: Qualitätsanforderung

Definition * : Qualitätsanforderung

[...] eine Anforderung, die sich **auf ein Qualitätsmerkmal bezieht**, das **nicht durch funktionale Anforderungen abgedeckt** wird.

- Gewünschte Qualitäten festlegen
- Häufig großer Einfluss auf Systemarchitektur

Auch bekannt unter
„Nichtfunktionale Anforderungen“



Qualitätsanforderungen nach ISO/IEC 25010:2011 *

- **Performanz**
 - Antwortzeitverhalten, Ressourcenverbrauch
- **Sicherheit**
 - Nachweisbarkeit, Authentizität, Vertraulichkeit, Integrität
- **Zuverlässigkeit**
 - Verfügbarkeit, Fehlertoleranz, Wiederherstellbarkeit
- **Benutzbarkeit / Ergonomie**
 - Barrierefreiheit, Erlernbarkeit, Bedienbarkeit, ...
- **Änderbarkeit / Wartbarkeit**
 - Wiederverwendbarkeit, Analysierbarkeit, Modifizierbarkeit, Prüfbarkeit
- **Übertragbarkeit**
 - Anpassbarkeit, Installierbarkeit, Austauschbarkeit



Funktionale vs. Qualitätsanforderungen

→ Funktionale Anforderungen beantworten die Frage
WAS soll das System TUN / BEWIRKEN

→ Qualitätsanforderungen beschreiben, welche
Nebenbedingungen das Produkt dabei *erfüllen*
muss.

Vorgehen:

- *Erst prüfen, ob eine Funktion / ein Verhalten beschrieben wird*
→ Funktionale Anforderung
- *Enthält die funktionale Anforderung auch Bedingungen bzgl. Performanz, Stabilität, Ergonomie, ... etc?*
→ Trennen von funktionalen und Qualitätsanforderungen



Anforderungsarten: Randbedingung

Definition * : Randbedingung (Constraint)

[...] eine Anforderung, die den Lösungsraum jenseits dessen einschränkt, was notwendig ist, um die funktionalen Anforderungen und die Qualitätsanforderungen zu erfüllen.

- Projektbeteiligte haben keinen Einfluss
- Vorgabe, oftmals in Vertrag festgelegt
- Führt zu keiner neuen Funktionalität, sondern schränkt ein
- für zu entwickelndes Produkt, oder für Entwicklungsprozess (z.B. „Verfügbarkeit spätestens Q2/2020“)



Funktionale Anforderung vs. Constraint

Ein Constraint sieht oftmals einer funktionalen Anforderung ähnlich:

Das System soll unter Microsoft Windows 10 ausführbar sein.

Steht die obige Anforderung alleine, oder gibt es mehrere Anforderungen bzgl. der unterstützen Plattformen? Z.B.

- *Das System soll Desktop-Betriebssysteme mit Maus-, Trackpad- und Toucheingabe unterstützen.*
- *Mobile Endgeräte sollen nicht unterstützt werden. Die Verwendung findet in einer Büroumgebung statt, mit Bildschirmgrößen ab 20 Zoll.*
- *Das System soll initial lediglich für Windows 10 verfügbar sein.*

Constraint drückt u.a. aus, dass dieser Anforderung eine explizite einschränkende Entscheidung zugrunde liegt.



Was tun mit all diesen Kategorien?

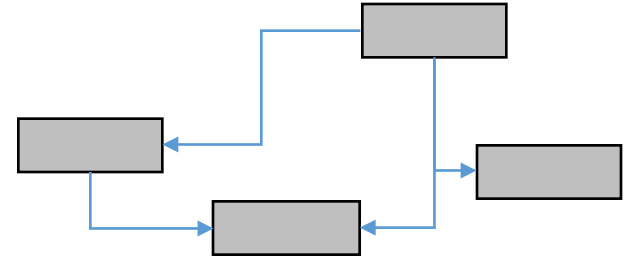
- Releaseplanung
 - Welche Features müssen mindestens in welches Release?
 - Hier benötigt man eine Liste aller ermittelten Anforderungen
 - Bei der Priorisierung kann Kano helfen.
- Architekturentwurf
 - Welche Qualitätsanforderungen müssen erfüllt werden?
 - Welche Gruppen von Funktionen gibt es (→ Komponenten)
 - Welche Plattformen und externen Systeme müssen unterstützt werden?
- Schnittstellenentwurf (UI + technische)
 - Welche Funktionen hat das System nach außen?
 - Interaktionszenarien



Drei Perspektiven

(Statische) Strukturperspektive

- Datenmodelle
- Nutzungs- und Abhängigkeitsbeziehungen von Komponenten



Funktionsperspektive

- Ein- und zugehörige Ausgaben, aus dem Systemkontext betrachtet



Verhaltensperspektive

- Ereignis- / zustandsorientierte Betrachtung

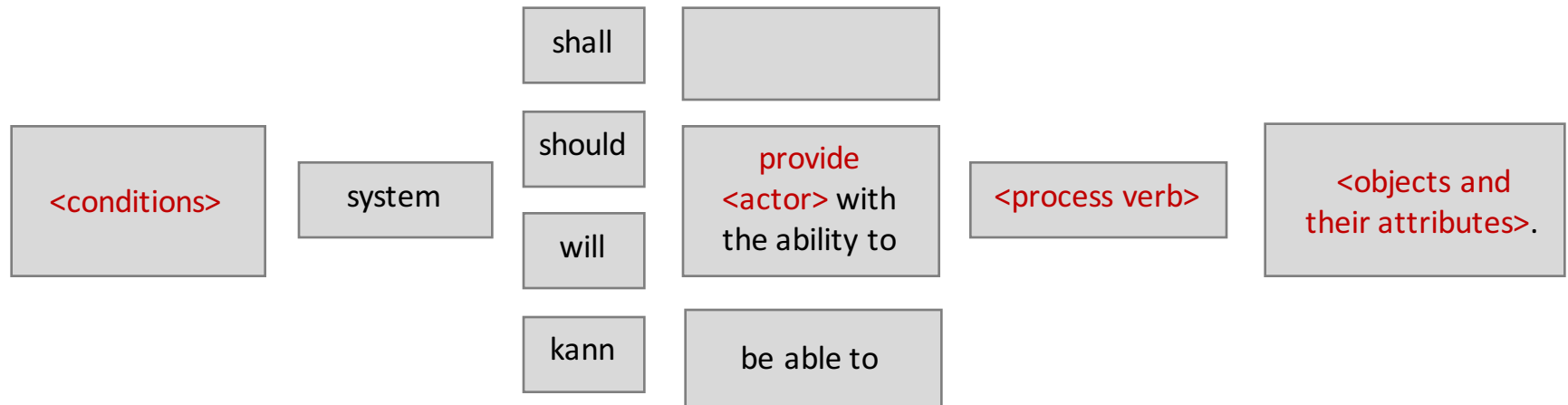


Dokumentation in natürlicher Sprache

- Vorsicht vor Verzerrungen durch Sprachtransformationen (Auslassungen, Verallgemeinerungen, etc.)
- Abhilfe: Sprachschablonen, Regeln zur konsistenten Dokumentation



Schablone für funktionale Anforderungen, *englisch*



Szenarien, Use Cases, Stories

- Interaktionen stellen i.d.R. großen Teil der Funktionalität dar
- Szenarien sind eine natürliche Art, diese Interaktionen zu dokumentieren

Use Cases / Szenarien =

(eine Art der)

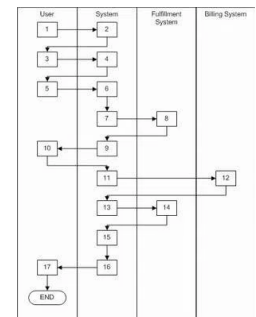
Dokumentation funktionaler Anforderungen



Szenario

Szenario:

- Eine geordnete Menge von Interaktionen zwischen Partnern, in der Regel zwischen einem System und einer Menge systemexterner Akteure.
- Kann eine konkrete Interaktionsfolge (**Beispielszenario**) oder eine Menge möglicher Interaktionen (**abstraktes Szenario** oder **Typszenario**) sein.
- Bekannteste Form: Jacobson/UML:
Use Case Scenario = Teil einer Use Case Beschreibung
- Einzelnes Szenario kann alternativ zu natürlicher Sprache auch mittels Sequenz- oder Aktivitätsdiagramm modelliert werden.



Use Case

- Sammlung von Szenarien mit demselben Benutzerziel
- Aufteilung in Hauptszenario (Main Success Scenario) und Erweiterungen / Alternativen
 - Liste von Erweiterungen im Use Case
 - Erweiterungen können durch eigene Use Cases modelliert werden
- Include-Beziehung
 - Verweise auf Unter-Use Cases
 - Abläufe, die Teil mehrerer größerer Abläufe sein



User Story

- Mentaler Anker, um Anforderungen zu dokumentieren
 - Fokus auf Kommunikation mit Stakeholdern
- Card + Conversation + Confirmation
- Karte hat Vorder- und Rückseite
 - *Als <Rolle> möchte ich <Ziel/Aktion> [, um <Geschäftsziel/Nutzen>] .*
 - Aufwand
 - Priorität
 - Akzeptanzkriterien
 - Referenzen / weiterführende Informationen



Unterscheidung User Story vs. Use Case?

Zu dieser Frage gibt es sehr viele Widersprüchliche Informationen in Literatur und Web. Aus meiner Erfahrung ist die nützlichste und eindeutigste Unterscheidung die folgende von Alistair Cockburn:

„A **user story** is the *title* of *one* scenario, whereas
a **use case** is the *contents* of *multiple* scenarios“



User Story: Vorsicht Größe + Kategorie

- Die agilen Vorreiter der User Stories unterscheiden nicht explizit zwischen funktionalen Anforderungen und Qualitätsanforderungen
- Manche Autoren verwenden Stories „für alles“, andere wiederum im Sinne der Umsetzbarkeit innerhalb eines Sprints
- Beispiel aus einer Präsentation zu „Offline First“:

USER STORY

As a ~~m~~obile app user,
I want to be sure
that my data is actually
available when I need it

USER STORY

As a ~~m~~obile app user, when I'm
offline, I want to be able to use
features that don't require a
connection



Detailstufe und Verfeinerung

Ultimatives Ziel aus Entwicklungssicht ist das Verfeinern funktionaler Anforderungen in Slices (a.k.a. Stories), welche innerhalb eines Sprints realisiert werden können.

- Einzelne Szenarien, bei denen ein Benutzer in einem bestimmten Kontext ein bestimmtes Ziel verfolgt
- UI ableitbar
- Akzeptanzkriterien ableitbar

... hierzu mehr im nächsten SU und der nächsten Aufgabe



Beispiele



Benutzer wünscht möglichst intuitive Bedienbarkeit.



Der Fahrgast soll in der Kabine jederzeit einen sofortigen Stopp der Kabine bewirken können.



Benutzer soll monopolaren Survey mit dem IPG des Patienten durchführen können.



Benutzer soll einen neuen Patienten einrichten können.



...



Fragen?

