

Inhalt der Vorlesung

- Einführung
- Kommunikation
- Software Qualität
- Vorgehensmodelle
- Requirements Engineering
- Software Architektur und – Design
- Konfiguration Management

- Pohl, Rupp: Basiswissen Requirements Engineering, dpunkt.verlag, 3. Auflage, 2011
Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level
- C.Rupp&die Sophisten: Requirements-Engineering und –Management, 6.Auflage, Hanser

- Requirements Engineering – was ist das
- Bedeutung des Requirements Engineering
- Anforderungen
- Aufgaben des RE
 - Systemkontext ermitteln
 - Anforderungen ermitteln
 - Anforderungen dokumentieren
 - Anforderungen prüfen und abstimmen
 - Anforderungen verwalten
- Toolunterstützung

- Requirements Engineering – was ist das
- Bedeutung des Requirements Engineering
- Anforderungen
- Aufgaben des RE
 - Systemkontext ermitteln
 - Anforderungen ermitteln
 - Anforderungen dokumentieren
 - Anforderungen prüfen und abstimmen
 - Anforderungen verwalten
- Toolunterstützung

Definition Requirements Engineering

Das Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit den folgenden Zielen:

1. Die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen, die Anforderungen konform zu vorgegebenen Standards zu dokumentieren und die Anforderungen systematisch zu managen.
2. Die Wünsche und Bedürfnisse der Stakeholder zu verstehen, zu dokumentieren sowie die Anforderungen zu spezifizieren und zu managen, um das Risiko zu minimieren, dass das System nicht den Wünschen und Bedürfnissen der Stakeholder entspricht.

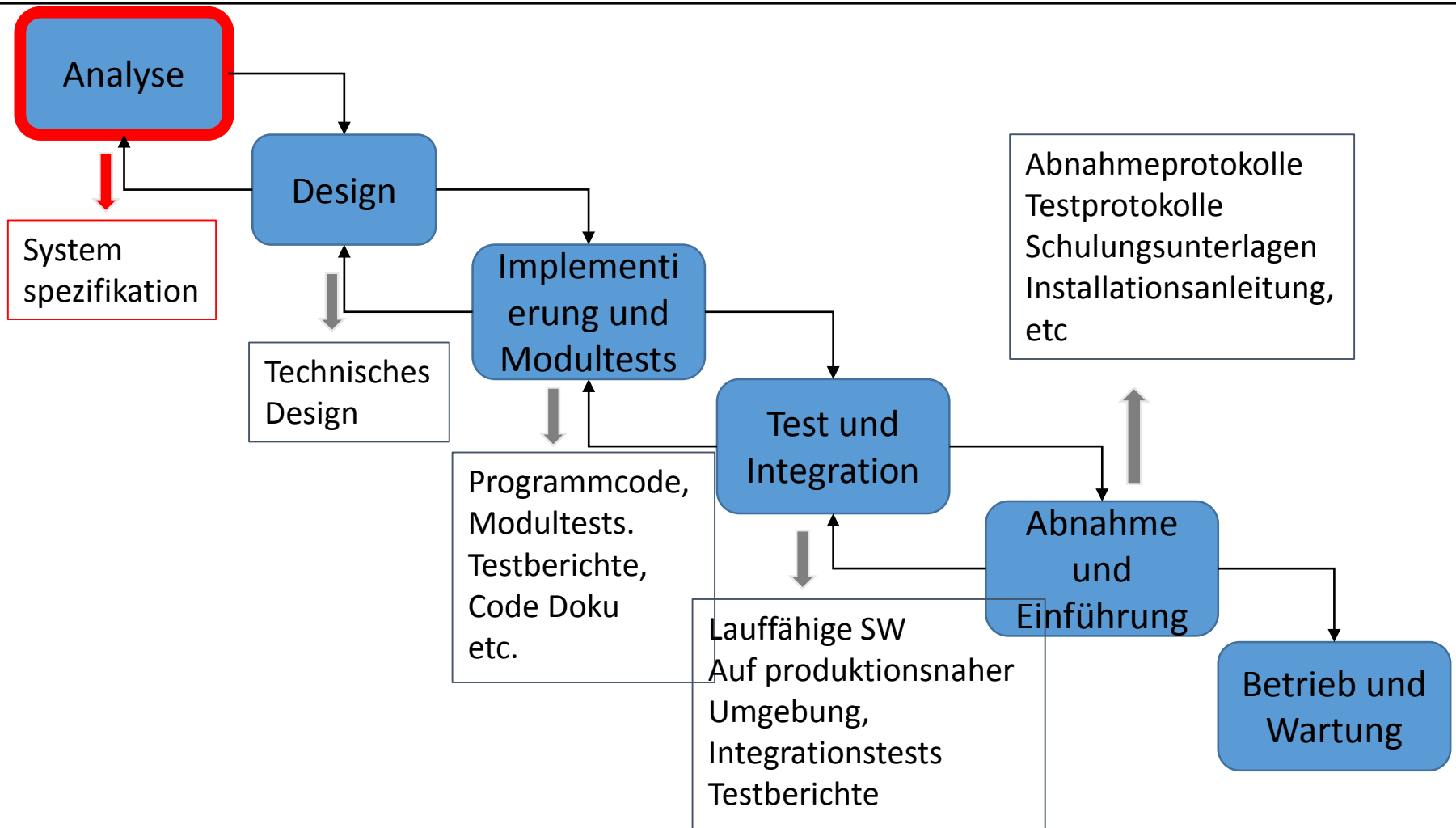
Aus: Pohl, Rupp: Basiswissen Requirements Engineering

Aus der Definition folgen bereits die Haupttätigkeiten des RE:

- (0) Systemkontext ermitteln
- Ermitteln
- Dokumentieren
- Prüfen und abstimmen
- Verwalten

- **Wasserfall:** Eigene Projektphase
- **Agile Prozesse:** kontinuierlicher phasenübergreifender Prozess, in die Systementwicklung integriert.

RE im Wasserfall



Nötige Fähigkeiten:

- Analytisches Denken
- Empathie
- Kommunikationsfähigkeit
- Konfliktlösungsfähigkeit
- Moderationsfähigkeit
- Selbstbewusstsein
- Überzeugungsfähigkeit

- Requirements Engineering – was ist das
- Bedeutung des Requirements Engineering
- Anforderungen
- Aufgaben des RE
 - Systemkontext ermitteln
 - Anforderungen ermitteln
 - Anforderungen dokumentieren
 - Anforderungen prüfen und abstimmen
 - Anforderungen verwalten
- Toolunterstützung

- Ca 60 % der Fehler in Systementwicklungsprojekten entstehen im Requirements Engineering (Boehm 1981)
- Fehler im RE, der in der Abnahmephase behoben wird, ist ca. 100 mal so teuer wie wenn er in der RE Phase behoben wird.
- Sinnvolle Planung: 15-30% Aufwand für das RE.

Quelle: Pohl/Rupp

Quelle: Eman, Khaled El; Koru, A. Günes: *A Replicated Survey of IT Software Project Failures*, in IEEE Software, September/October 2008, S.84-90

- Viele Software Projekte scheitern
- Hauptgründe für Projektabbruch:
 1. **Änderungen der Anforderungen** und des Umfangs
 2. Mangelnde Einbindung des höheren Managements
 3. Engpass im Budget
 4. Fehlende Projektmanagement Fähigkeiten

Nicht: Technische Probleme

- Requirements Engineering – was ist das
- Bedeutung des Requirements Engineering
- Anforderungen
- Aufgaben des RE
 - Systemkontext ermitteln
 - Anforderungen ermitteln
 - Anforderungen dokumentieren
 - Anforderungen prüfen und abstimmen
 - Anforderungen verwalten
- Toolunterstützung

Anforderungen

Anforderungen legen fest, was man von einem Softwaresystem als Eigenschaften erwartet.

Arten von Anforderungen

Anforderung

```
graph TD; A[Anforderung] --> B[Funktionale Anforderung]; A --> C[Qualitätsanforderung]; A --> D[Randbedingung];
```

Funktionale Anforderung

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

Qualitätsanforderung

Anforderung, die sich auf ein Qualitätsmerkmal bezieht, das nicht durch funktionale Anforderungen abgedeckt wird

Randbedingung

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

Gefahr:

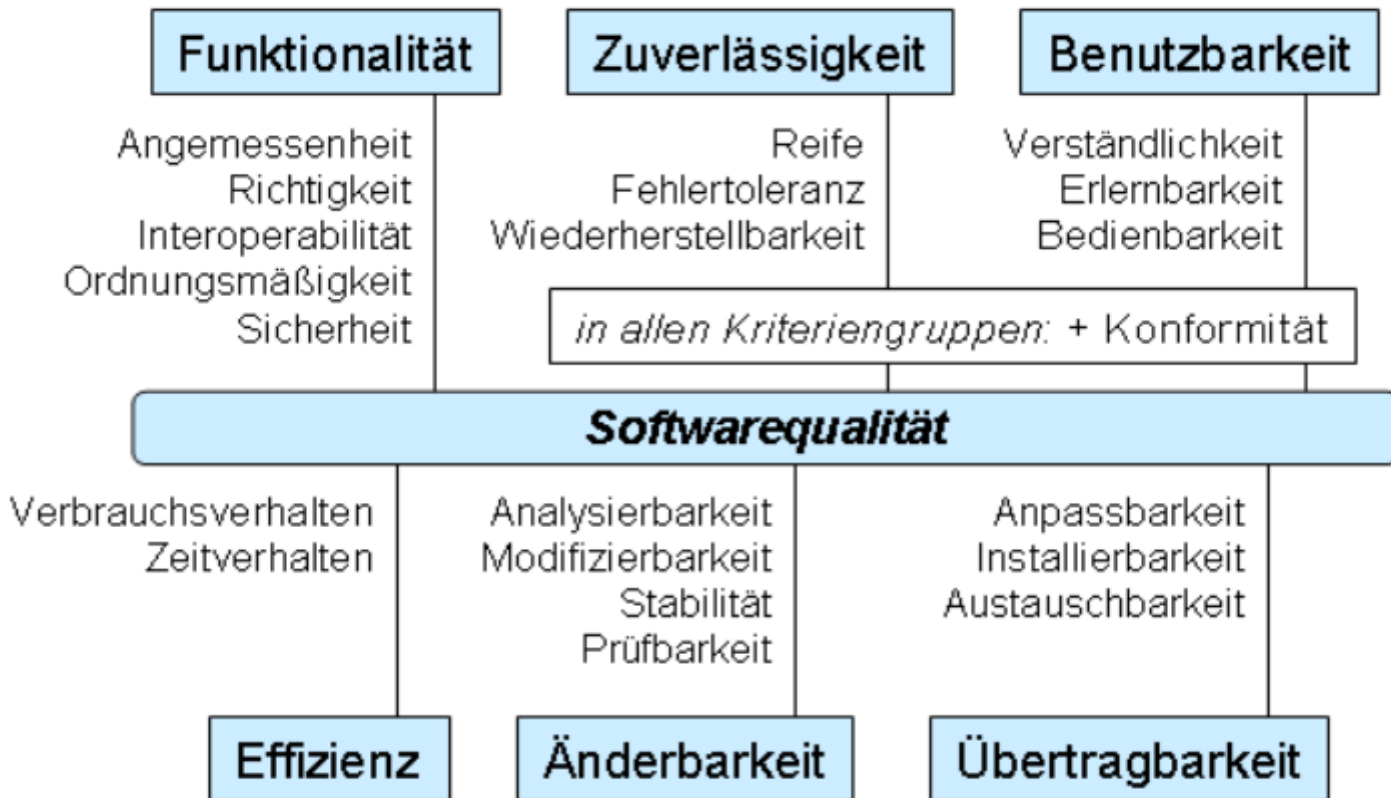
- Konzentration auf die funktionalen Anforderungen
- Unzureichende Dokumentation der Qualitätsanforderungen
- Mangelnde Abstimmung der Qualitätsanforderungen mit den Stakeholdern

➔ Entgegenwirken durch strukturierte Herangehensweise, z.B.

- Kategorisierung nach ISO (siehe nächste Folie) zur Verwendung als Checkliste
- Verwendung von Dokumententemplates
- Verwendung von konkreten Checklisten

6 Qualitätsmerkmale nach ISO/IEC 9126-1

Qualitätsmerkmale von Softwaresystemen (ISO 9126)



Quelle: http://de.wikipedia.org/wiki/ISO/IEC_9126

Qualitätsanforderungen

Qualitätsanforderungen stehen oft mit mehr als einer funktionalen Anforderung in Beziehung (oft beziehen sie sich auch auf das Gesamtsystem).

Daher ist es empfohlen und üblich, die Qualitätsanforderungen in einem eigenen Kapitel des Anforderungsdokuments zu behandeln.

- Requirements Engineering – was ist das
- Bedeutung des Requirements Engineering
- Anforderungen
- Aufgaben des RE
 - (0) Systemkontext ermitteln
 - Anforderungen ermitteln
 - Anforderungen dokumentieren
 - Anforderungen prüfen und abstimmen
 - Anforderungen verwalten
- Toolunterstützung

Systemkontext

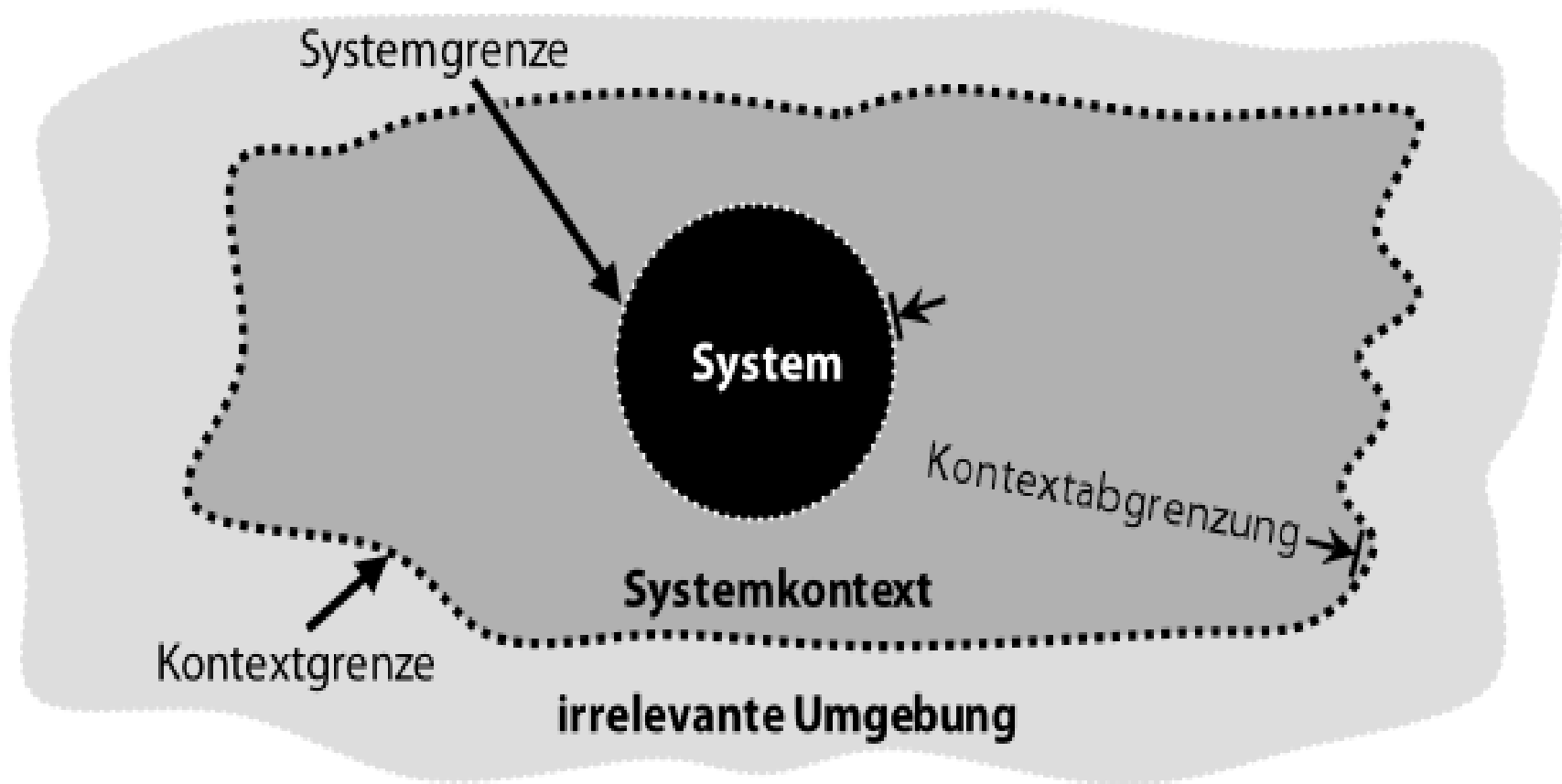
Definition:

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

Aspekte im Systemkontext

- Personen
- Systeme in Betrieb
- Geplante Systeme
- Prozesse
- Ereignisse
- Dokumente
- ...

Systemgrenze/Kontextgrenze



Quelle der Abb: Pohl, Rupp: Basiswissen Requirements Engineering, dpunkt.verlag, 3. Auflage, 2011

Systemgrenze

Die Systemgrenze separiert das geplante System von seiner Umgebung. Sie grenzt den im Rahmen des Entwicklungsprozesses gestaltbaren und veränderbaren Teil der Realität von Aspekten in der Umgebung ab, die durch den Entwicklungsprozess nicht verändert werden können.

Kontextgrenze

Die Kontextgrenze separiert den relevanten Teil der Umgebung eines geplanten Systems vom irrelevanten Teil, d.h. dem Teil der Umgebung, der keinen Einfluss auf das geplante System und damit auch keinen Einfluss auf die Anforderungen dieses Systems hat.

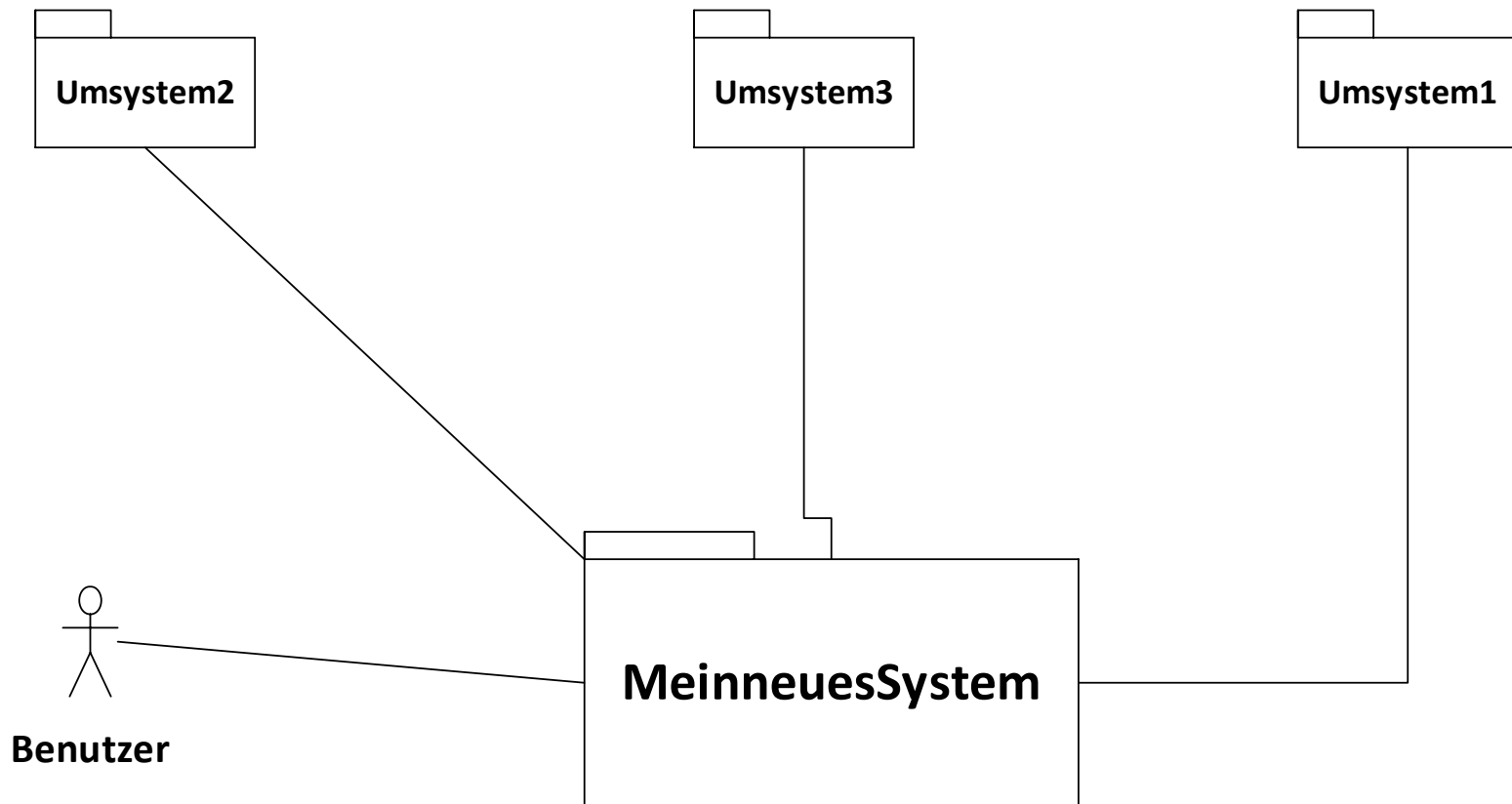
Die Definition dieser Grenzen liegt in der Verantwortung des Requirement Engineers.

Der Systemkontext wird häufig mithilfe verschiedener Diagrammtypen dokumentiert
z.B.

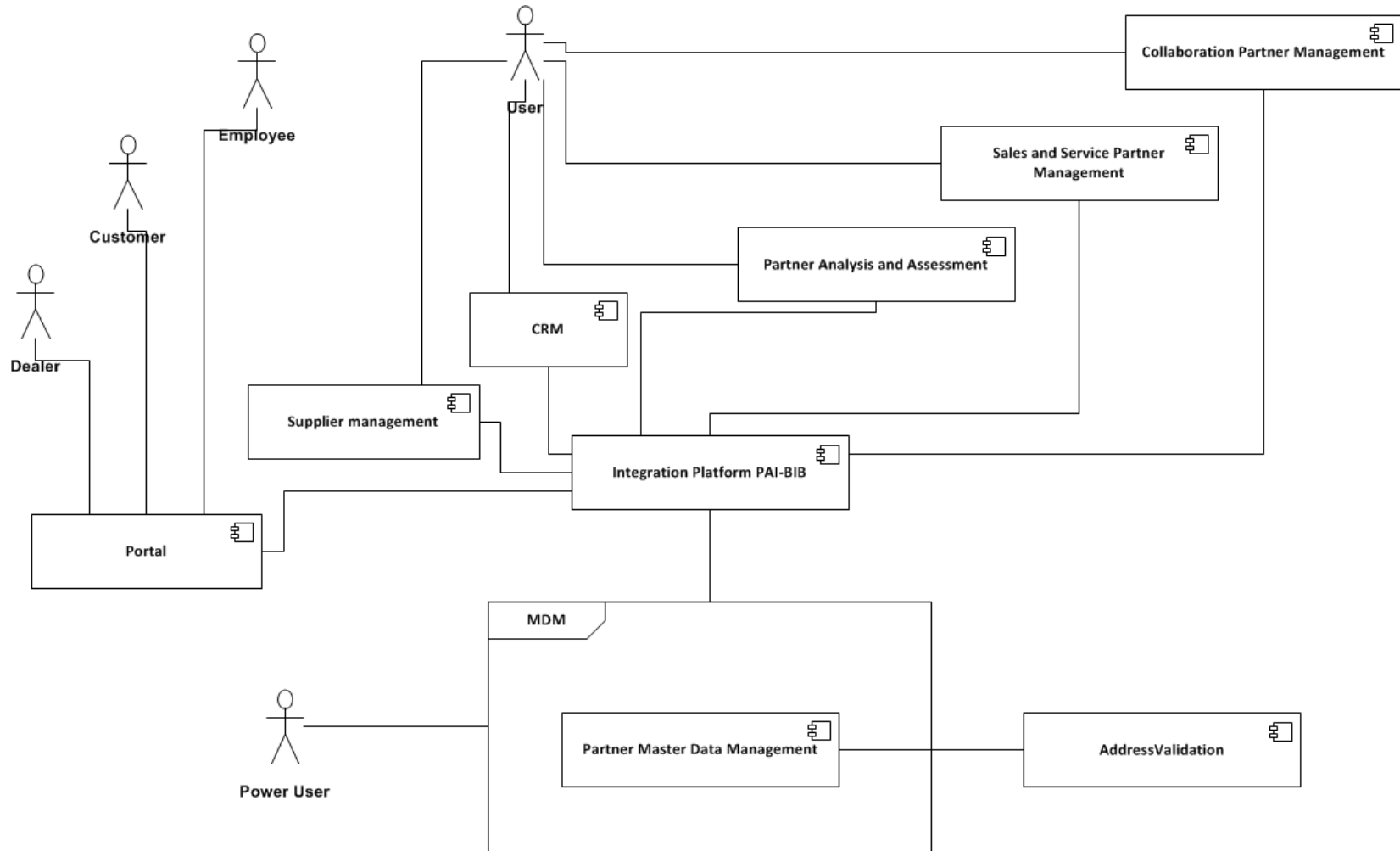
- Paket Diagramme
- Usecase Diagramme
- Komponentendiagramme
- Klassendiagramme
- Nicht formalisierte Grafiken
- ...

Kontext Diagramm - Beispiel

Kontextdiagramm Bsp incl Schnittstellen



Bsp für ein Kontextdiagramm



- Requirements Engineering – was ist das
- Bedeutung des Requirements Engineering
- Anforderungen
- Aufgaben des RE
 - (0)Systemkontext ermitteln
 - Anforderungen ermitteln
 - Anforderungen dokumentieren
 - Anforderungen prüfen und abstimmen
 - Anforderungen verwalten
- Toolunterstützung

Drei Arten von Anforderungsquellen:

- Stakeholder
- Dokumente
- Systeme (in Betrieb oder geplant)

Stakeholder:

Person oder Organisation, die (direkt oder indirekt) Einfluss auf die Anforderungen hat.

Beispiele für Stakeholder:

- Nutzer des Systems
- Betreiber des Systems
- Auftraggeber
- Tester
- (Entwickler)
- (Architekten)

Aufgaben des Requirements Engineers in Hinblick auf Stakeholder:

- Identifikation der Stakeholder
- Sammeln, Dokumentieren, Konsolidieren der Anforderungen der Stakeholder
- Moderation (manchmal auch Mediation) bei widersprüchlichen Anforderungen und/oder Konflikten
- Übereinstimmung erzielen



1. Effektiver Umgang (Betroffene zu Beteiligten machen)
2. Verwendung von Tools und Formalismen

1. Effektiver Umgang (Betroffene zu Beteiligten)
2. Verwendung von Tools und Formalismen

Stakeholder Vereinbarung

Um Missverständnissen, Kompetenzstreitigkeiten etc. vorzubeugen, ist es zweckmäßig, dass der Requirements Engineer mit jedem Stakeholder jeweils formal vereinbart, welche Aufgaben, Verantwortungsbereiche, Weisungsbefugnisse und individuellen Ziele bestehen und welche Kommunikationswege und Feedback-Schleifen von dem Stakeholder genutzt werden können.

Je nach Unternehmenskultur kann diese Vereinbarung in einfacher Form durch eine mündliche Vereinbarung oder auch formeller ausgestaltet und schriftlich getroffen werden.

Die Vorgesetzten sollten den jeweiligen Vereinbarungen zustimmen.

Stakeholder Vereinbarung

*Quelle: Pohl, Rupp: Basiswissen
Requirements Engineering,
dpunkt.verlag, 3. Auflage, 2011*

Der Requirements Engineer

- spricht die Sprache der Stakeholder,
- arbeitet sich in das Fachgebiet gründlich ein,
- erstellt ein Anforderungsdokument,
- kann die Arbeitsergebnisse verständlich machen (z.B. mithilfe von Diagrammen oder Grafiken),
- pflegt einen respektvollen Umgang mit den Stakeholdern,
- präsentiert seine Ideen und Alternativen und deren Realisierung,
- ermöglicht es den Stakeholdern, Eigenschaften zu fordern, die das System einfach und benutzerfreundlicher machen,
- sorgt dafür, dass das spezifizierte System den funktionalen und qualitativen Ansprüchen der Stakeholder gerecht wird.

Stakeholder Vereinbarung

Der Stakeholder

- führt den Requirements Engineer in das Fachgebiet ein,
- versorgt den Requirements Engineer mit Anforderungen,
- formuliert die Anforderungen zielgerecht und gewissenhaft,
- trifft Entscheidungen zeitgerecht,
- respektiert die Einschätzung der Kosten und Machbarkeit des Requirements Engineer,
- priorisiert die Anforderungen,
- überprüft die dokumentierten Anforderungen des Requirements Engineers, wie Prototypen usw.,
- kommuniziert unverzüglich Änderungen in den Anforderungen,
- befolgt den vorgegebenen Änderungsprozess,
- respektiert das vorgegebene Requirements Engineering.

1. Effektiver Umgang (Betroffene zu Beteiligten)
2. Verwendung von Tools und Formalismen

Stakeholder Liste

- Listet alle für das Projekt relevanten Stakeholder auf
- Beschreibt die Wichtigsten Informationen über jeden einzelnen Stakeholder

- Name/Rolle
- Beschreibung des Stakeholders
- Wissensgebiete
- Konkrete Vertreter
- Verfügbarkeit
- Begründung

Rolle der Stakeholder	Beschreibung	Konkrete Vertreter	Verfügbarkeit	Wissensgebiet	Begründung
Kneipen-Management	Nennt Produkt- und Projektziele	Herr Müller Tel.: 4711 muel-ler@bl.de	5% verfügbar	Kennt alle Vorgänger des Produkts im Detail, da vorher selbst Anwender des Produkts	Entscheidung über Realisierung, Geldgeber
Anwender: Koch	Ist ein Benutzer des Systems (nicht des Mobilteils)	Herr Meier Tel.: 0815 E-Mail: Meier@bl.de	Urlaub vom 20.12.06 bis 07.01.07; 20% verfügbar	Experte der Speisenver-buchung	Anwender des Systems, muss zukünftig Speisen verbuchen
...

Stakeholder Portfolio

Wohllwollen

Beispiel

Schmidt

Huber

Nicht für die Projektunterlagen!

Meier

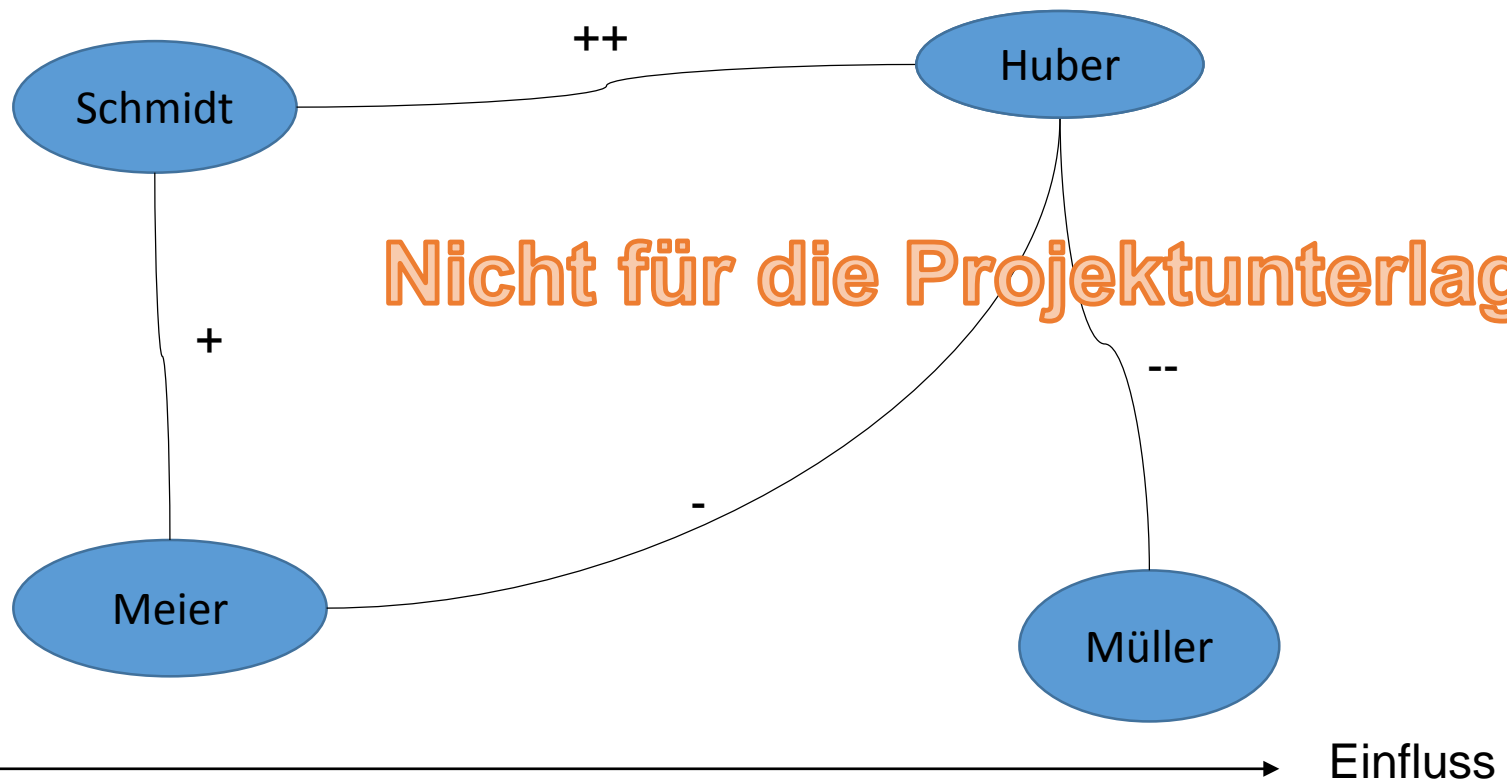
Müller

Einfluss

Stakeholder Portfolio

Wohllwollen

Beispiel



- Befragungstechniken
 - Z.B. Interview, Fragebogen
- Kreativitätstechniken
 - Brainstorming
 - Brainstorming paradox
 - Perspektivenwechsel
 - Analogietechniken
- Dokumentenzentrierte Techniken
 - Systemarchäologie
 - Perspektivenbasiertes Lesen
 - Wiederverwendung

- Beobachtende Techniken
 - Feldbeobachtung
 - Apprenticing
- Unterstützende Techniken
 - Mindmapping
 - Workshops
 - Audio und Videoaufzeichnungen
 - Prototypen
 - ...

Welche Technik soll verwendet werden?

Antwort abhängig von:

- Termin- und Budgetvorgaben
- Verfügbarkeit relevanter Stakeholder
- Erfahrung des Requirements Engineer mit der entsprechenden Ermittlungstechnik
- Chancen und Risiken des Projekts.
- Ermittlung bewusster oder unbewusster Anforderungen

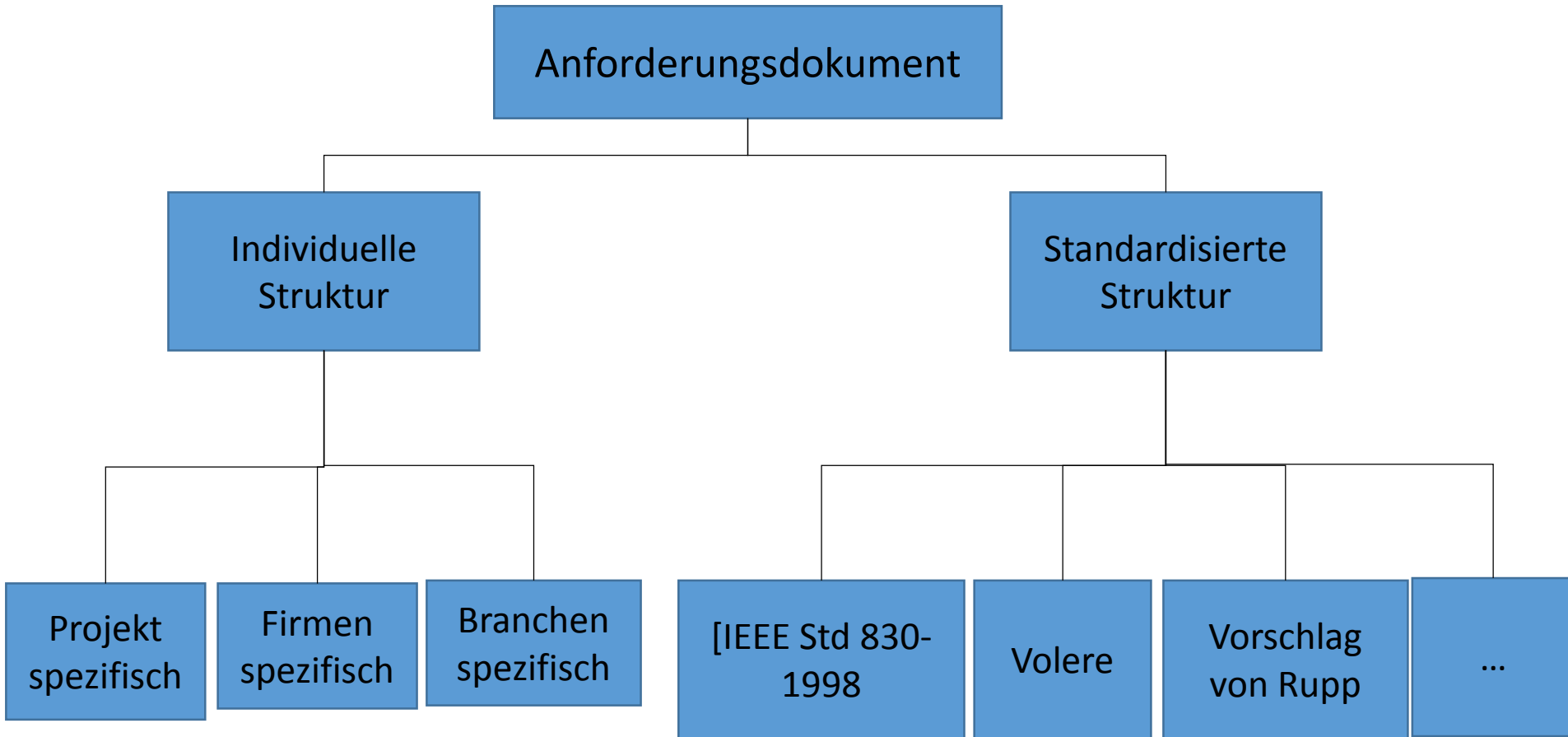
- Requirements Engineering – was ist das
- Bedeutung des Requirements Engineering
- Anforderungen
- Aufgaben des RE
 - Systemkontext ermitteln
 - Anforderungen ermitteln
 - Anforderungen dokumentieren
 - Anforderungen prüfen und abstimmen
 - Anforderungen verwalten
- Toolunterstützung

- Dokumentation von Zwischenschritten
 - Interviewprotokolle
 - Protokolle zu Abstimmungsaktivitäten
 - Änderungsanträge
- **Anforderungsdokument/Anforderungsspezifikation**
 - Eine Anforderungsspezifikation ist eine systematisch dargestellte Sammlung von Anforderungen (typischerweise für ein System oder eine Komponente), die vorgegebenen Kriterien genügt.

Gründe für die Dokumentation der Anforderungen

- Zentrale Bedeutung der Anforderungen für die Systementwicklung
- Rechtliche Relevanz
- Komplexität der Anforderungsdokumente
- Zugreifbarkeit

Dokumentenstrukturen



Beispiel Volere Template

Das Volere Template liegt unter „Material“ auf der eLearning Plattform.

Anforderungsdokumente als Grundlage für weitere Arbeiten

- Planung
- Architekturentwurf
- Implementierung
- Test
- Änderungsmanagement
- Systemnutzung und Systemwartung
- Vertragsmanagement

Qualitätskriterien für das Anforderungsdokument

- Eindeutigkeit und Konsistenz [IEEE Std 830-1998]
- Klare Struktur
- Modifizierbarkeit und Erweiterbarkeit [IEEE Std 830-1998]
- Vollständigkeit [IEEE Std 830-1998]
- Verfolgbarkeit [IEEE Std 830-1998]

- Abgestimmt
- Bewertet
- Eindeutig
- Gültig und aktuell
- Korrekt
- Konsitent
- Prüfbar
- Realisierbar
- Verfolgbar
- Vollständig
- Verständlich

Anforderungsspezifikation in der agilen Welt.

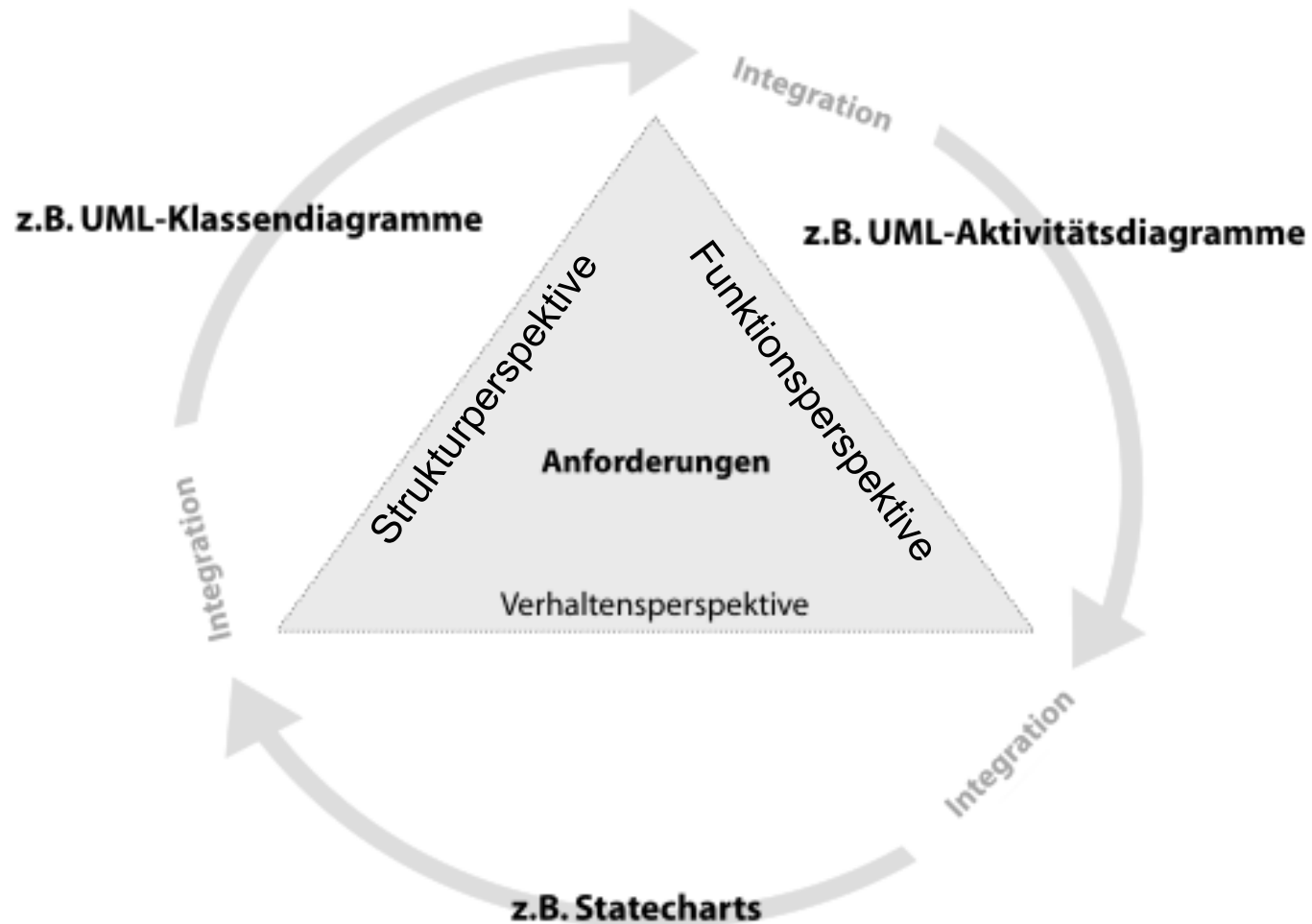
Verwendung des Product Backlogs

- ➔ Product Backlog Items (nicht gleich Anforderungen)
- ➔ Product Backlog Item ist nicht vollständig, bevor es diskutiert wurde

While a product backlog can be thought of as a replacement for the requirements document of a traditional project, it is important to remember that the written part of an agile user story (“As a user, I want ...”) is incomplete until the discussions about that story occur. It’s often best to think of the written part as a pointer to the real requirement. User stories could point to a diagram depicting a workflow, a spreadsheet showing how to perform a calculation, or any other artifact the product owner or team desires.

Aus <https://www.mountangoatsoftware.com/agile/user-stories>

Arten der Dokumentation - Perspektiven



■ **Dokumentation in natürlicher Sprache**

- Vorteile:
 - Kann jeder, insbsd. auch die Stakeholder
- Nachteile
 - Mehrdeutig
 - Perspektiven werden vermischt.

■ **Dokumentation durch konzeptionelle Modelle**

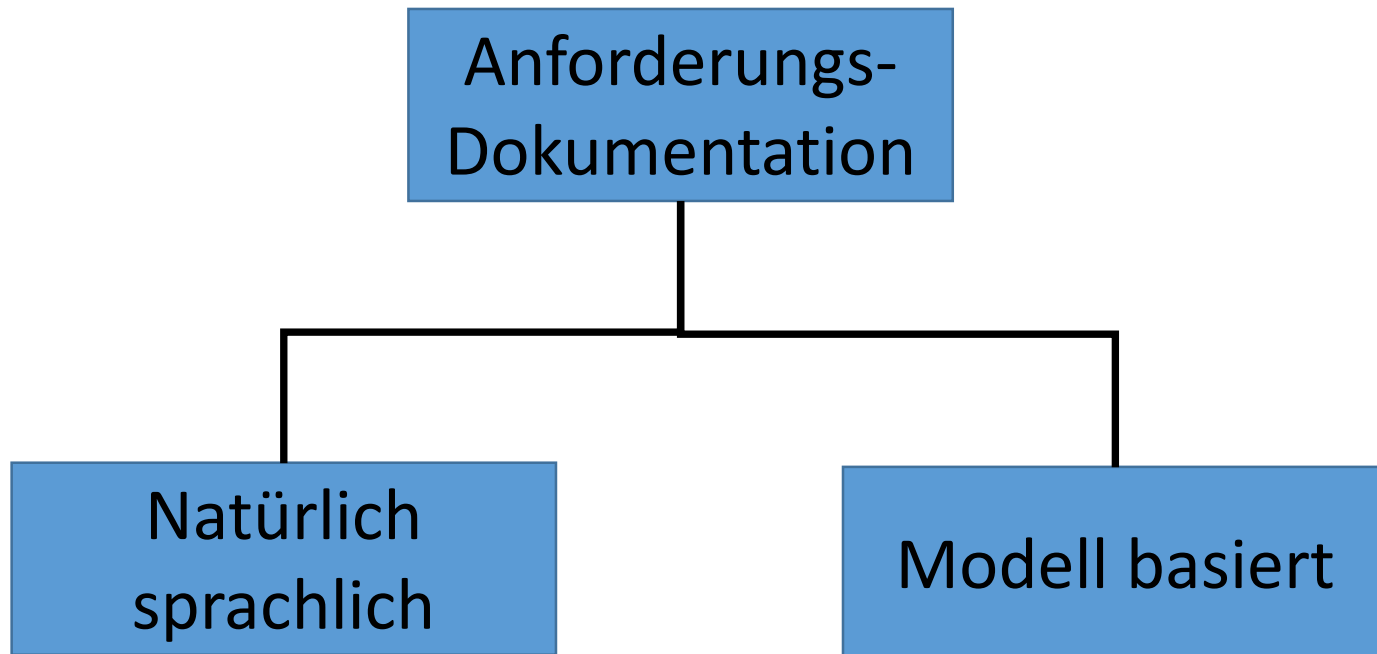
- Vorteile
 - Kompakt
 - Eindeutig
 - Saubere Trennung der Perspektiven

Beispiele der Verwendung

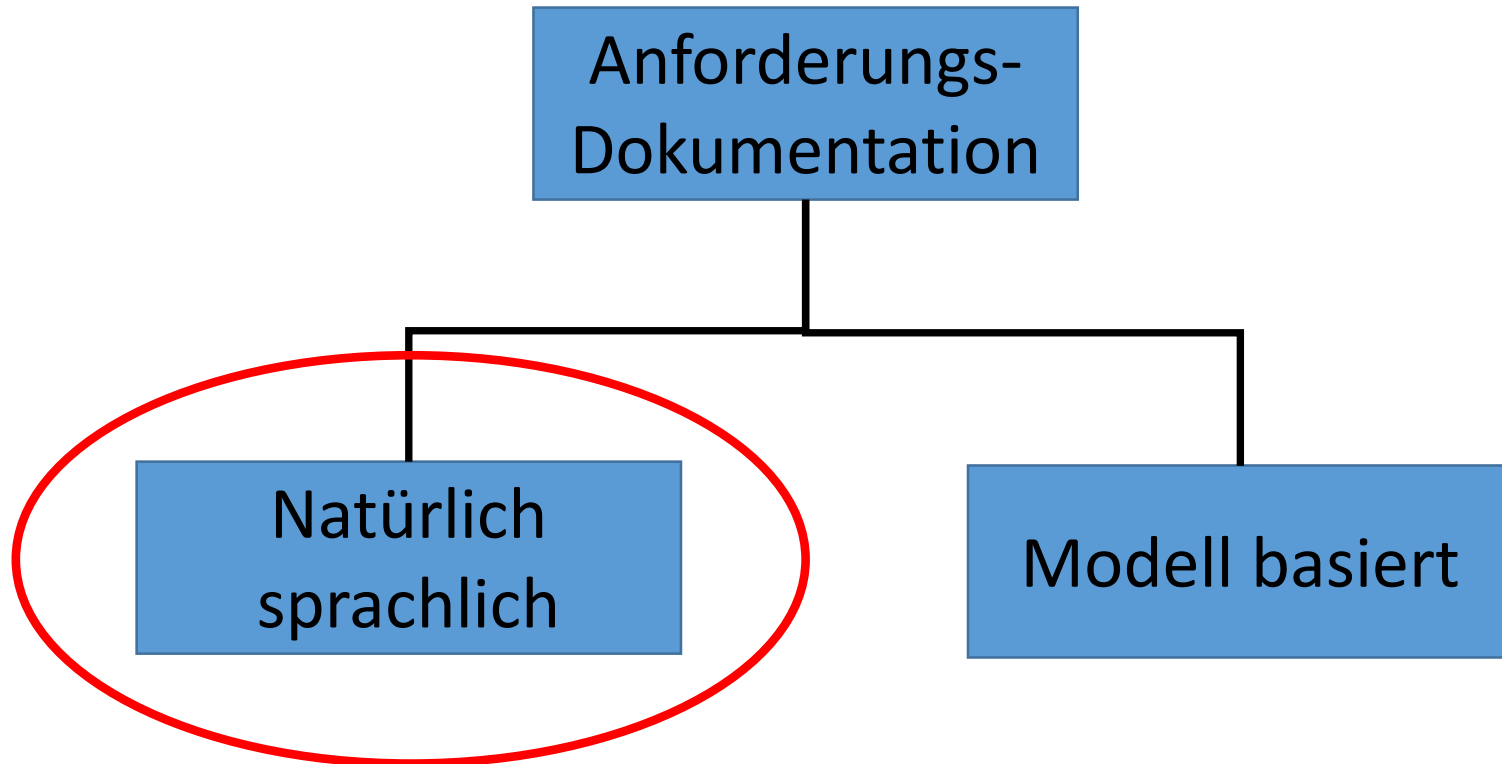
- Überblick über die Systemfunktionalität
 ➔ Usecase Diagramm
- Begriffssysteme und Datenmodellierung
 ➔ Klassendiagramm
- Ablaufmodellierung
 ➔ Aktivitätsdiagramm
- Ereignisgesteuertes Verhalten
 ➔ Zustandsdiagramm

Anforderungen dokumentieren

Siehe oben:



Anforderungen dokumentieren



Nominalisierung: Prozess wird zu Ereignis

Bsp: „Bei einem Systemabsturz soll ein Neustart des Systems erfolgen.“

Zu adressieren:

„Systemabsturz und „Neustart“ sind näher zu beschreibende Prozesse.

Bsp (aus C.Rupp&die Sophisten: Requirements-Engineering und -Management, 6.Auflage, Hanser):
„Das System muss eine Archivierung ermöglichen.“

Besser:

1. „Das Bibliothekssystem muss dem Benutzer die Möglichkeit bieten, einen Kundendatensatz für die Archivierung auszuwählen.“
2. „Falls das Kundenkonto gleich null ist, muss das Bibliothekssystem dem Benutzer die Möglichkeit bieten, den ausgewählten Kundendatensatz zu archivieren.“

Universalquantoren: absolute Angaben über Häufigkeiten, Bsp. „nie“, „immer“, „kein“, „jeder“ ...

Bsp: „Das System soll in jedem Untermenü alle Datensätze anzeigen.“

Zu adressieren:

Wirklich in jedem Untermenü? Wirklich alle Datensätze?

Unvollständig spezifizierte Bedingungen:

Bsp: „Das Restaurantsystem soll einem registrierten Gast bei einem Alter von über 17 Jahren alle im Lokal angebotenen Getränke anzeigen.“

Zu adressieren:

Welche Getränke soll das System einem Gast anzeigen, der 17 Jahre oder jünger ist?

Unvollständig spezifizierte Verben, oftmals durch Verwendung des Passiv:

Bsp: „Zur Anmeldung des Benutzers werden die Login-Daten eingegeben.“

Zu adressieren:

Wer wo und wie die Login-Daten eingeben kann.

Passiv Bsp

Bsp:

„Jedem Leihobjekt muss ein eindeutiger Standort in der Bibliothek zugewiesen werden.“

Besser:

„Der Bibliothekar muss jedem Leihobjekt einen eindeutigen Standort in der Bibliothek zuweisen.“

Natürlichsprachliche Effekte sind unter anderem

- Nominalisierung
- Universalquantoren
- Unvollständig spezifizierte Bedingungen
- Unvollständig spezifizierte Prozesswörter

Wenn der Req Engineer diese Effekte kennt, dann kann er gezielt darauf eingehen.

Reduzierung sprachlicher Effekte

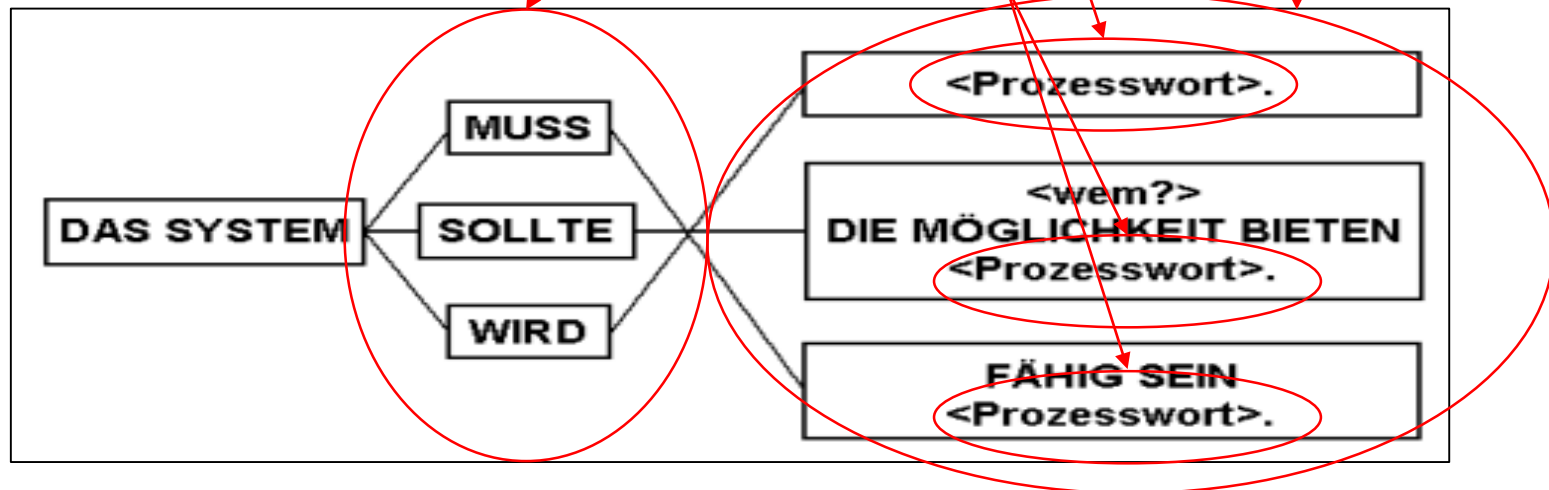
Die besprochenen negativen sprachlichen Effekte können durch die Verwendung von Satzschablonen reduziert werden.

Eine Satzschablone (Requirements Template) ist ein Bauplan für die syntaktische Struktur einer einzelnen Anforderung.

Verwendung einer Satzschablone

- **Schritt 1:** Festlegen der rechtlichen Verbindlichkeit
- **Schritt 2:** Der Kern der Anforderung.
- **Schritt 3:** Charakterisieren der Aktivität des Systems

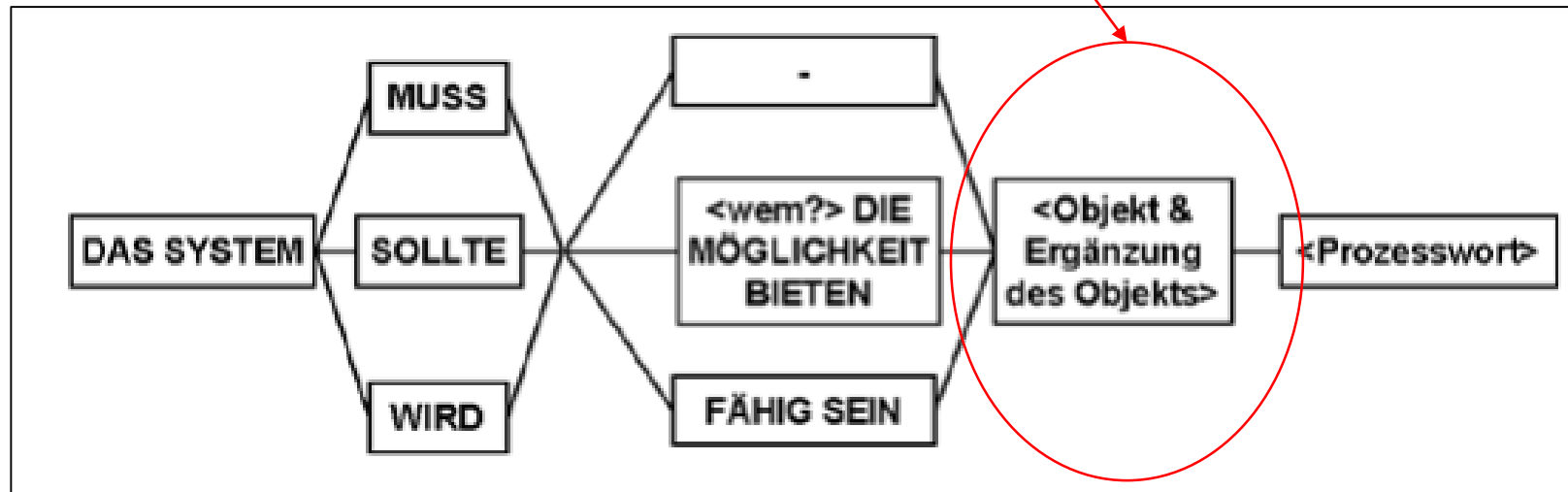
➔ Grundgerüst der Anforderung steht und kann in die Satzschablone gefüllt werden.



Verwendung einer Satzschablone

• Schritt 4: Objekte einfügen

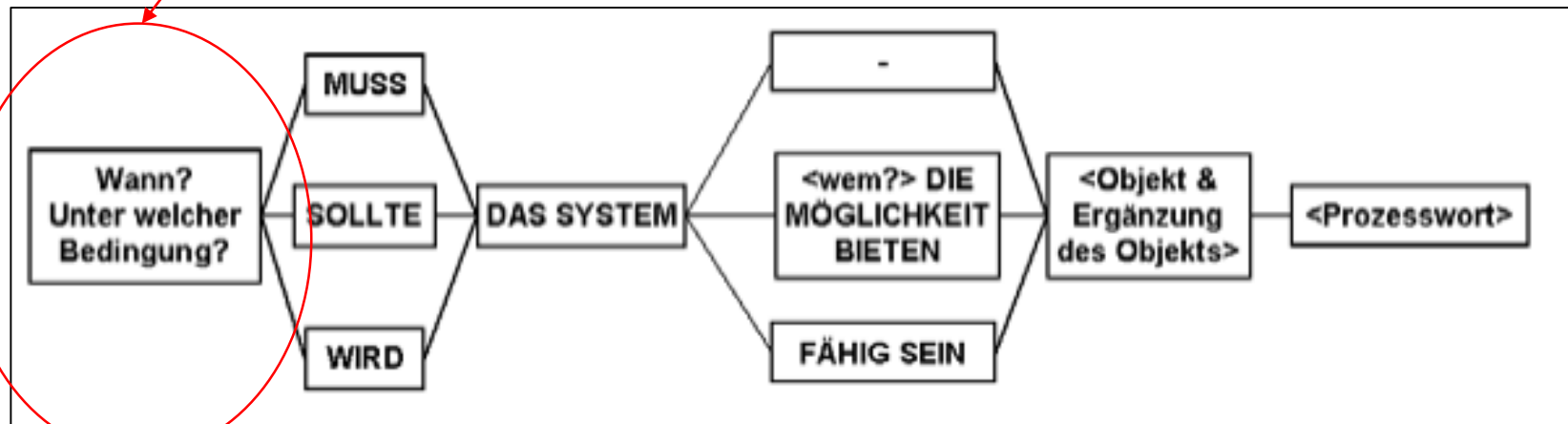
➔ Vollständige Satzschablone ohne Bedingungen



Verwendung einer Satzschablone

- Schritt 5 Formulierung von logischen und zeitlichen Bedingungen

➔ Vollständige Satzschablone inkl. Bedingungen



Mike Cohn hat folgende Struktur für **User Stories** eingeführt
(<https://www.mountaingoatsoftware.com/agile/user-stories>):

As a <type of user>, I want <some goal> so that <some reason>.

Übernommen von B. Gloger:

Als Anwender <mit der Rolle> benötige ich eine <Funktionalität>, damit ich den <Nutzen> become.

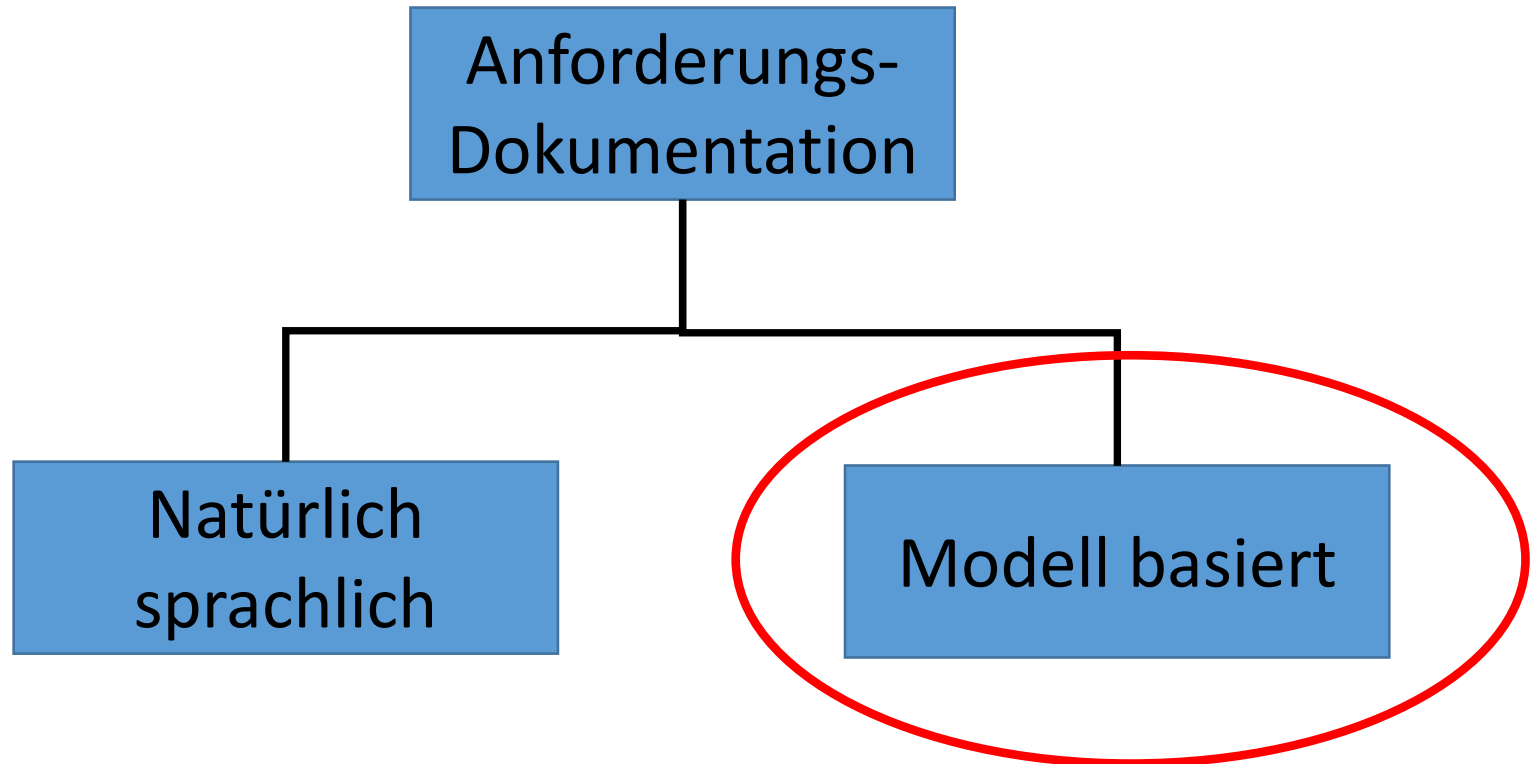
Begründung für die Cohnsche Form

Siehe

<https://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template>

1. Verwendung der Ich Form führt zu Identifikation mit der User Story.
2. Einheitliche Struktur hilft dem Product Owner bei der Priorisierung.
3. Sinnvolle Struktur (Bitte überprüfen Sie das Argument aus der Quelle oben selbst)

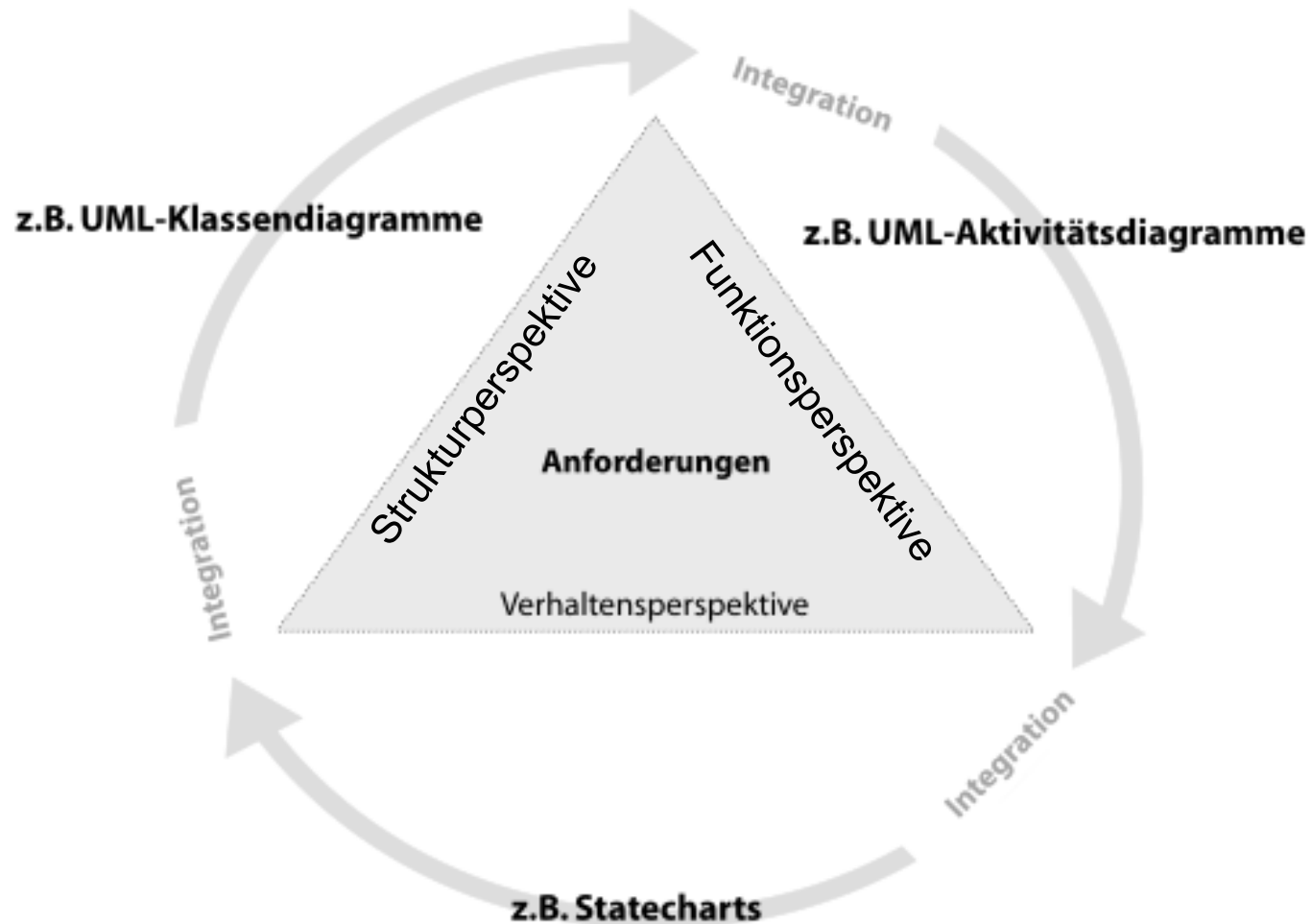
Anforderungen dokumentieren



Vorteile von Anforderungsmodellen

- Bessere Verständlichkeit (bildhaft dargestellte Information wird schneller erfasst)
- Unterstützung von Dokumentationsperspektiven (definierter Fokus durch die verwendeten Modelle)
- Zweckabhängige Abstraktion

Arten der Dokumentation - Perspektiven



Drei Ausprägungen von Anforderungen werden ergänzend zueinander dokumentiert:

- **Ziele**
beschreiben Intentionen von Stakeholdern oder Stakeholdergruppen.
- **Usecases und Szenarien**
dokumentieren beispielhafte Abläufe der Systemnutzung.
- **Anforderungen**
beschreiben detaillierte Funktionalitäten und Qualitäten, die das zu entwickelnde System umsetzen soll und die möglichst vollständig und präzise als Eingabe für die weiteren Entwicklungsschritte dienen.

Drei Ausprägungen von Anforderungen werden ergänzend zueinander dokumentiert:

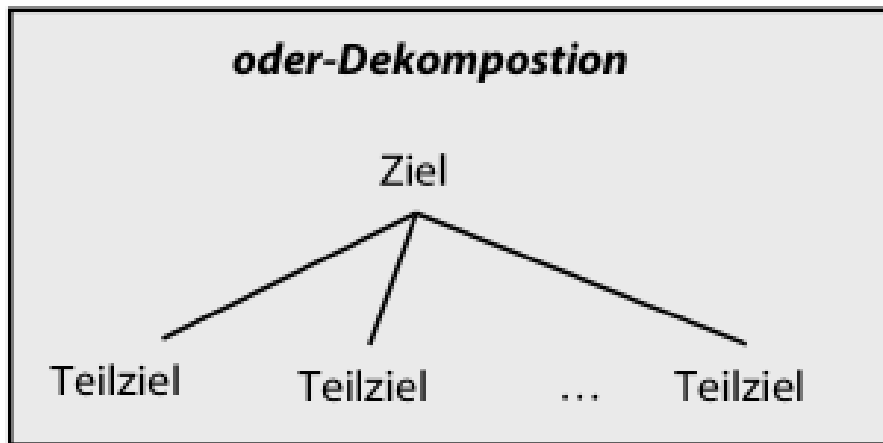
- **Ziele**
- **Usecases und Szenarien**
- **Anforderungen**

Zielmodelle

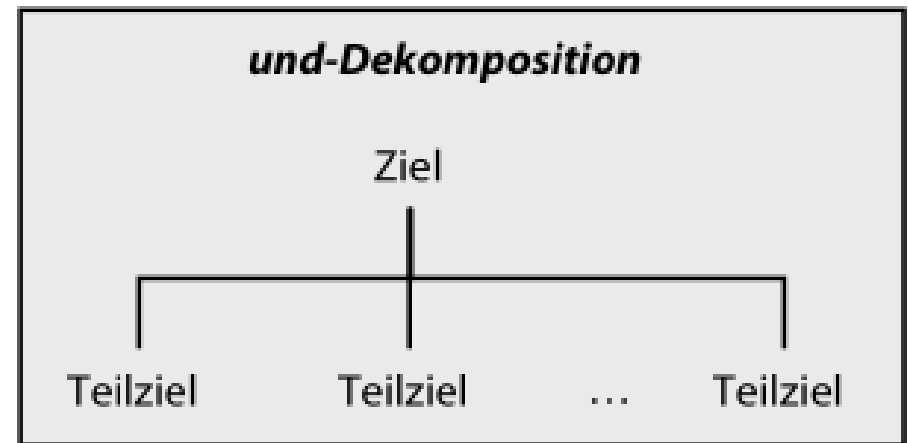
- Geeignet zur Verfeinerung der Vision der Stakeholder.
- Textuell oder z.B. mit und/oder Bäumen

Zieldokumentation mit und-oder Bäumen

Notation:



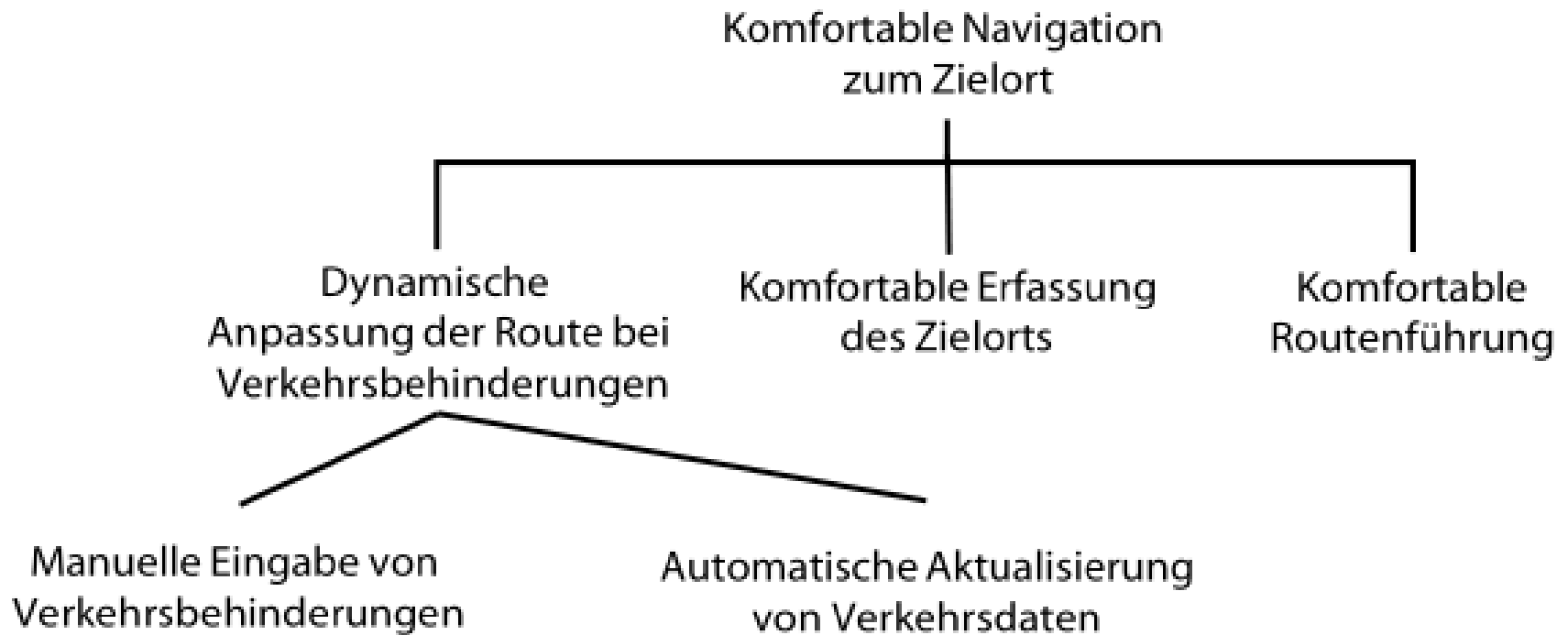
Mindestens ein Teilziel muss erfüllt sein, damit das Ziel erfüllt ist.



Jedes Teilziel muss erfüllt sein, damit das Ziel erfüllt ist.

Und-Oder Bäume - Beispiel

Ziel: Komfortable Navigation zum Zielort



Drei Ausprägungen von Anforderungen werden ergänzend zueinander dokumentiert:

- Ziele
- Usecases und Szenarien
- Anforderungen

Usecases

- Use Case Diagramme
- Use Case Spezifikationen

➔ Bekannt aus SE

Drei Ausprägungen von Anforderungen werden ergänzend zueinander dokumentiert:

- **Ziele**
- **Usecases und Szenarien**
- **Anforderungen**

Anforderungsmodellierung aus den drei verschiedenen Perspektiven mithilfe von zum Beispiel:

- Klassendiagrammen
- Activity Diagrammen
- Zustandsdiagrammen

- Requirements Engineering – was ist das
- Bedeutung des Requirements Engineering
- Anforderungen
- Aufgaben des RE
 - Systemkontext ermitteln
 - Anforderungen ermitteln
 - Anforderungen dokumentieren
 - Anforderungen prüfen und abstimmen
 - Anforderungen verwalten
- Toolunterstützung

- **Ziel:**
 - Erkennen von Fehlern in den Anforderungen
 - Gemeinsames Verständnis der Anforderungen

- **Qualitätsaspekte der Überprüfungen:**
 - Inhalt
 - Dokumentation
 - Abgestimmtheit

Techniken zur Überprüfung von Anforderungen

- ➔ bereits behandelt:
Siehe statische Analyse

Prinzipien der Prüfung von Anforderungen

- Prinzip 1: Beteiligung der richtigen Stakeholder
- Prinzip 2: Trennung von Fehlersuche und Fehlerkorrektur
- Prinzip 3: Prüfung aus unterschiedlichen Sichten
- Prinzip 4: Geeigneter Wechsel der Dokumentationsform
- Prinzip 5: Konstruktion von Entwicklungsartefakten
- Prinzip 6: Wiederholte Prüfung

Quelle: Klaus Pohl, Chris Rupp: Basiswissen Requirements Engineering

➔ Konflikte identifizieren und auflösen

Einzelne Schritte

- Konfliktidentifikation
- Konfliktanalyse
- Konfliktauflösung
- Dokumentation der Konfliktauflösung

- Requirements Engineering – was ist das
- Bedeutung des Requirements Engineering
- Anforderungen
- Aufgaben des RE
 - Systemkontext ermitteln
 - Anforderungen ermitteln
 - Anforderungen dokumentieren
 - Anforderungen prüfen und abstimmen
 - Anforderungen verwalten
- Toolunterstützung

- Attributierung
- Sichten auf Anforderungen
- Priorisierung
- Verfolgbarkeit von Anforderungen
- Versionierung von Anforderungen
- Änderungen

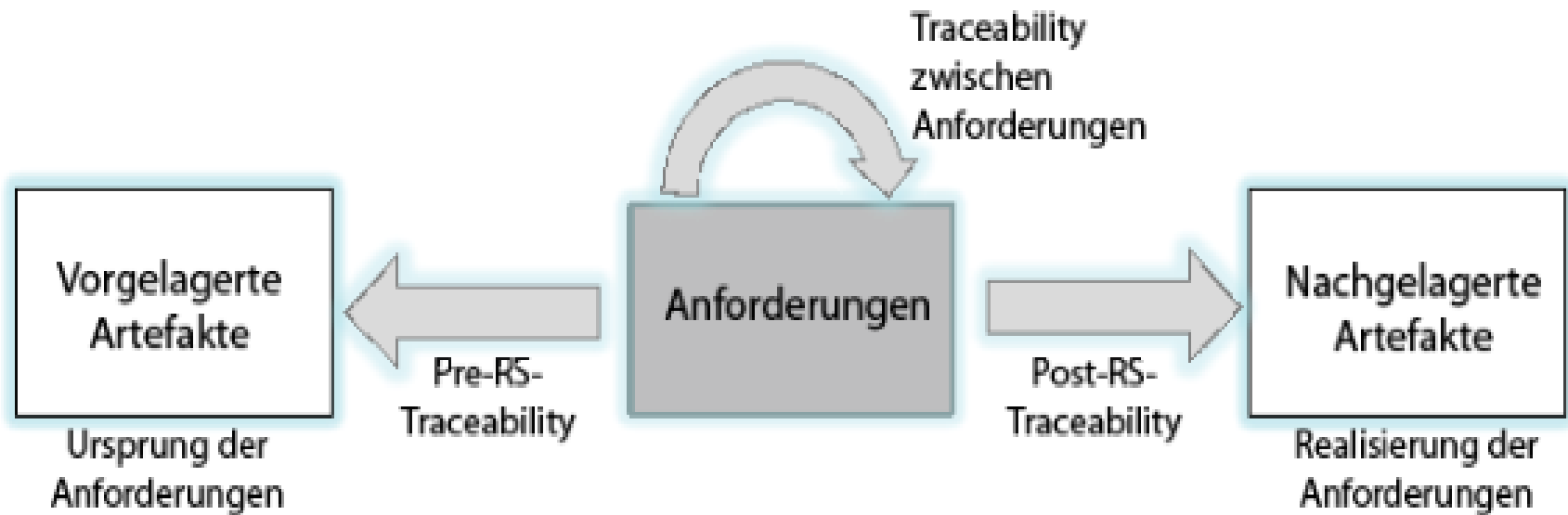
Auf den folgenden Folien besprochen: Attributierung, Priorisierung, Verfolgbarkeit, Änderungsverwaltung

Häufig verwendete Attributtypen

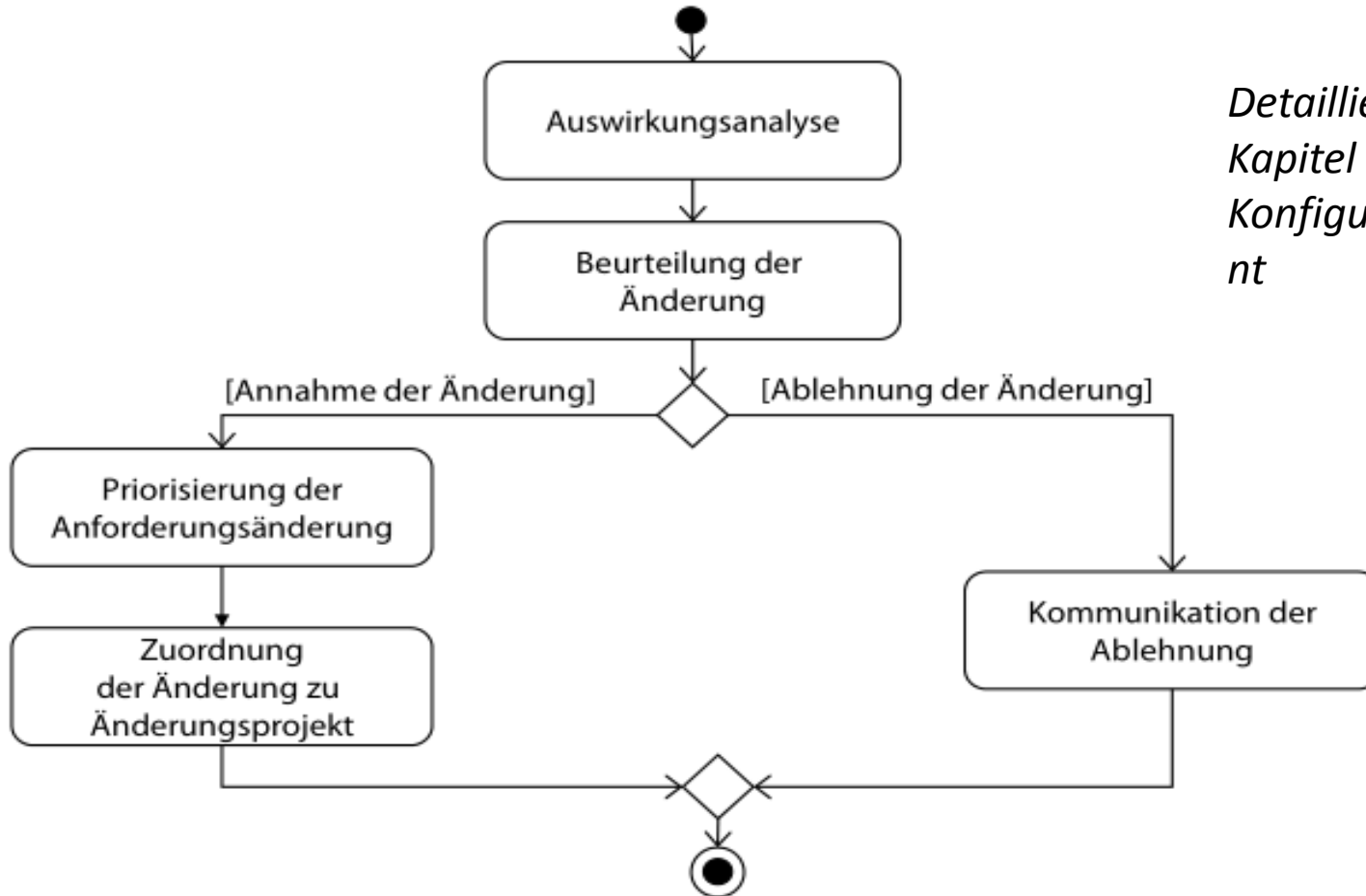
Attributtyp	Bedeutung
Identifikator	Kurze, eindeutige Identifikation eines Anforderungsartefakts in der Menge der betrachteten Anforderungen.
Name	Eindeutiger, charakterisierender Name.
Beschreibung	Beschreibt in komprimierter Form den Inhalt der Anforderung.
Version	Aktueller Versionsstand der Anforderung.
Autor	Benennt den/die Autor(in) der Anforderung.
Quelle	Benennt die Quelle bzw. Quellen der Anforderung.
Begründung	Beschreibt, weshalb diese Anforderung für das geplante System von Bedeutung ist.
Stabilität	Benennt die voraussichtliche Stabilität der Anforderung. Stabilität ist dabei der Umfang, in dem künftig noch Veränderungen bzgl. dieser Anforderung erwartet werden. Mögliche Unterscheidung: »fest«, »gefestigt«, »volatil«.
Kritikalität	Im Sinne einer Abschätzung der Schadenshöhe und Eintrittswahrscheinlichkeit.
Priorität	Benennt die Priorität der Anforderung hinsichtlich der gewählten Merkmale zur Priorisierung, z.B. »Bedeutung für die Akzeptanz am Markt«, »Reihenfolge der Umsetzung«, »Schaden bzw. Opportunitätskosten durch Nichtrealisierung«.

*Quelle: Klaus Pohl,
Chris Rupp:
Basiswissen
Requirements
Engineering*

Arten der Verfolgbarkeit



Änderungen

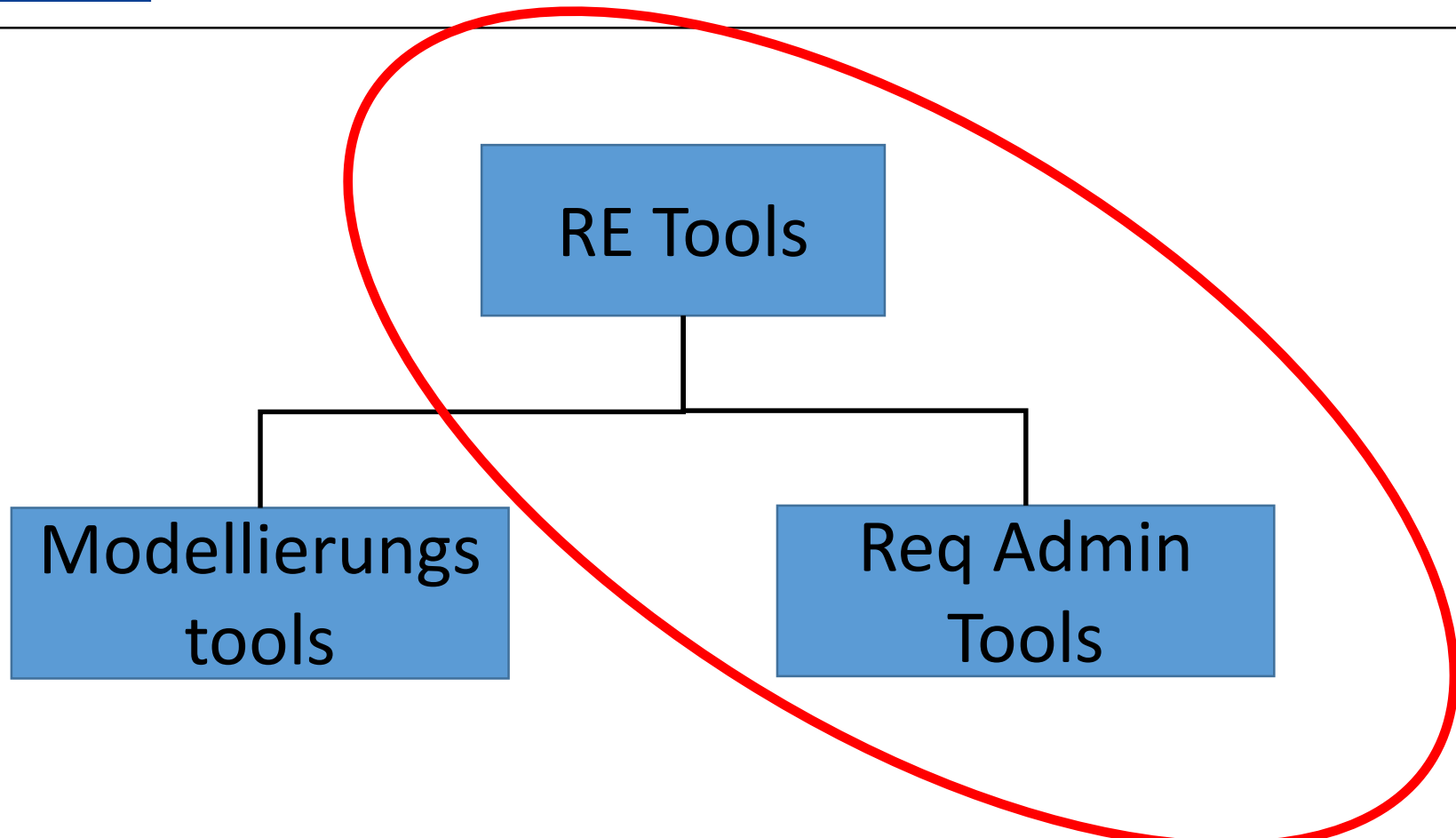


*Detaillierter siehe
Kapitel
Konfigurationsmanageme
nt*

Quelle: Klaus Pohl, Chris Rupp: Basiswissen Requirements Engineering

- Requirements Engineering – was ist das
- Bedeutung des Requirements Engineering
- Anforderungen
- Aufgaben des RE
 - Systemkontext ermitteln
 - Anforderungen ermitteln
 - Anforderungen dokumentieren
 - Anforderungen prüfen und abstimmen
 - Anforderungen verwalten
- Toolunterstützung

Tools für das Requirements Engineering



Normalerweise Synonym verwendet

- Verwalten verschiedener Informationen (z.B. natürlichsprachliche Anforderungen, konzeptuelle Modelle, Skizzen, Testpläne, Änderungswünsche)
- Verwalten von logischen Beziehungen zwischen verschiedenen Informationen (Verfolgbarkeit, z.B. zwischen Anforderungen, zwischen Anforderungen und deren Umsetzung)
- Eindeutige Identifizierbarkeit (z.B. jedes verwaltete Artefakt sollte über eine eindeutige ID verfügen)

- Bearbeiten der verwalteten Informationen (Mehrbenutzerfähigkeit, Zugriffskontrolle, Konfigurations- und Versionsmanagement)
- Bilden von unterschiedlichen Sichten auf die verwalteten Informationen je nach Einsatzzweck
- Organisieren der verwalteten Informationen (Gruppierung, Hierarchiebildung, Attributierung und Annotation zusätzlicher Informationen)
- Erstellen von Reports oder Auswertungen über die verwalteten Informationen (z.B. Reports über Änderungsanträge für Anforderungen)

Tools

- Liste von Tools unter

<http://www.volere.co.uk/tools.htm>