

Grundlagen des Requirements Engineering

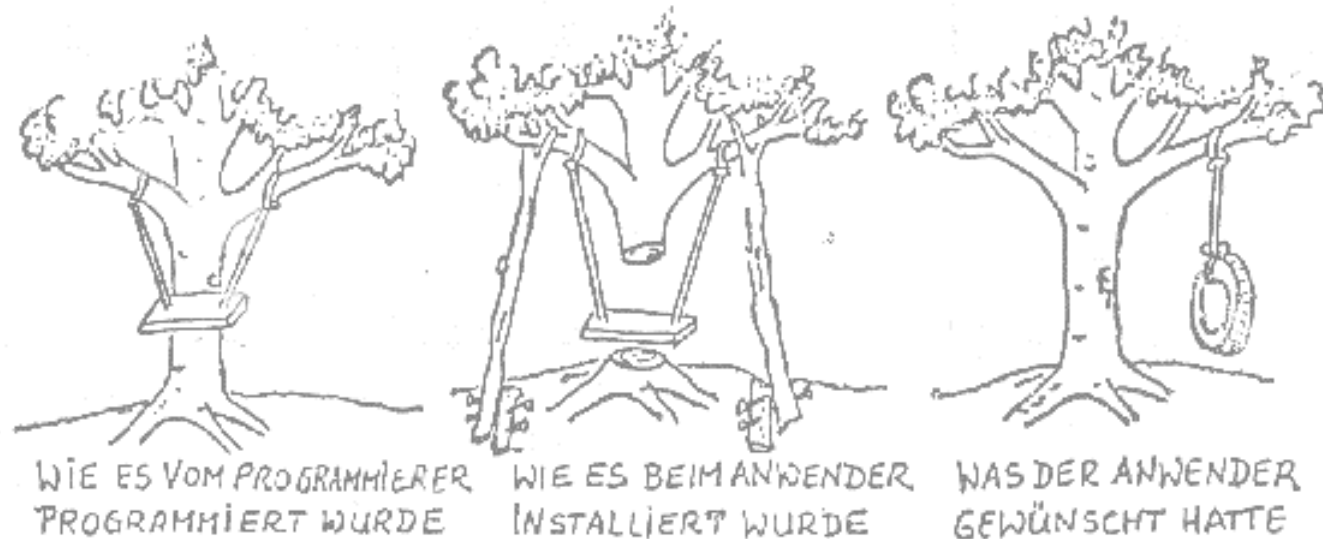
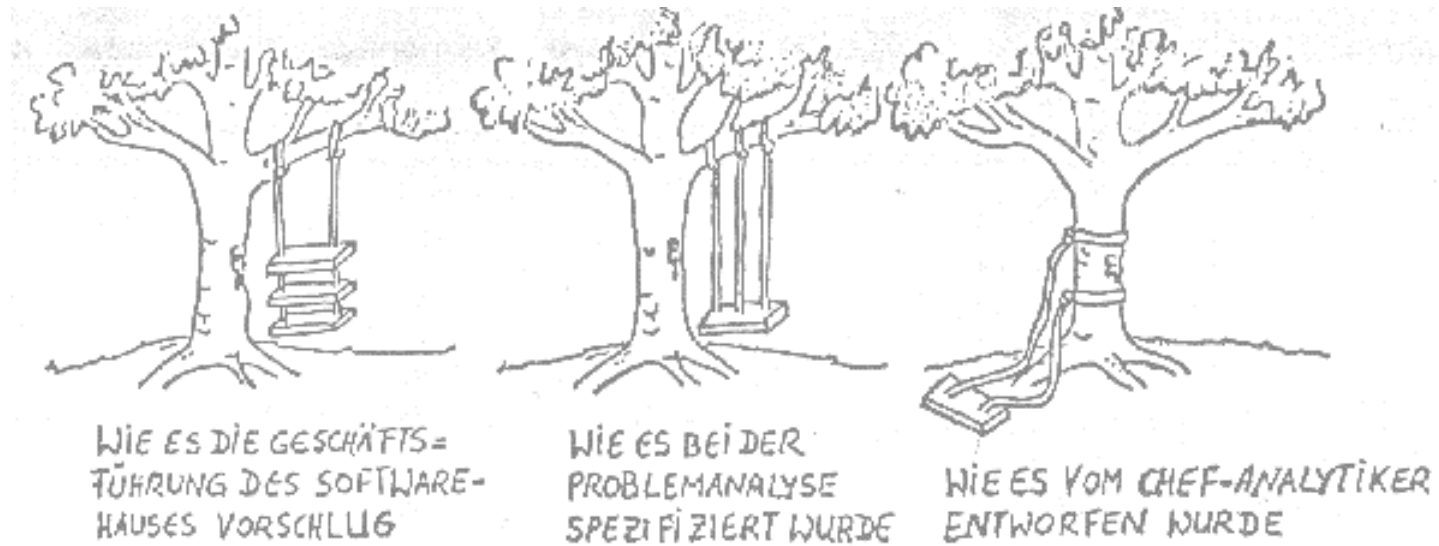
Vorlesungsreihe zum Thema Software Engineering von Software im Automobil

24.05.2004

BMW Group



Motivation



Kriterien erfolgreicher Systementwicklung

- **Das Produkt wird termingerecht ausgeliefert**
- **Das Produkt wird in den Grenzen der vorausgeschätzten Kosten fertiggestellt**
- **Der Kunde bekommt, was er eigentlich wollte**
- **Die Anwender erhalten, was sie erwartet hatten**

Wirklichkeit der Produktentwicklung

- **Geringe Erfolgsrate**
 - 53 % gehen schief,
 - 31 % teilweise erfolgreich
 - nur 16 % erfolgreich
- **Planungen sind völlig unrealistisch**
- **45 % der Funktionen werden nie genutzt**
- **Hohe Geldverluste**
 - durch erfolglose Projekte
 - durch Überentwicklung, die aber dennoch Wartung erfordert

„Small is beautiful“ - Je größer das Projekt, um so größer wird die Wahrscheinlichkeit eines Misserfolgs

Warum Projekte schiefgehen

1	Unvollständige Requirements	13.1%
2	Benutzer nicht involviert	12.4%
3	Zuwenig Ressourcen	10.6%
4	Übermäßige Erwartungen	9.9%
5	Keine Management Unterstützung	9.3%
6	Spezifikationsänderungen	8.7%
7	Fehlende Planung	8.1%
8	Nicht mehr benötigt	7.5%

Quelle: Standish Group 1995
& Scientific American, Sept 1994

Keines scheiterte aus technischen Gründen!

Schlüssel: Wissen wohin man will



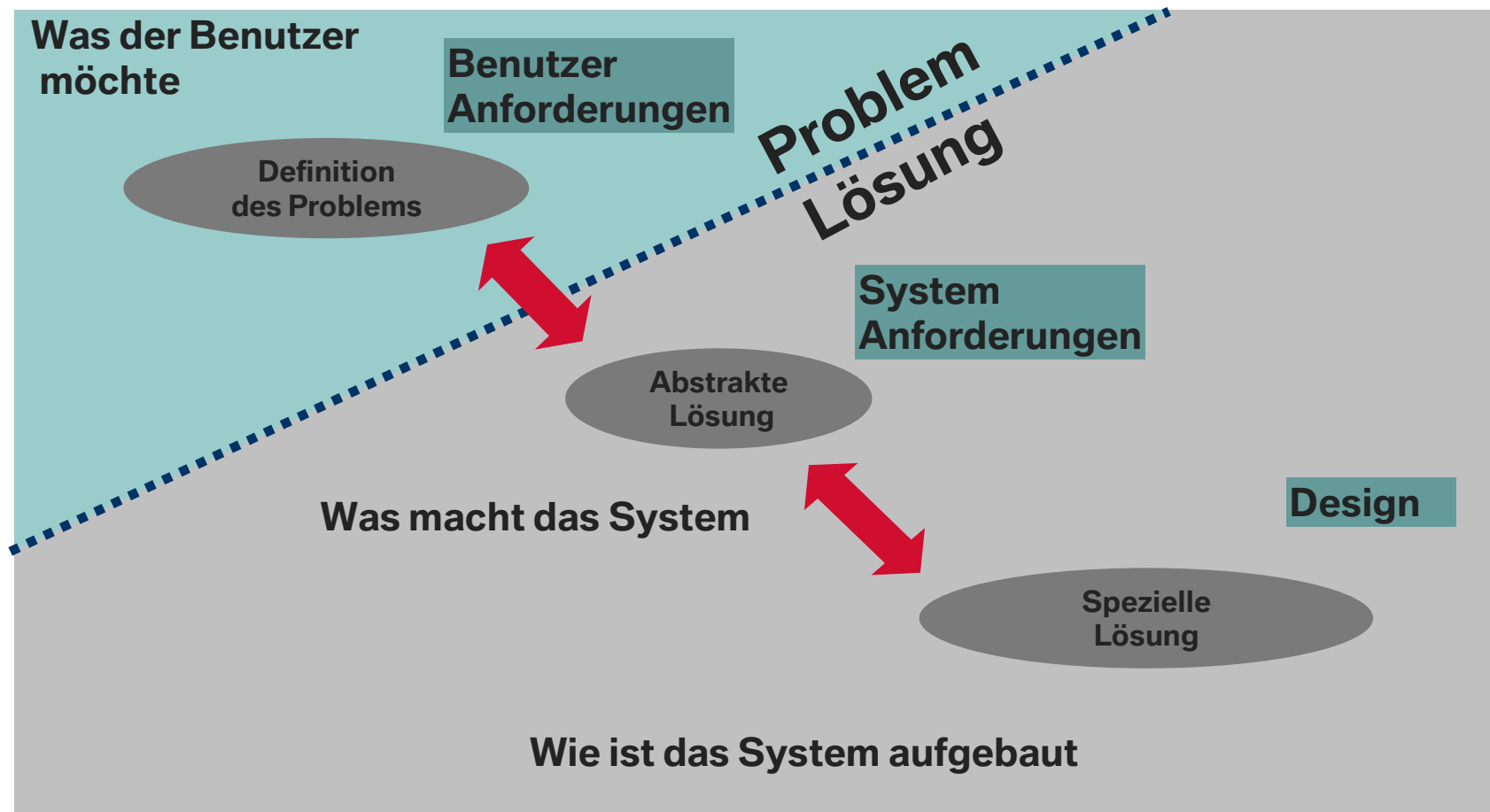
Anforderungen erfassen um zu verstehen, welche Anwenderprobleme das Produkt lösen soll

Ohne gute und gut strukturierte Anforderungen wird kein klarer Weg zum Produkt erkennbar

Mitglieder im Team gehen in verschiedene Richtungen

Problem- und Lösungsraum

Problem und Lösung



Was sind Anforderungen

Abgrenzung Anforderung

- **Bedürfnis**

Ein Problem oder Wunsch, den ein Benutzer oder Kunde formuliert

- **Anforderung**

Ein für das System vorgeschlagener Weg um das Bedürfnis zu befriedigen

- **Feature**

Eine konkrete Systemeigenschaft

Definition

Anforderungen im RE/RM:

„Sammlung von Aussagen, die in einer klaren, konsistenten und eindeutigen Art alle signifikanten Aspekte des zu entwickelnden Systems beschreiben“

Herausforderungen

- Anforderungen sind schwer zu sichtbar zu machen
- Anforderungen ändern sich
- Kommunikation zwischen allen Beteiligten
- Anforderungsmanagement ist selten als Teil des ingenieurmäßigen Vorgehens anerkannt
- Abbildung der Prozesse, Methoden und Werkzeuge in das reale Unternehmensumfeld

Beispiele für Anforderungen

Cabrio Verdeck

ID	Requirements for convertible top
1	1 Benutzeranforderungen
3	1.1 Das Verdeckschließen bzw. -öffnen muss durch die Betätigung eines Bedienelementes im Fahrzeug ausgelöst werden können ▶
5	1.2 Das Verdeckschließen bzw. -öffnen muss durch die Betätigung eines Bedienelementes der Fernsteuerung ausgelöst werden können ▶
6	1.3 Das Verdeck soll bei Regen im Stand automatisch schließen ▶
7	1.4 Das Verdeck darf bei Geschwindigkeiten > 30 km/h weder Öffnen noch Schließen beginnen
8	1.5 Wird beim Öffnen oder Schließen ein Einklemmen erkannt, soll die Operation abgebrochen werden
9	1.6 Das Bedienelement im Fahrzeug muss Vorrang vor dem Fernsteuerung haben
10	1.7 Nach einem Abbruch soll das Verdeck je nach Bedienung mit Öffnen oder Schließen fortfahren können
11	1.8 Befindet sich eine Person im Fahrzeug, muss das automatische Schließen bei Regen deaktiviert werden
12	1.9 Der Bewegungsvorgang des Verdecks kann per Fernsteuerung oder Bedienelement jederzeit abgebrochen werden
2	2 Systemanforderungen
13	2.1 Das Verdeck muss erkennen können, ob es offen, oder geschlossen ist ◀

Anforderungsklassen

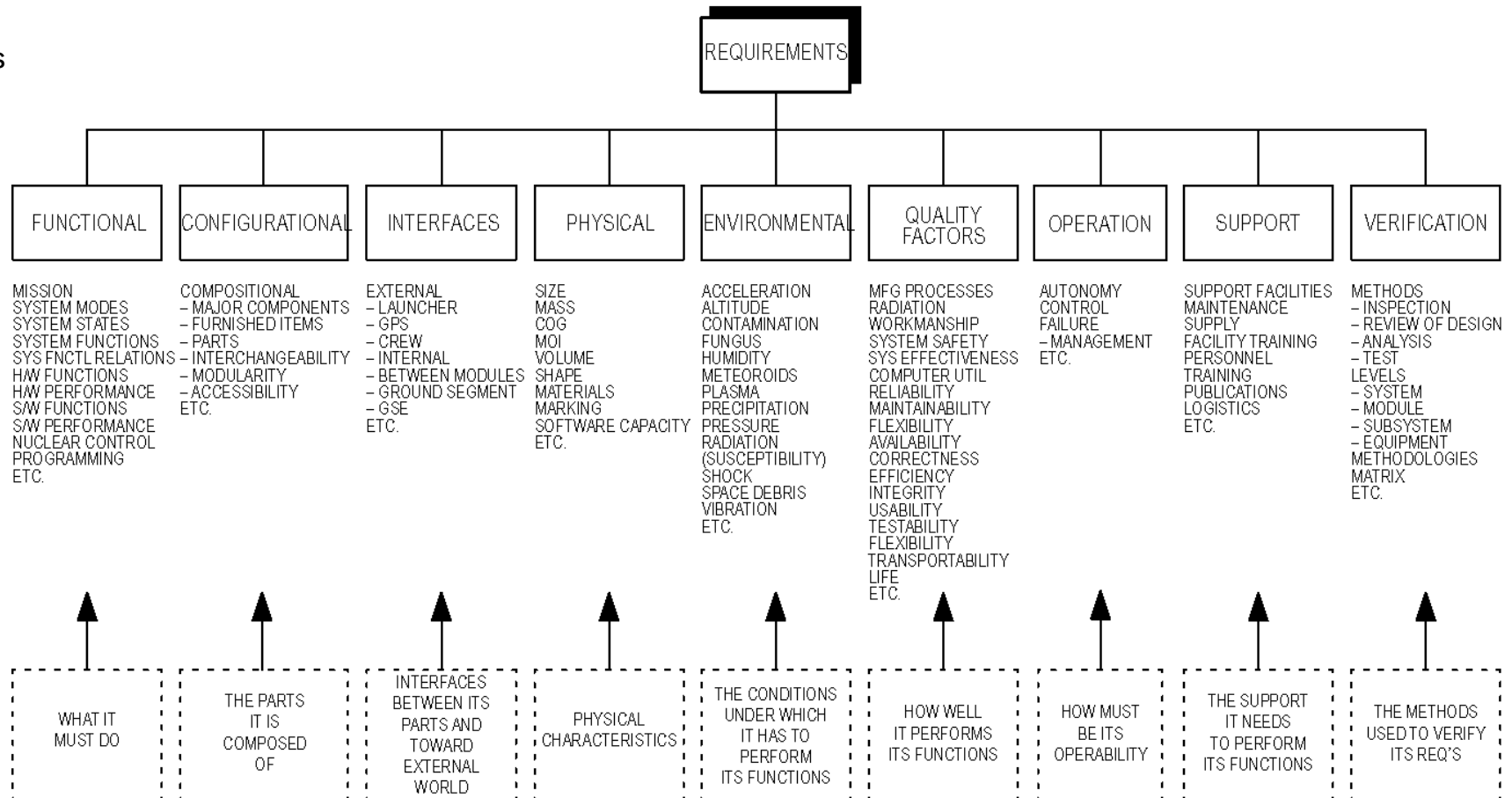
Anforderungsklassen

- **Funktionale Anforderungen:**
 - legt die Funktionalität des Systems fest
 - definiert, wie das System sich dem Nutzer darstellt und was es für des Nutzer tut

- **Nicht-funktionale Anforderungen:**
 - Standards, Gesetze, Regeln, Normen und Verträge die einzuhalten sind
 - Performance, Wiederverwendbarkeit, externe Schnittstellen
 - Design und Implementierungsrandbedingungen
 - Qualitätsattribute
 - **Sicherheit**

Beispiele für Anforderungsklassen

Beispiel ESA
 ECSS Standards

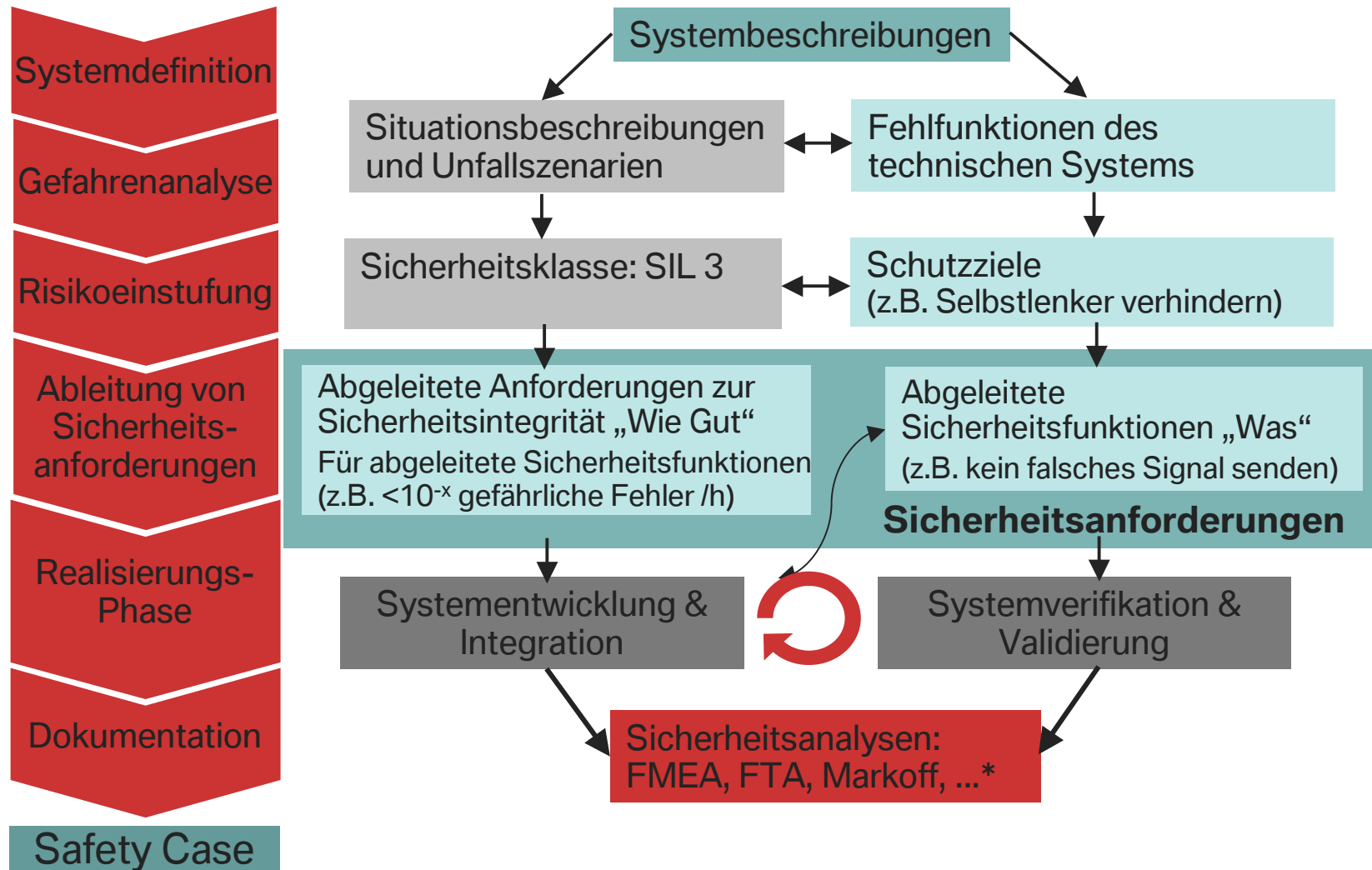


Quelle: ECSS-E-10a

Beispiel für nichtfunktionale Anforderungen

Sicherheitsanforderungen

Sicherheitsanforderungen



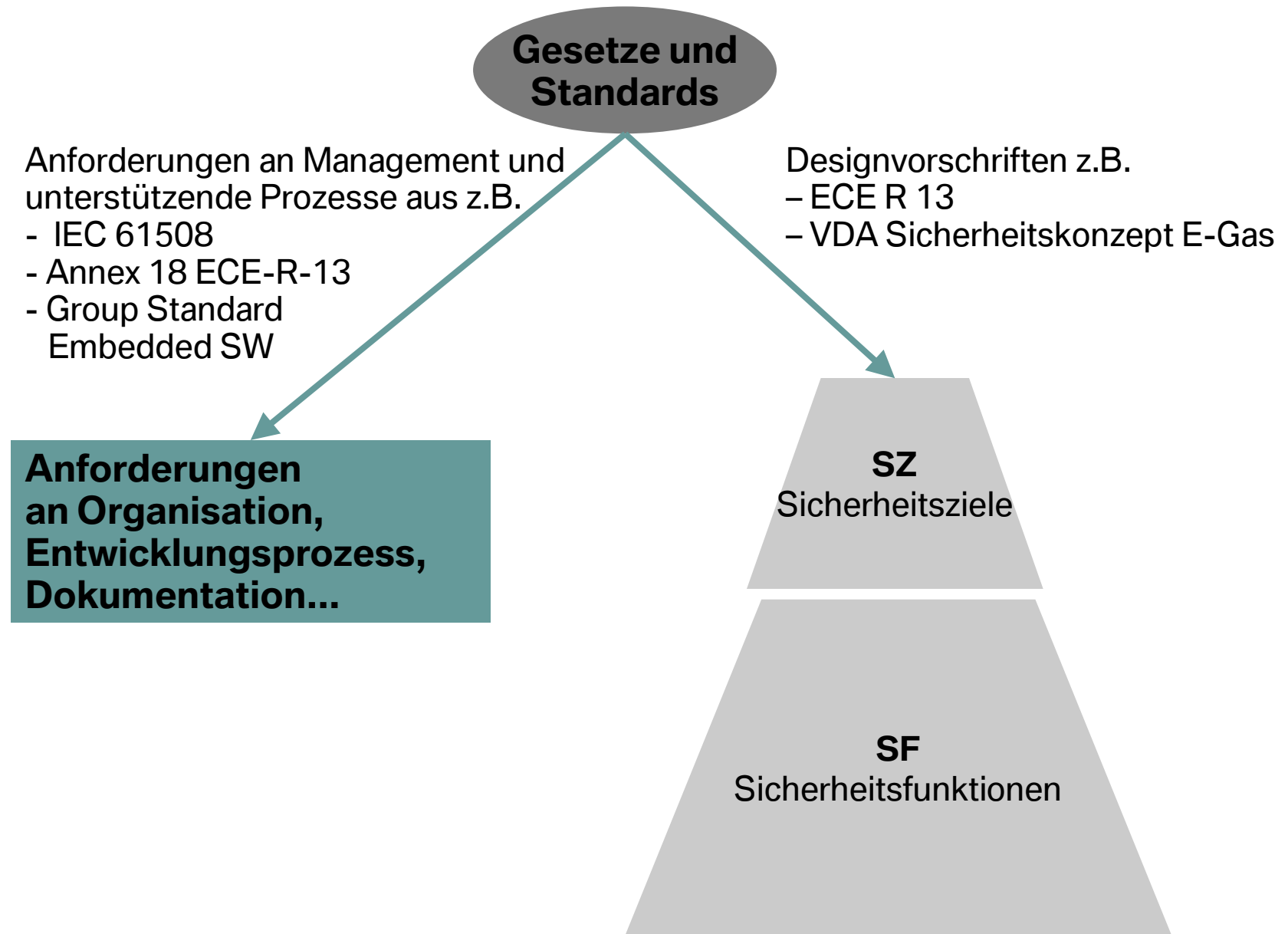
* FMEA = Failure Mode and Effects Analysis

FTA = Fault Tree Analysis

Quellen für Sicherheitsanforderungen



Quellen für Sicherheitsanforderungen



Anforderungen an Sicherheitsanforderungen

Sicherheitsanforderungen müssen folgende Kriterien erfüllen:

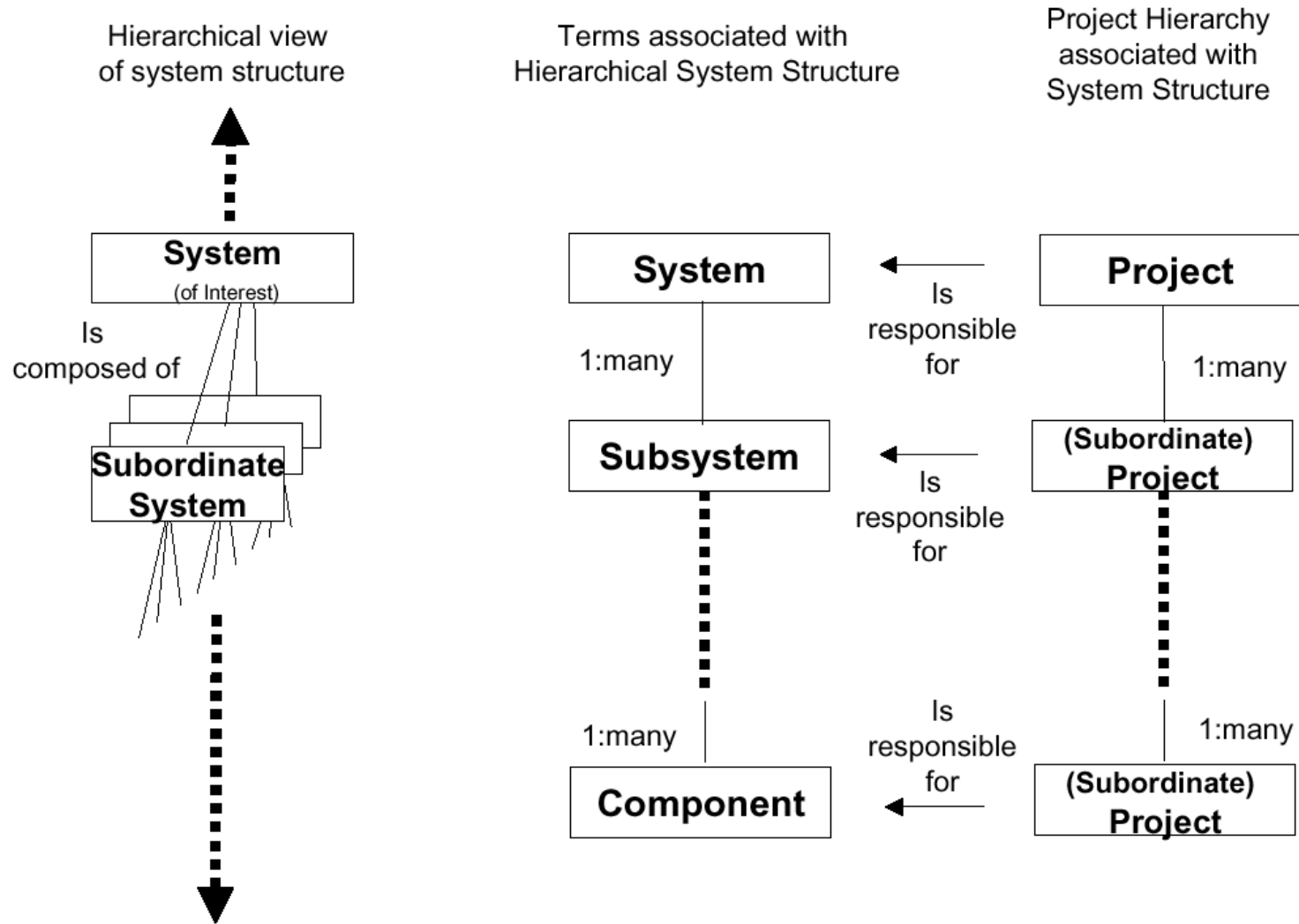
- Eindeutig und widerspruchsfrei formuliert
 - Spezifikation der Zeitanforderung (innerhalb 100ms, für mindestens 10 min..)
 - Spezifikation des Systemzustandes (beim Start, bei einem beliebigen Ausfall, ...)
 - Spezifikation der Sicherheitsintegrität: SIL, PDF, HFT

- Verfolgbar
 - Herkunft nachvollziehbar (ableitbar)
 - Erfüllung nachweisbar (Tests, Analysen)

- Anforderung braucht Adressat (wer setzt sie um, wer testet, ...)

Anforderungsdekomposition

Systemstrukturierung



Quelle: ISO 15288

Requirements Dekomposition

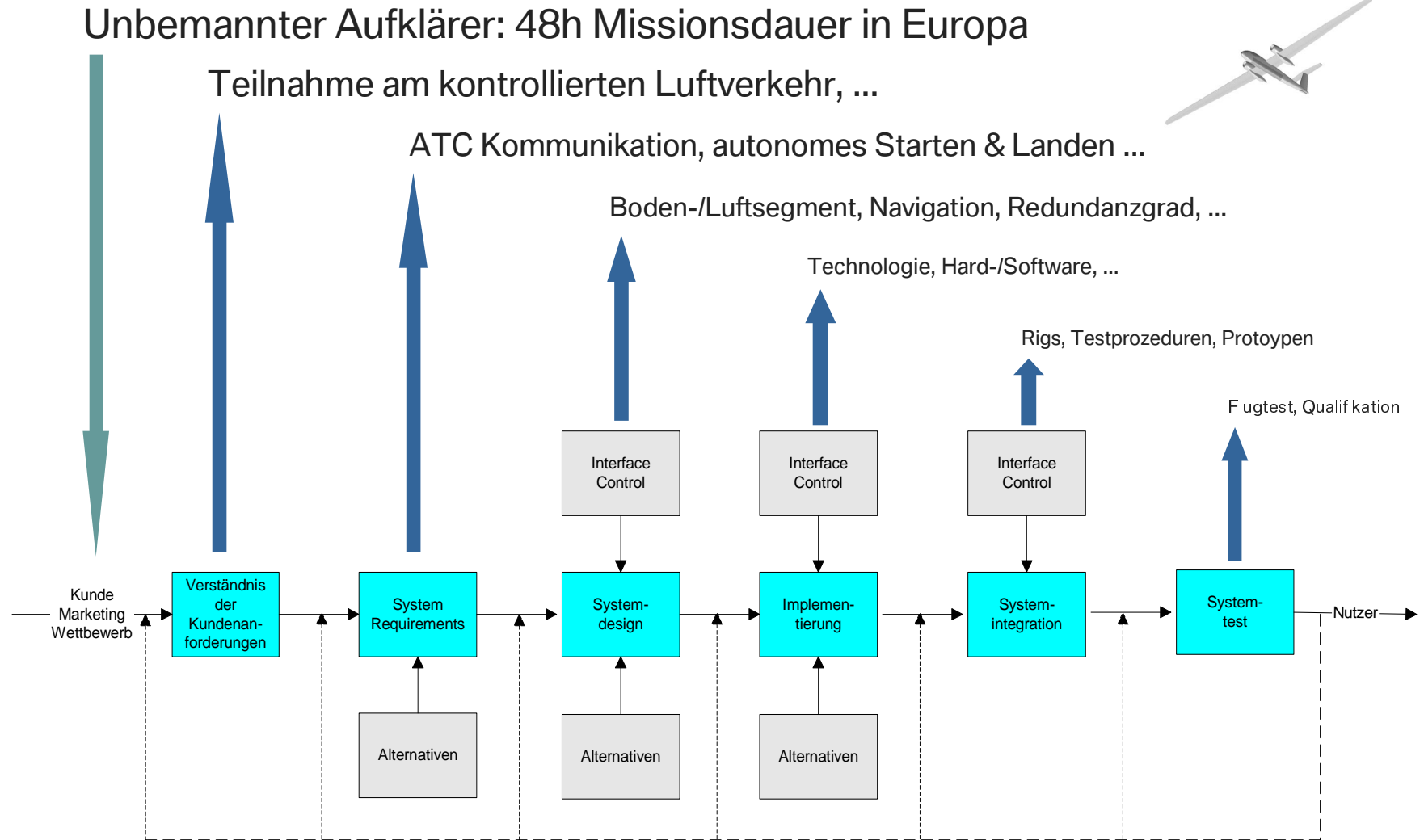
- **Operationelle** Requirements legen die Anforderungen an das System aus Benutzersicht fest
- **Funktionale** Requirements werden aus operationellen Requirements abgeleitet
- **Subsystem** Requirements werden aus Funktionalen Requirements abgeleitet

- Subsystem Requirements werden auf Systemkomponenten abgebildet
- Systemkomponenten bestehen aus Modulen

- **Operationelle** Requirements werden durch Tests validiert
- **Subsystem** Requirements werden durch Tests verifiziert

Beispiel aus der Luft- und Raumfahrtindustrie

Beispiel URAV



Lasten- und Pflichtenheft

Lasten- versus Pflichtenheft

- **Lastenheft - Anforderungsspezifikation**
 - Aufgabenbeschreibung
 - Definition der Sollanforderungen.
 - Verantwortung Auftraggeber
- **Pflichtenheft - Umsetzungsspezifikation**
 - Lösungsansatz
 - Zusammenfassung der Lösung in einer Leistungsbeschreibung
 - Verantwortung Auftragnehmer

Lastenheft – Inhalt

- Funktionalität
- Funktionale Randbedingungen
- Design Randbedingungen
- Datenformate und Kommunikationsprotokolle
- Projekt Management
- Umweltaanforderungen

Lastenheft – Kriterien

- **Vollständigkeit (Gültigkeit, Genauigkeit)**
 - „wird das wirklich verlangt“
- **Vollständigkeit**
 - „nichts vergessen“
- **Konsistenz**
 - „passt alles zusammen“
- **Verifizierbar**
 - „ist die Umsetzung prüfbar“
- **Verständlichkeit und Eindeutigkeit**
 - „verstehen alle das gleiche“
- **Verfolgbarkeit**
 - „durch alle Designphasen sichtbar“

Verifikation und Validierung

Nachweis

- **Anforderungen müssen auf jeder Spezifikations- und Implementierungsebene nachweisbar sein**
- **Anforderungen müssen prüfbar formuliert sein**
- **Die Prüfart sollte zu jeder Anforderung mit spezifiziert werden**

Definition

- **Validierung – “Bauen wir das richtige System?”**
Nachweis der vom Kunden formulierten gewünschten Systemeigenschaften (operationelle Requirements)
- **Verifikation – “Bauen wir das System richtig?”**
Nachweis der korrekten (technischen) Umsetzung der System Requirements (funktionale Requirements)
- **Nachweisverfahren**
Test (T)
Demonstration (D)
Analyse (A)
Inspektion (I)

Anforderungsspezifikation / Verifikationsmethoden

- **Rapid Prototyping**
- **Animation**
- **Reviews**
- **Wissensbasierte Methoden**
- **Separation of concerns (Aufteilung von Modellen nach Anforderungen)**

Requirements Engineering

Definition

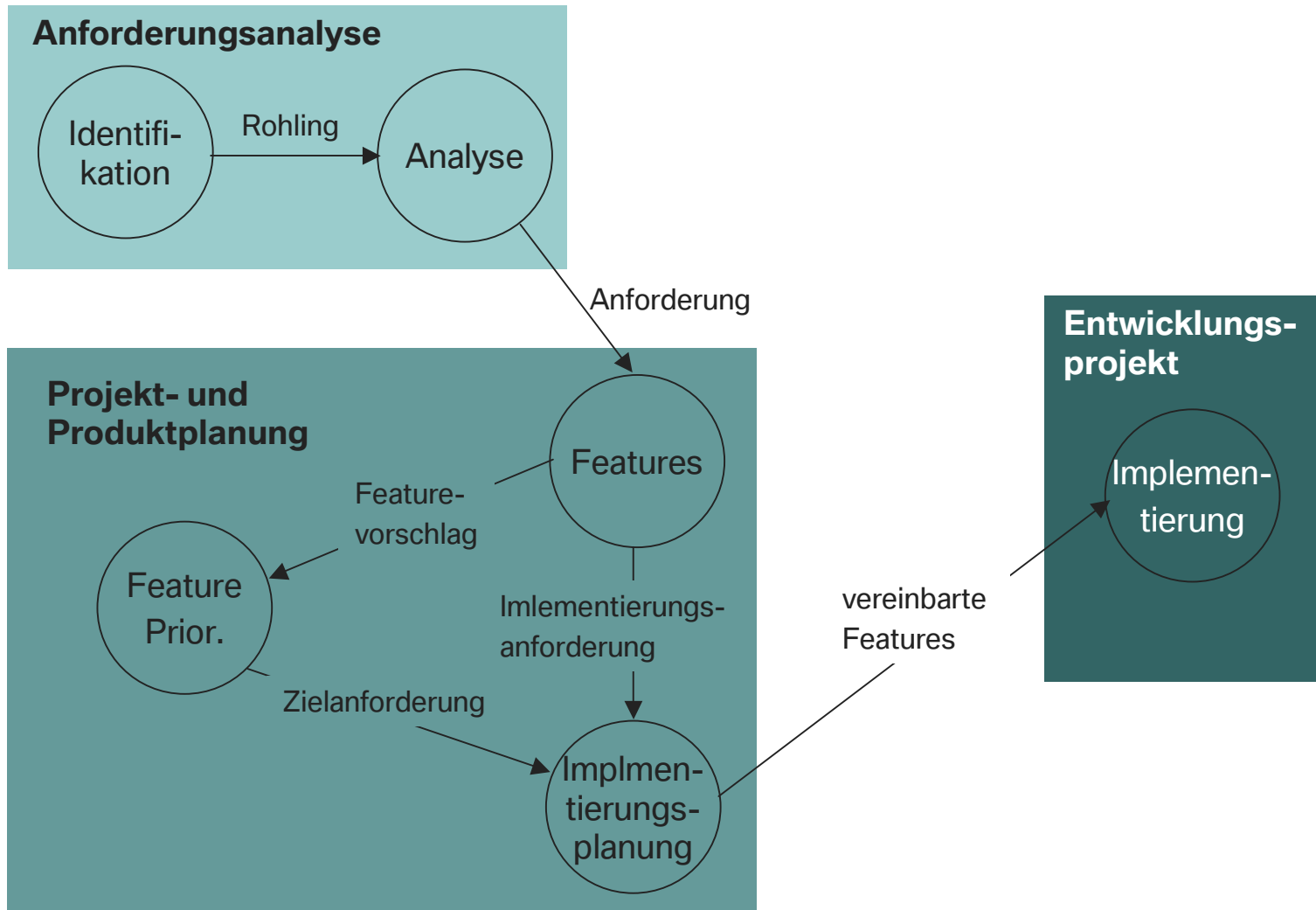
Requirements-Engineering (RE)

RE beschreibt das ingenieurmäßige Vorgehen zur Identifikation, Erhebung, Modellierung, Analyse und Bewertung von Anforderungen.

Anforderungen - Lebenszyklus

- **Idee, Bedürfnis, Wunsch - „Rohling“**
- **Analysierte und verstandene Anforderung - „extrahierte Anforderung“**
- **Vorgeschlagene Anforderung - „Feature Kandidat“**
- **Ausgewählte Anforderung - „abgestimmtes Feature“**
- **Implementierbare Anforderung - „technische Anforderung“**
- **Implementierte Anforderung**
- **Getestete Anforderung**

RE Prozess



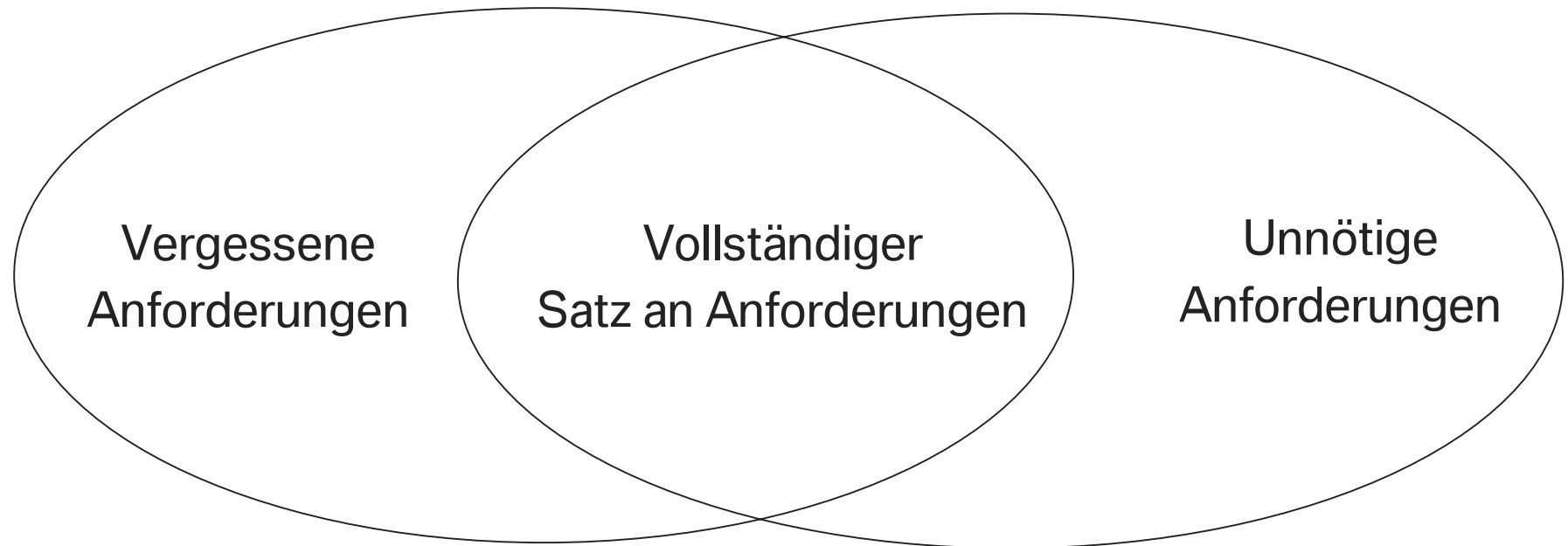
Anforderungserhebung (Requirements Elicitation)

Ziele

- **Gemeinsame Vereinbarung über das Sollverhalten und Funktionalität des Systems**
- **Basis für weiterführende, implementierungsspezifische Spezifikationen**
- **Basis für die Verifikation und Validierung des Systems**
- **Basis für die betriebswirtschaftliche Bewertung des Projektes**

Anforderungserhebung - Abgrenzung

Die Kunst die richtige Anforderungsmenge zu finden



**Das „falsche“
Produkt**

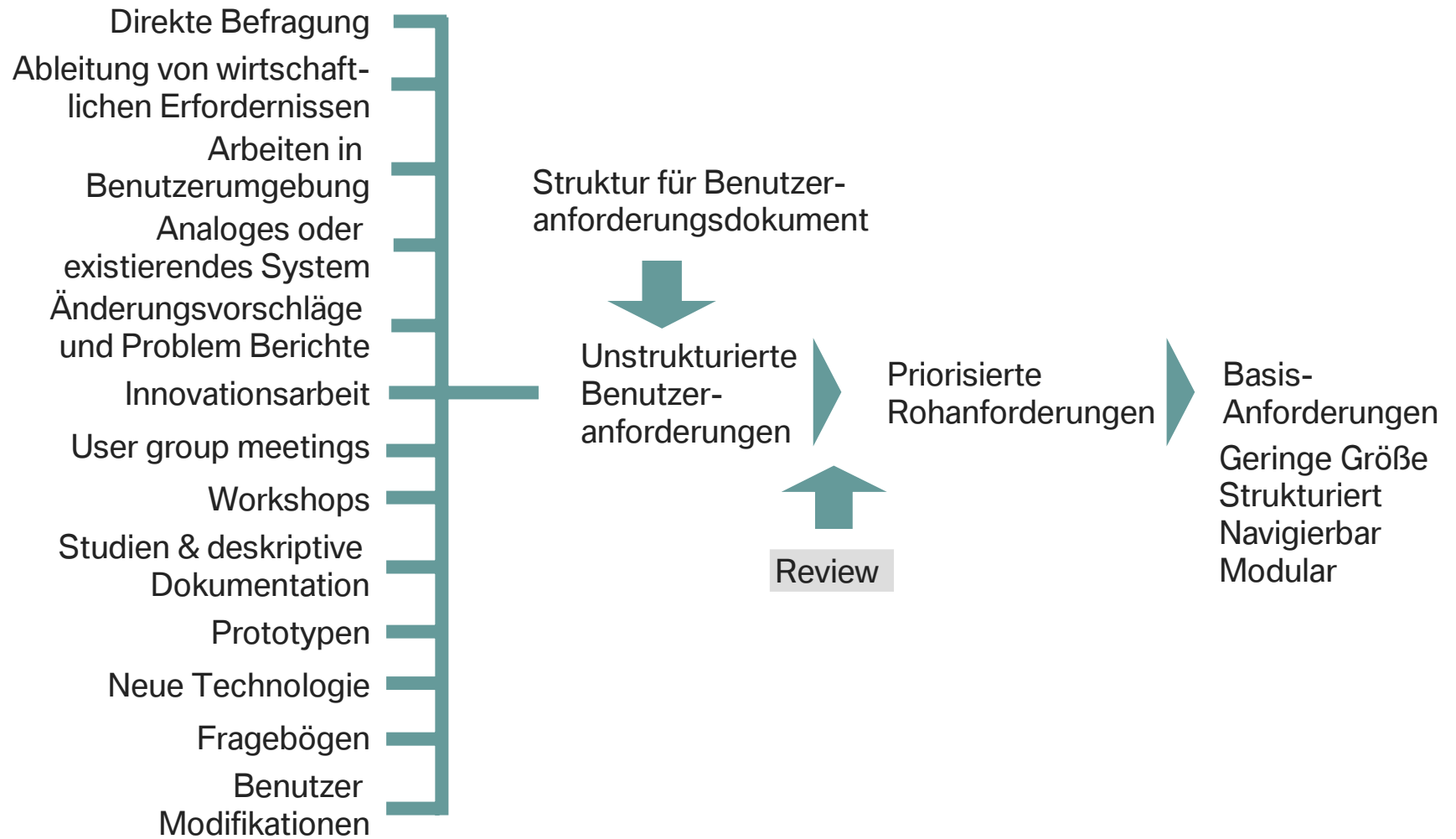
**Das „richtige“
Produkt**

**Das „vergoldete“
Produkt**

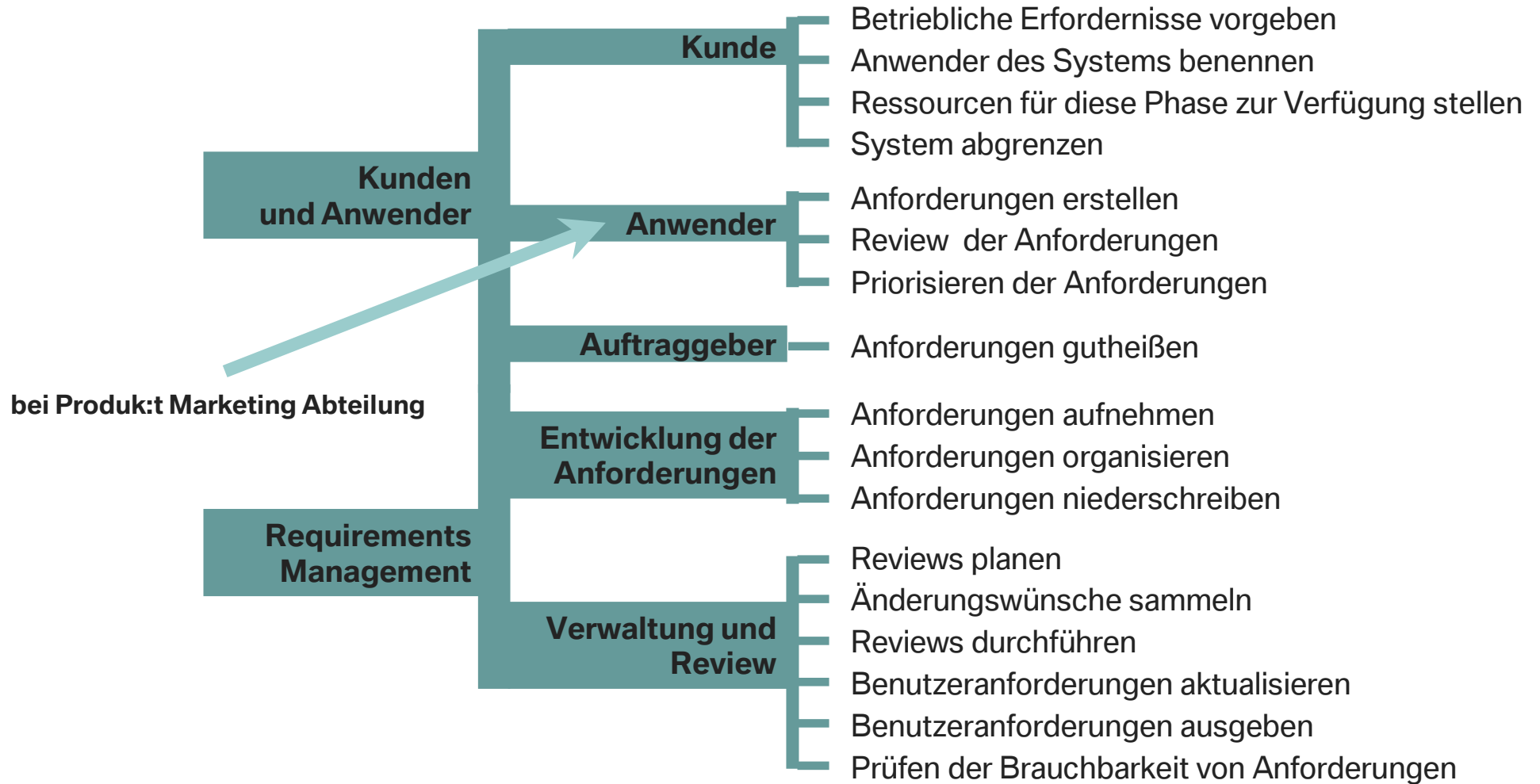
Mythen

- **Anforderungen beschreiben das „Was“, nicht das „Wie“**
In der Praxis lassen sich beide nicht immer trennen
- **Nur der Kunde hat Anforderungen, nur seine Anforderungen können der Treiber für die Spezifikation sein**
Die Implementierung hat oft Rückwirkungen
- **Alle formulierten Anforderungen müssen implementiert sein**
Anforderungen sind priorisierbar

Woher kommen Benutzeranforderungen?



Verantwortlichkeiten in der Benutzer Anforderungs Phase



Methoden der Anforderungserhebung

Anforderungserhebung

- **Beobachtung**
- **Unstrukturierte Interviews**
- **Strukturierte Interviews**
- **Protokoll Analyse**
- **Brainstorming**
- **Rapid Prototyping**
- **Brainstorming**
- **Szenario Analyse**
- **Ethnographische Methoden**

Interviews

Unstrukturierte Interviews

- Lange Diskussionen
- Einfache Methode
- Leicht ablenkbar

(Semi-)Strukturierte Interviews

- Interviews mit einem vorbereiteten Muster und Struktur
- Vorbereitung notwendig (Fragenkatalog)
- Systematisch und konsistent über alle Interviews
- Kann durch falsche Struktur vorgeprägt sein

Beobachtung

- **Beobachtung des tatsächlichen Verhaltens eines Vergleichssystems**
 - Setzt voraus, daß die Anforderungen beobachtbar sind
 - Potentiell werde große Mengen an irrelevanten Daten erhoben
 - Verfügbarkeit der Beobachtungsmittel

Protokollanalyse

- **Benutzer spricht während er die Tätigkeit ausführt**
 - Geeignete, existierende Tätigkeiten müssen vorhanden sein
 - Erlaubt Zugriff auf reale Vorgehensweisen
 - Erlaubt Zugriff auf Gedankengänge des Benutzers
 - Wirkt künstlich

Brainstorming

- **Eine Gruppe schlägt frei Ideen vor**
 - Kann eine große Bandbreite von Anforderungen abdecken
 - Schwer zu fokussieren
 - Produziert unstrukturierte Ergebnisse

Ethnographische Methoden

- **Ein Anforderungs-Ingenieur verbringt Zeit in der Organisation des Nutzers**
 - Erfordert Zeit und Aufwand
 - Bringt tiefes Verständnis der Anwendungsdomäne und der Anforderungen
 - Kann vorurteilsbehaftet sein
 - Seltene Situationen werden nicht erfasst

QFD

- **Ein in Japan entwickelter Ansatz um Kundenanforderungen zu verstehen und zu priorisieren**
 - Basiert auf der Idee, Qualitätsattribute zu identifizieren um diese bezüglich des Mehrwertes für den Kunden zu untersuchen

Rapid Prototyping

- **Der Nutzer wird gebeten einen Prototypen zu kommentieren**
 - Erfordert mächtige Entwicklungstools
 - Macht das System konkret und für den Nutzer faßbar
 - Erfordert viel Zeit
 - Architektur nicht optimal
 - Kann in Anforderungsänderungen münden

Szenario Analysen

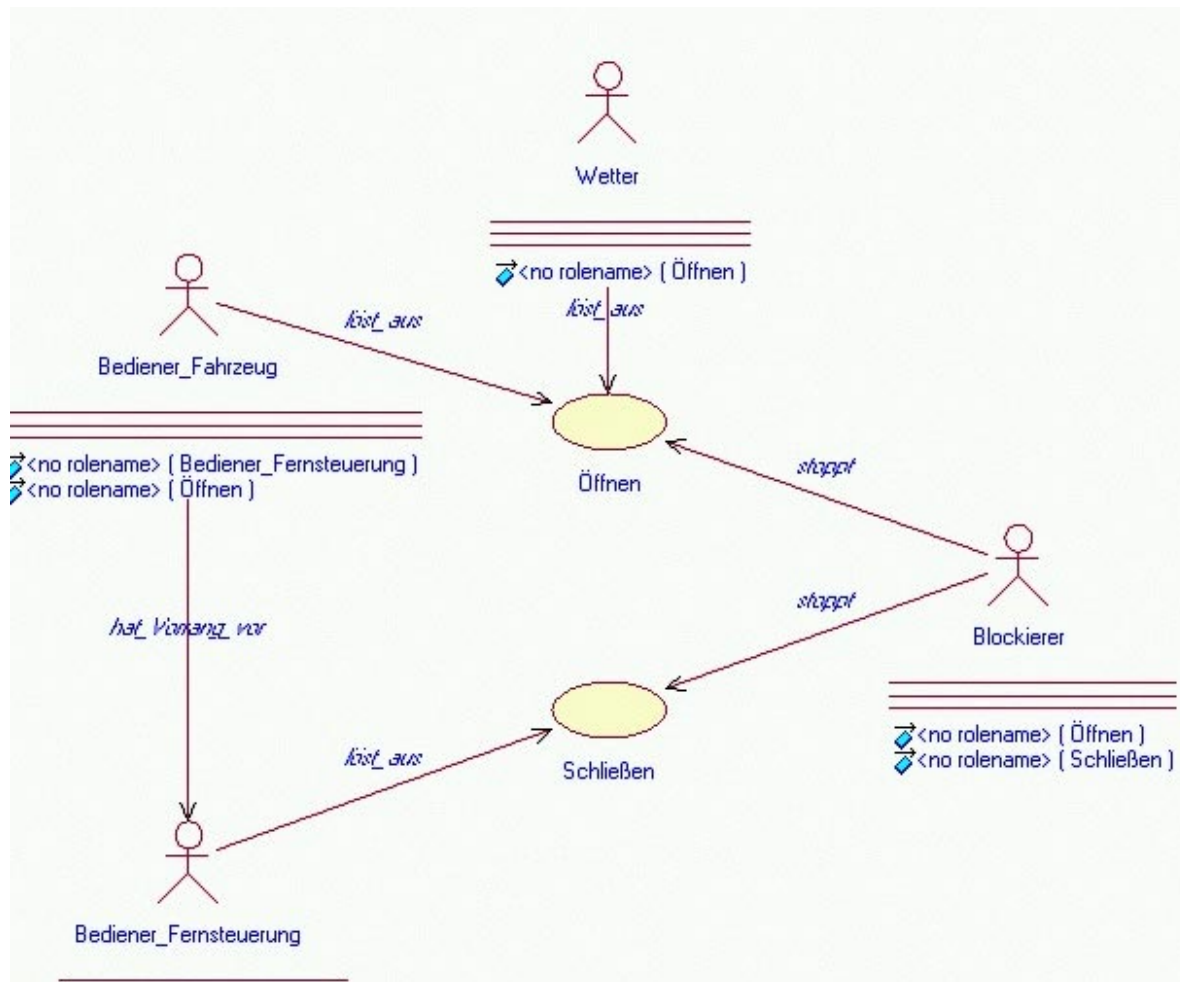
- **Beschreibung einer Sequenz von Aktionen.
Analyse der Sequenz (z.B. Use Cases)**
 - Es muss ein Satz von repräsentativen Szenarios generiert werden
 - Konkret und praktikabel
 - Erfordert viel Zeit

Spezifikationstechniken

Spezifikationstechniken

Use Cases

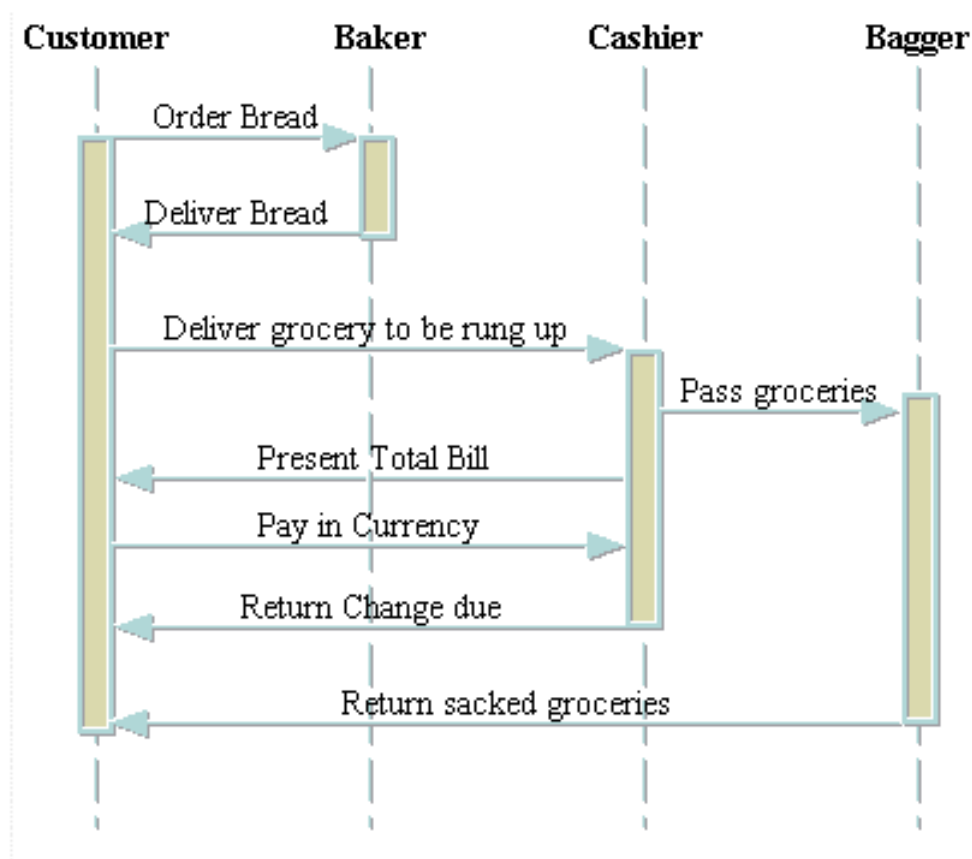
- Beschreibung der (Nutzer-)interaktionen mit dem System



Spezifikationstechniken

Sequence Charts

- **Beschreibung der (zeitlichen) Abfolge von Ereignissen z.B. aus Nutzersicht**

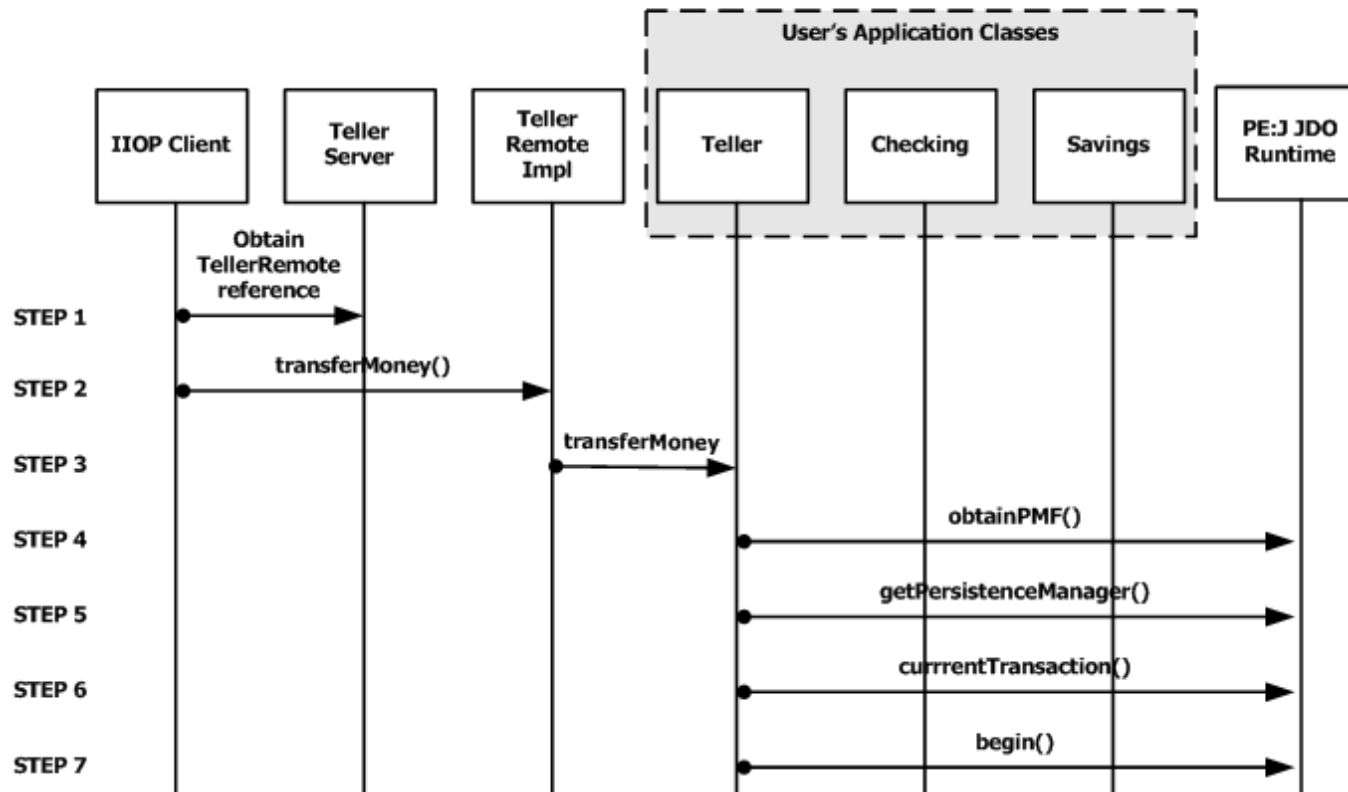


Spezifikationstechniken

Message Sequence Charts

- Beschreibung des zeitlichen Ablaufs von Kommunikation bzw. Datenflüssen zw. Objekten

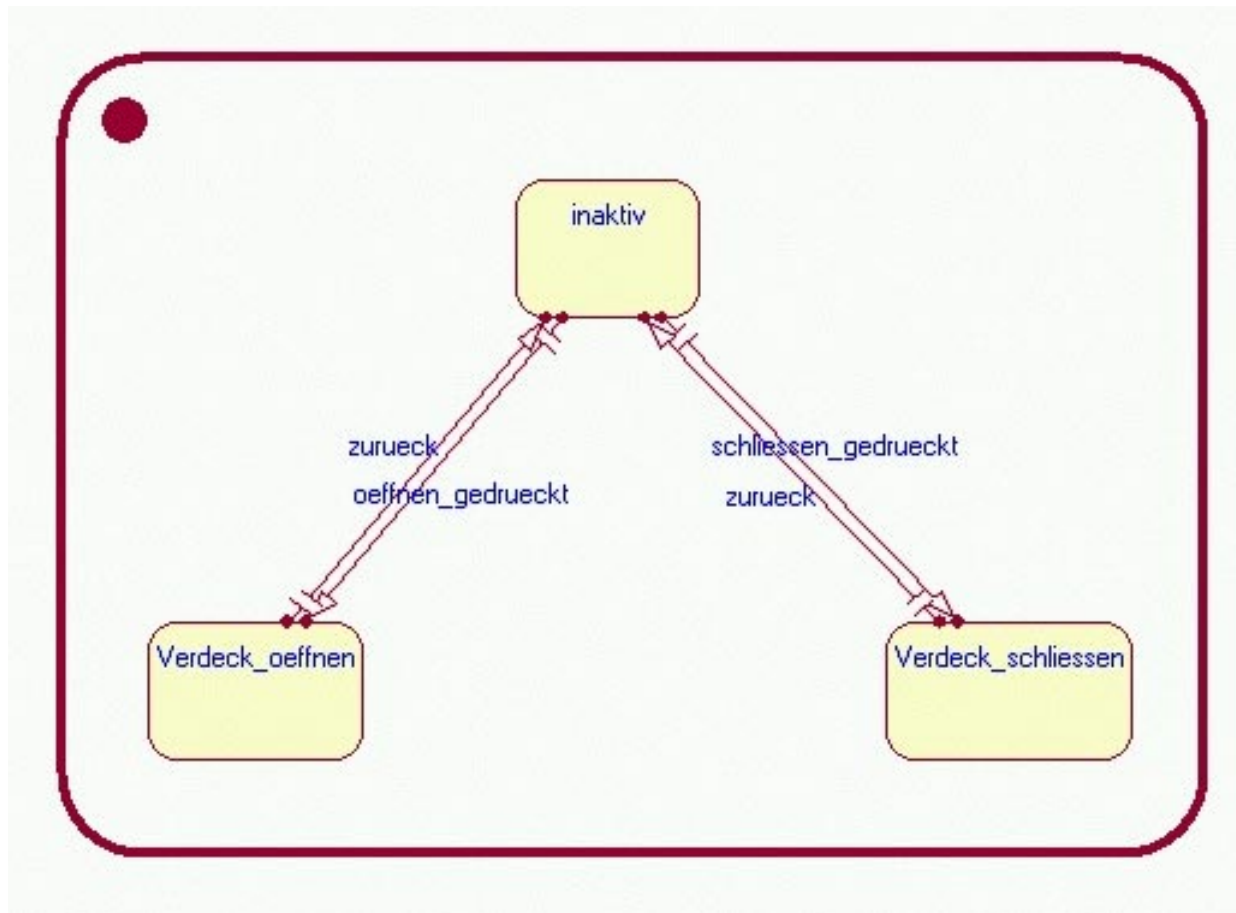
Transfer Money Object Interaction Diagram



Spezifikationstechniken

State Charts

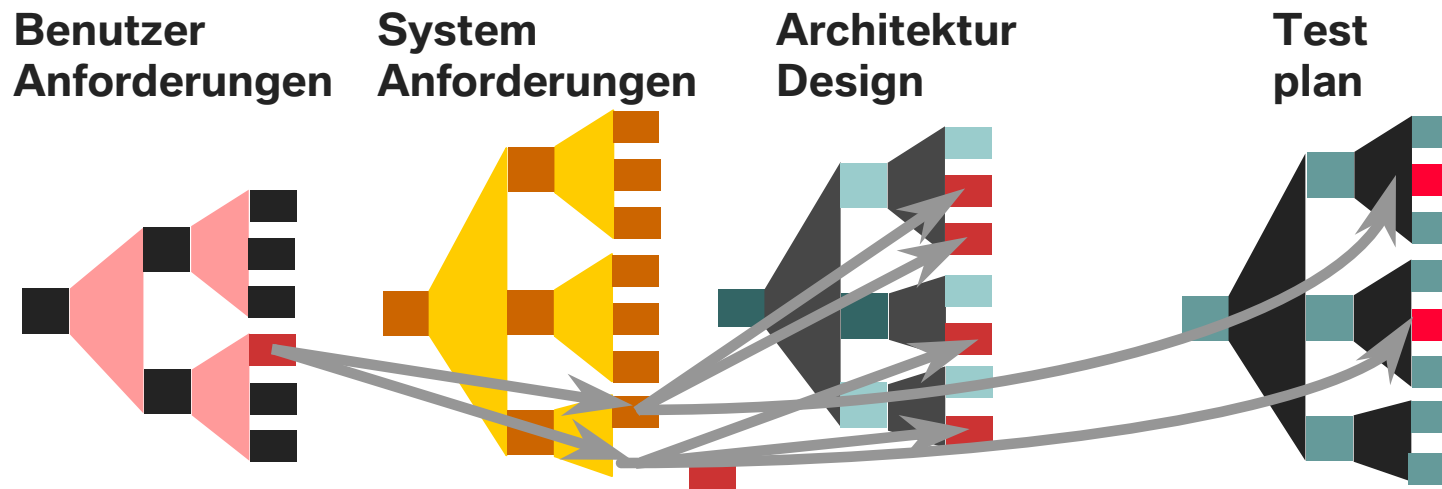
- **Beschreibung der internen Zustände und der Übergangsbedingungen**



Anforderungsverfolgung (Traceability)

Traceability

- **Sicherstellen der Verfolgbarkeit einer Anforderung über den Lebenszyklus**
- **Sicherstellen der Abbildungen von Anforderungen auf Designobjekte**



Requirements-Management

Definition

Requirements-Management (RM)

RM beschreibt die projekthafte Einbettung des RE als Teil des übergeordneten Systems Engineering Prozesses

Demo

Anforderungen

Spezifikationstechniken

Traceability

Zusammenfassung

Zusammenfassung

- Requirements formen das Rückgrat jeder komplexen Entwicklungsaktivität
- Je komplexer das System, umso größer die Notwendigkeit zur guten Verwaltung der Anforderungen
- Requirements bilden die Basis für:
 - die Planung des Projekts
 - Risiko Management
 - Akzeptanz Tests
 - Verhandlungen mit Auftraggeber
 - Änderungskontrolle
- Gutes Requirementsengineering spart Geld
„Qualität ist gratis dabei“