

# Requirements Management und Modellierung - VO

Einheit 1 – WS 2013/2014

Dipl.-Ing. Mag. Dr. Michael Tesar

[michael.tesar@fhwn.ac.at](mailto:michael.tesar@fhwn.ac.at)

# Terminplan und LV-Inhalte

	Datum	Uhrzeit	Einheiten	Inhalt
<b>VO – 01</b>	Mo., 14.10.2013	17:30 – 21:00	4	Einführung in die Lehrveranstaltung. Einführung in das Requirements Engineering Ziele und Modelle, Arten von Anforderungen
<b>VO – 02</b>	Mo., 21.10.2013	17:30 – 21:00	4	Anforderungen ermitteln Rollen, Faktoren, Techniken Anforderungen formulieren Vorbereitungen zur guten Dokumentation
<b>VO – 03</b>	Mo., 28.10.2013	17:30 – 21:00	4	Anforderungen validieren Prüftechniken Qualitätsmetriken
<b>VO – 04</b>	Mi., 13.11.2013	17:30 – 21:00	4	Versionsmanagement Change- und Releasemanagement Wiederverwendung von Anforderungen Arbeiten in verteilten Projektteams
<b>PR</b>	<b>Mi., 27.11.2013</b>	<b>17:30 – 18:30</b>	<b>1</b>	Schriftliche Prüfung über die Inhalte der Vorlesung
<b>UE – 01</b>	Mi., 27.11.2013	19:00 – 21:00	2	1. Übungseinheit

# Benotung Vorlesung

- > Schriftliche Prüfung
- > Ohne Unterlagen
- > 60 Minuten
- > Lehrveranstaltungs-Stoff

## **Benotung:**

ab 60% = Genügend

ab 70% = Befriedigend

ab 80% = Gut

ab 90% = Sehr gut

# Anwesenheitspflicht

- > Es besteht keine Anwesenheitspflicht in der Vorlesung.
- > Es steht den Studierenden frei, bei den Lehrveranstaltungsterminen anwesend zu sein.

# Literatur Vorlesung

- > Folien & Mitschrift
- > Buch:  
Chris Rupp; **Requirements-Engineering und -Management: Professionelle, Iterative Anforderungsanalyse für die Praxis**; Hanser Verlag; 2009; 5. Auflage
- > Moodle-Kurs mit weiterführenden Materialien



# Literatur Übung

- > <http://www.omg.org/spec/UML/2.4.1/>
- > Folien & Mitschrift
- > Moodle-Kurs mit weiteren Materialien
- > Buch (nicht verpflichtend):  
**UML 2 glasklar**  
Chris Rupp & die Sohnpisten  
Hanser



# Einführung in die Thematik

Was ist  
Systemmodellierung?

Was ist Requirements-  
Engineering?

Wozu braucht  
man das?

# Der Einstieg in die Systemmodellierung

*Man muss die Dinge so tief sehen,  
dass sie einfach werden*

*- Konrad Adenauer*



# Begriffsdefinition „System“

Definiert nach Weilkiens bzw. INCOSE

*Ein **System** ist ein von Menschen erstelltes Artefakt bestehend aus Systembausteinen, die gemeinsam ein Ziel verfolgen, das von den Einzelelementen nicht erreicht werden kann. Ein Baustein kann aus Software, Hardware, Personen oder beliebigen anderen Einheiten bestehen.*

# Begriffsdefinition „Systems Engineering“

Definiert nach Weilkiens bzw. INCOSE

*Das **Systems Engineering** konzentriert sich auf die Definition und Dokumentation der Systemanforderungen in der frühen Entwicklungsphase, die Erarbeitung eines Systemdesigns und die Überprüfung des Systems auf Einhaltung der gestellten Anforderungen unter Berücksichtigung des Gesamtproblems: Betrieb, Zeit, Test, Erstellung, Kosten & Planung, Training & Support und Entsorgung*

# Systems vs. Software Engineering

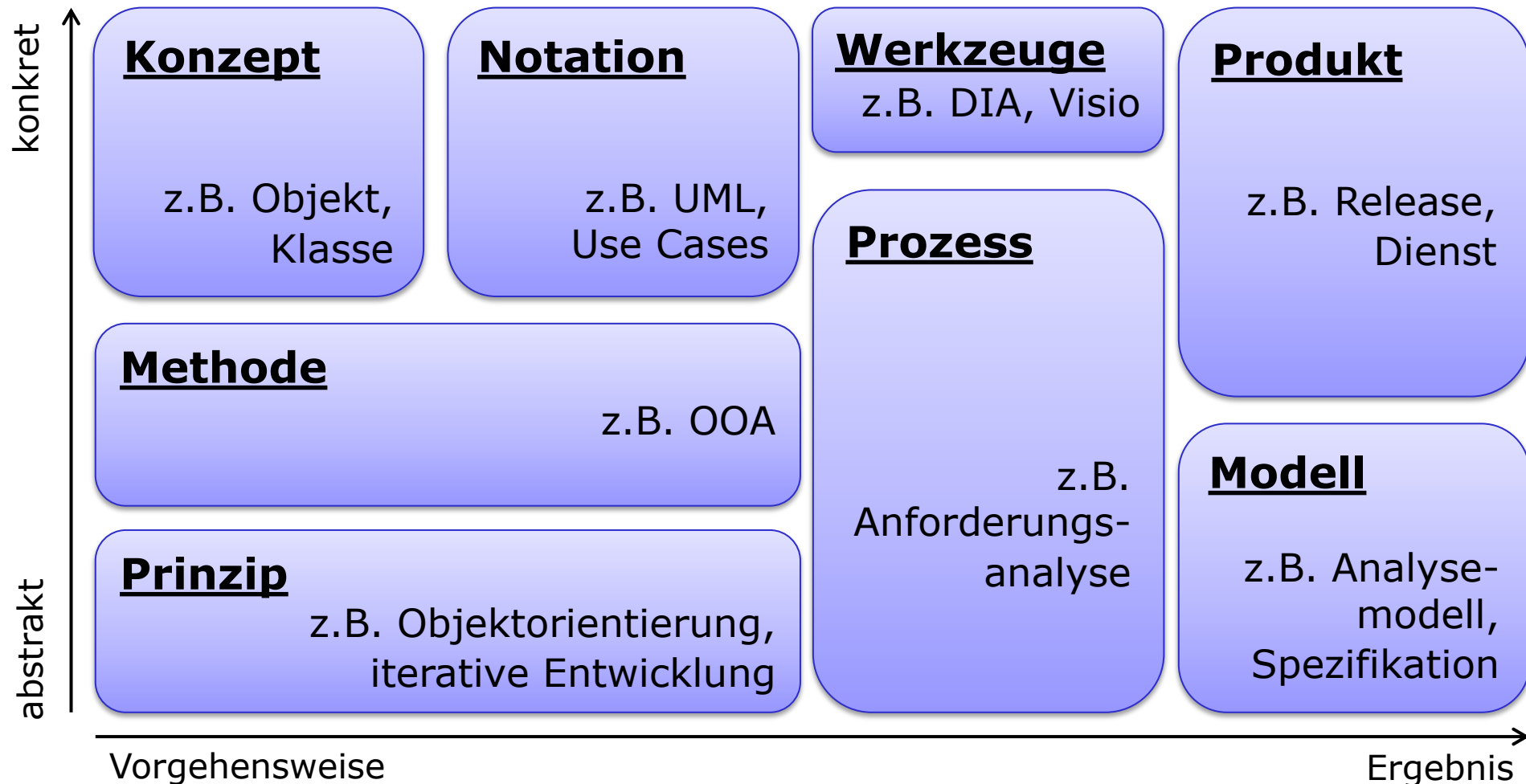
Nach Weilkiens

Software Engineering, Hardware Engineering, Verfahrenstechnik usw. sind Disziplinen, die bestimmte Bausteine des Systems entwickeln. Das Systems Engineering betrachtet das System ganzheitlich! (Inkl. Entwicklung, Betrieb und Entsorgung)

## Produkt:


...ist ein Wirtschaftsgut, das in einem Wertschöpfungsprozess geschaffen wird, in dem Produktionsfaktoren umgewandelt werden. vereinfachend darzustellen.

definierten Reihenfolge zu erreichen.



# Komplexität vs. Kompliziertheit

Nach Klaus, 1969



Viele Systeme sind **komplex** und **kompliziert**! Das ist die Herausforderung des Systems Engineering!

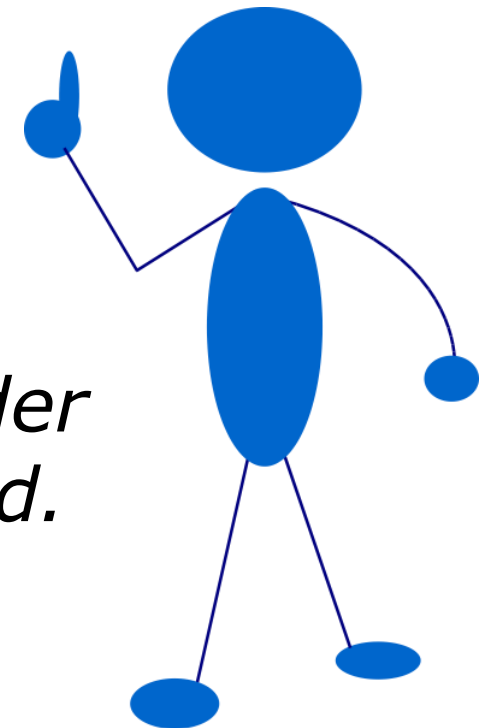
Die **Komplexität** bezieht sich auf die Anzahl und Art der Beziehungen zwischen Elementen in einem System.

Die **Kompliziertheit** bezieht sich auf die Anzahl der unterschiedlichen Elemente.

# Begriffsdefinition „Anforderung“ - 1

Definiert nach der IEEE  
(übersetzt von Christof Ebert)

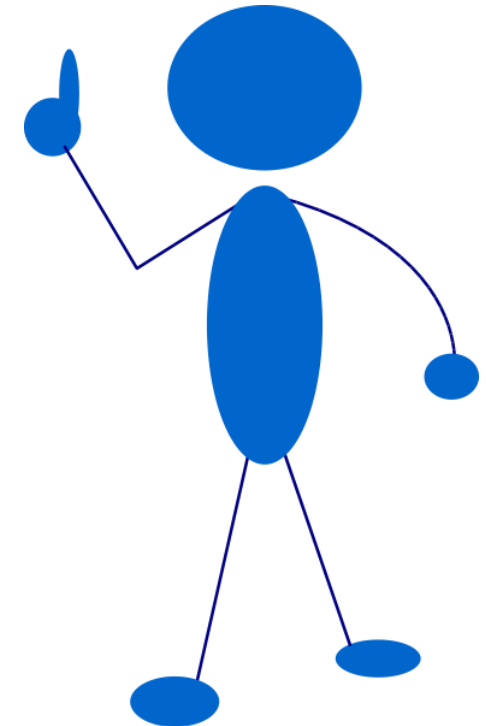
*Eine Eigenschaft oder Bedingung, die von einem Benutzer (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.*



# Begriffsdefinition „Anforderung“ - 2

Definiert nach der IEEE  
(übersetzt von Christof Ebert)

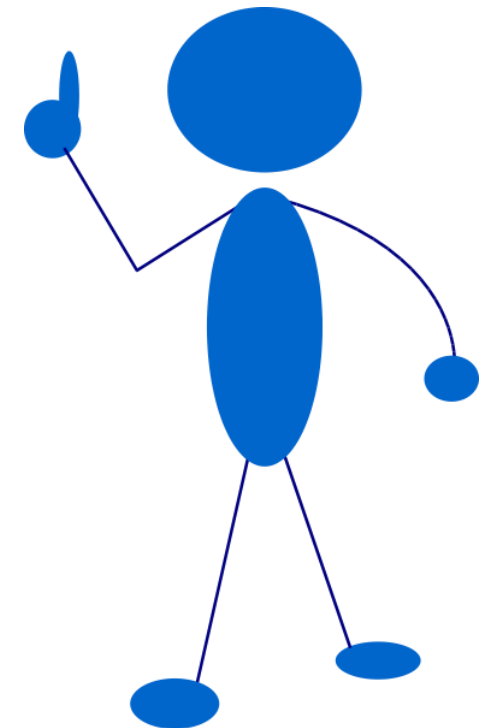
*Eine Eigenschaft oder Bedingung, die ein System oder eine Systemkomponente erfüllen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.*



# Begriffsdefinition „Anforderung“ - 3

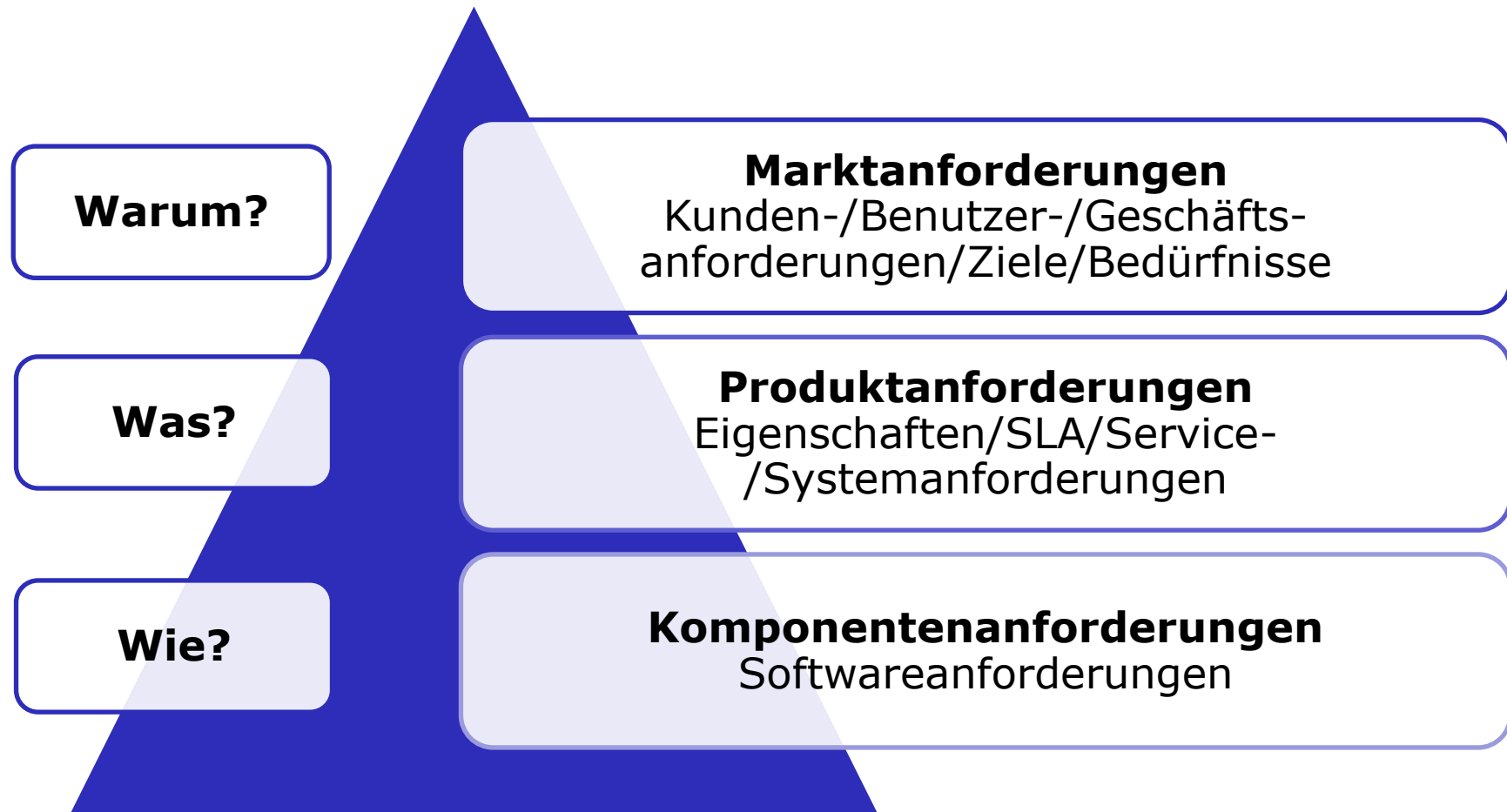
Definiert nach der IEEE  
(übersetzt von Christof Ebert)

*Eine dokumentierte Repräsentation einer Eigenschaft oder Bedingung wie in den ersten beiden Punkten beschrieben.*





# Sichten auf Anforderungen



# Systemmodellierung

Vorgang zur Erstellung einer abstrakten Repräsentation eines beliebigen Systems, im speziellen informationstechnischer Systeme.

# Requirements Engineering

ist ein **kooperativer, iterativer, inkrementeller** Prozess, dessen Ziel es ist zu gewährleisten, dass...

- > Anforderungen im erforderlichen Detaillierungsgrad sind bekannt
- > Stakeholder eine Übereinstimmung über die Anforderungen erzielen
- > Anforderungen dokumentiert und spezifiziert sind

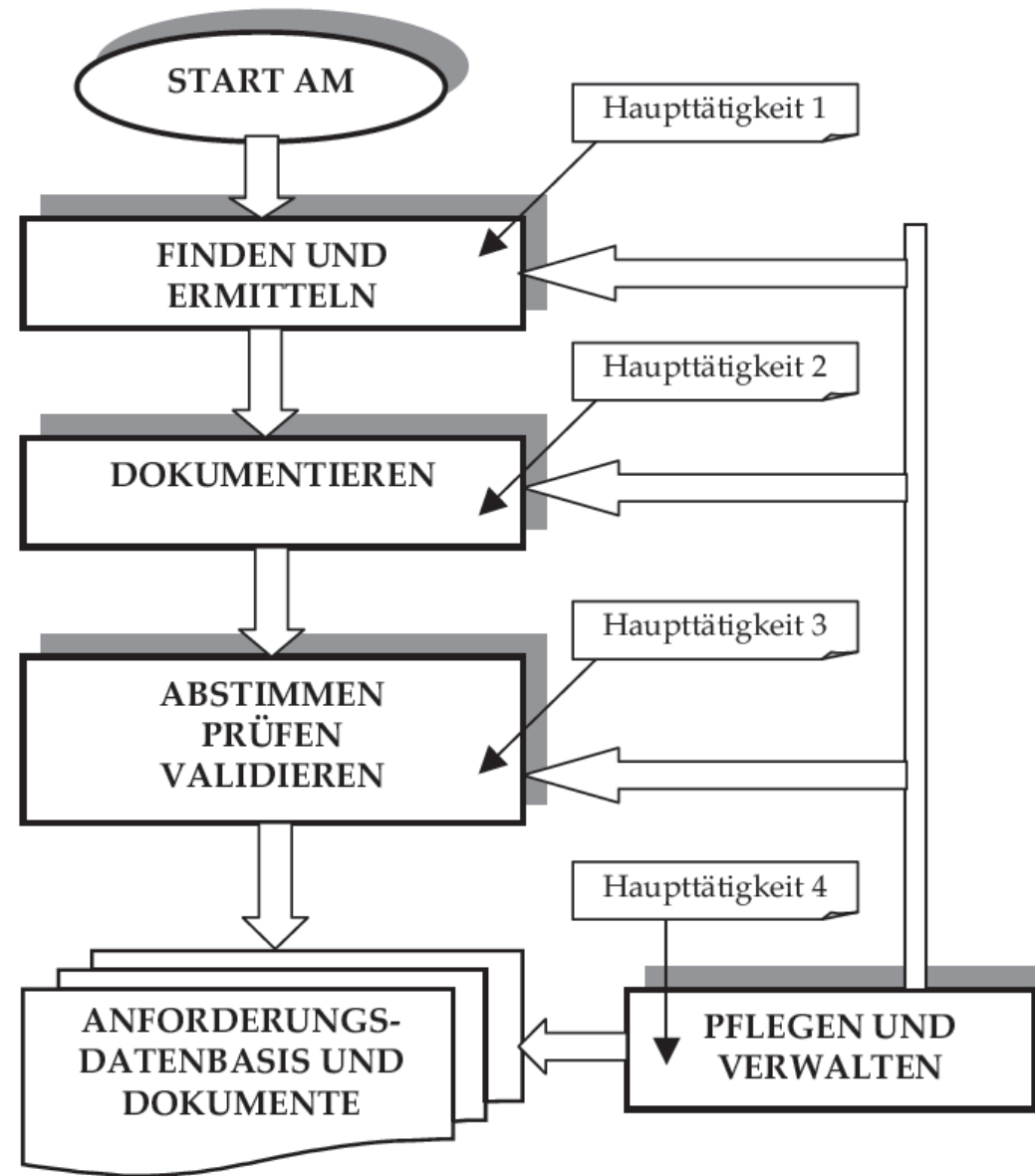
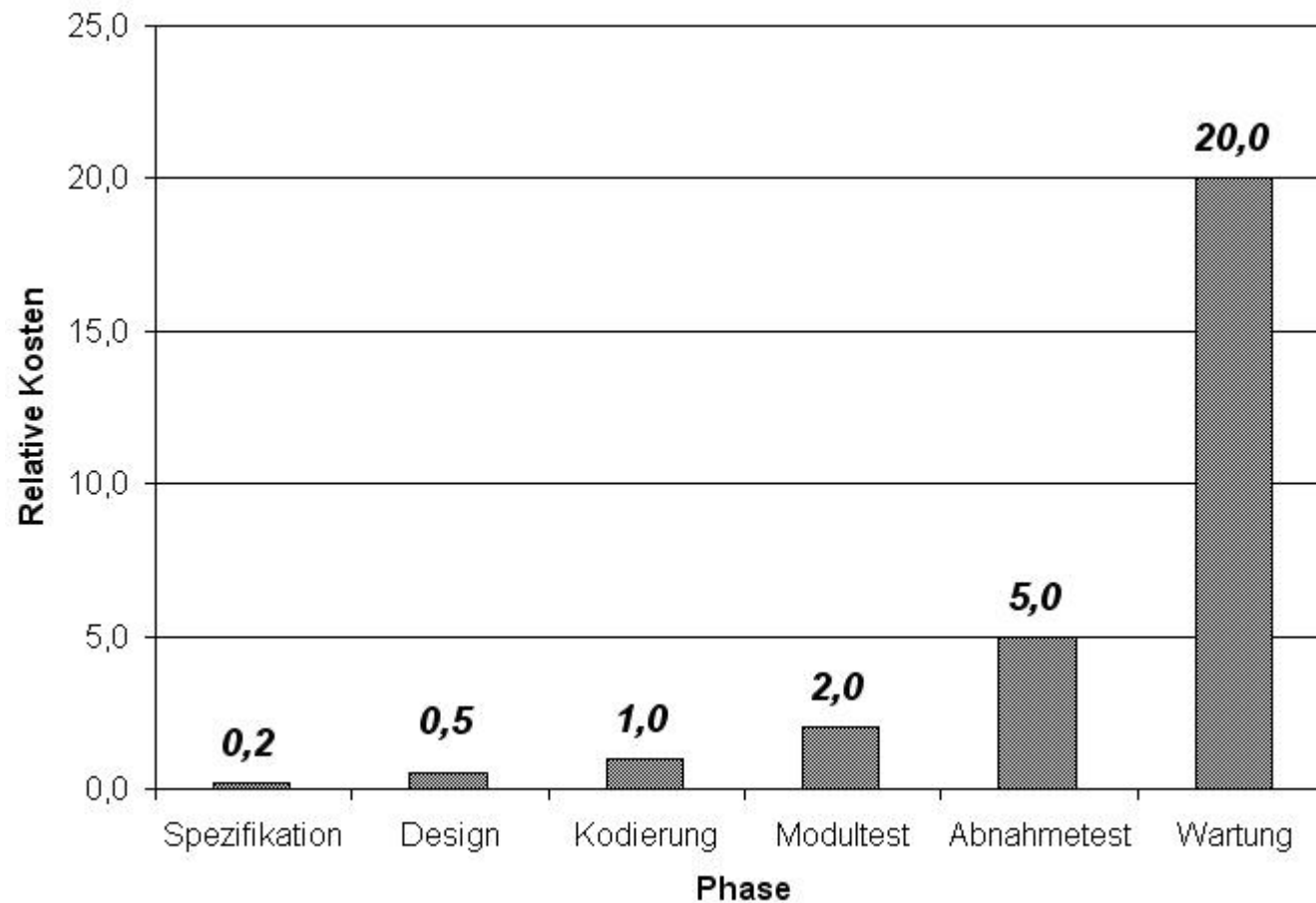


Abbildung 2-2: Haupttätigkeiten im Anforderungsmanagement

# Wozu braucht man das?

- > Was soll ich entwickeln?
- > Absicherung zwischen Kunden und AuftragnehmerIn
- > Qualitätssteigerung
- > Erfolgsquote steigern
- > Vermeidung von Fehlern in der Entwicklung

# Wozu braucht man das? II



**Abb: Relative Kosten für die Beseitigung eines Softwarefehlers**

# Die Illusion der Kommunikation



"A whole year to build  
a house here?  
No problem."

"Good. Let's get  
started. I'm in  
a hurry."

Weiß der Kunde wirklich was er braucht? Oder **glaubt** er nur es zu wissen?

Haben wir **verstanden** was der Kunde braucht?

Haben wir alles gut genug **spezifiziert**? Und was bedeutet „alles“:  
Prozesse, Systemgrenzen, Technologie,...?

Reden wir über die gleiche **Vision**?

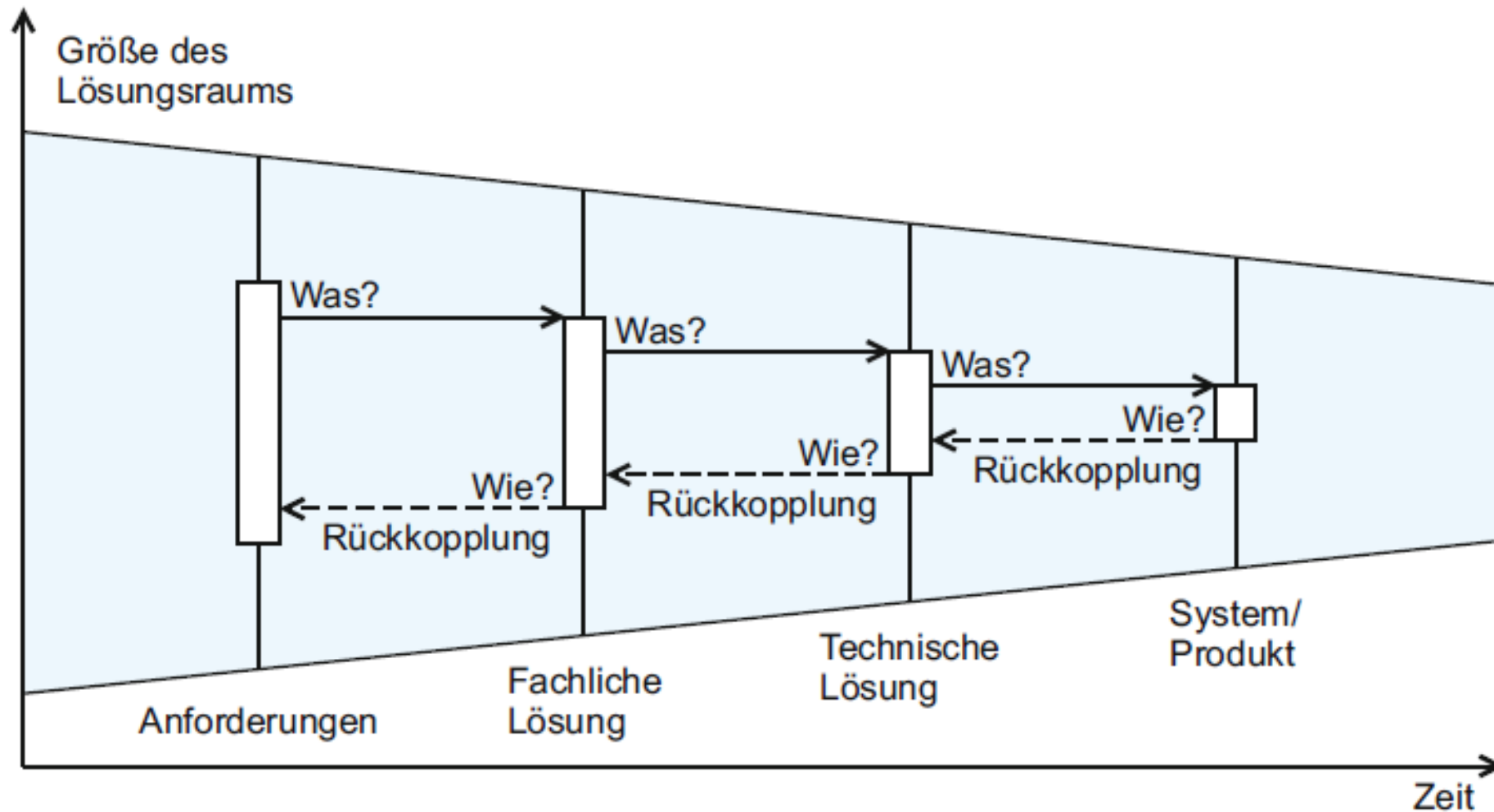
Reden wir über die gleiche **Mission**?

# Eigenschaften einer/s RE's

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



# Entwicklung des Lösungsraums



# Vorgehensmodelle

Wasserfall – brauch' ich  
mehr?

V-Modell - der Weisheit  
letzter Schluss?

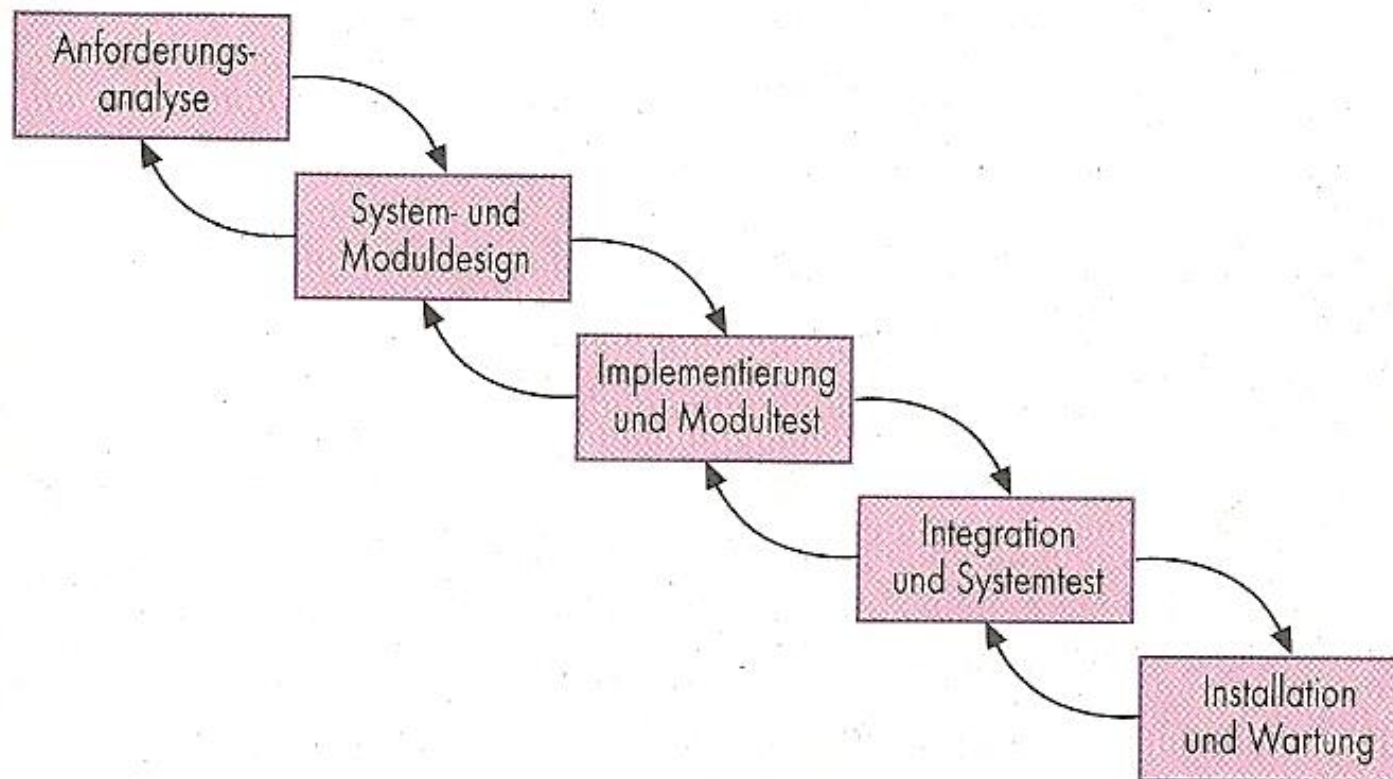
Rational Unified Process –  
optimal für IT-Projekte?

Agile Methoden – sinnvoll  
oder nur „in“?

# Wozu ein Vorgehensmodell?

- > Strukturierung des Projekts
- > Zeitlich und methodisch
- > Vorlage => Sicherheit
- > Positionsbestimmung
- > Transparenz
- > Basis für Analysen, Verbesserungen und Planungen

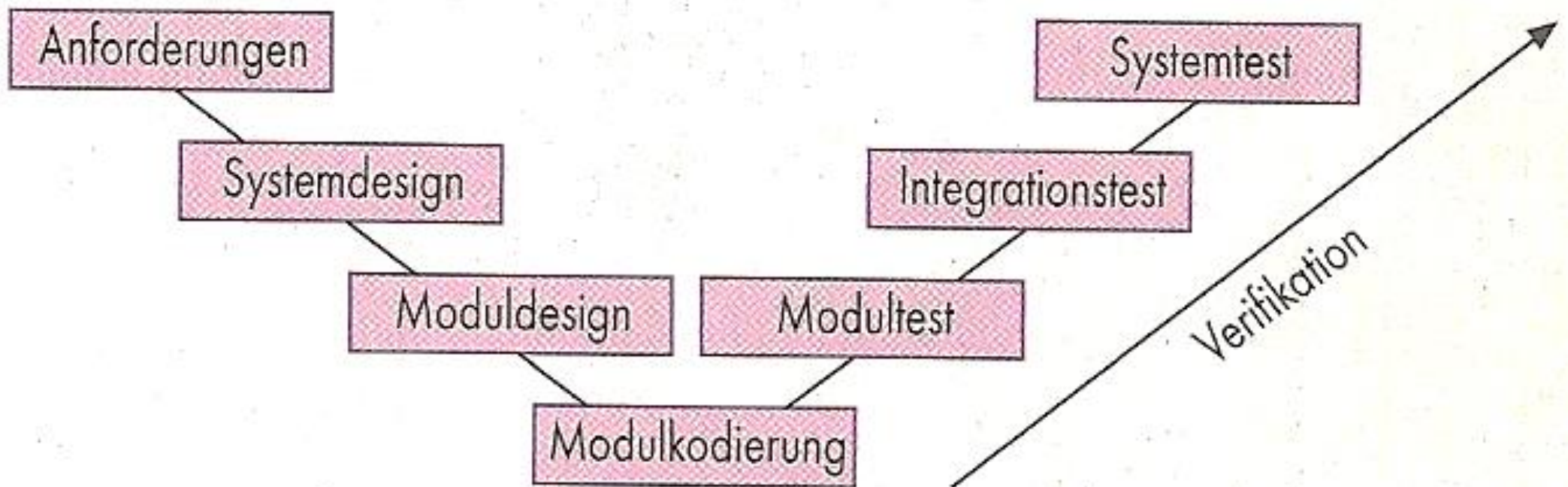
# Wasserfallmodell



# Wasserfallmodell

- > + Der Klassiker
- > + Kontrollierbarer und leicht verständlicher Prozessablauf
- > - Risiken nicht frühzeitig erkennbar!
- > - Feedback-Möglichkeiten fehlen

# V-Modell

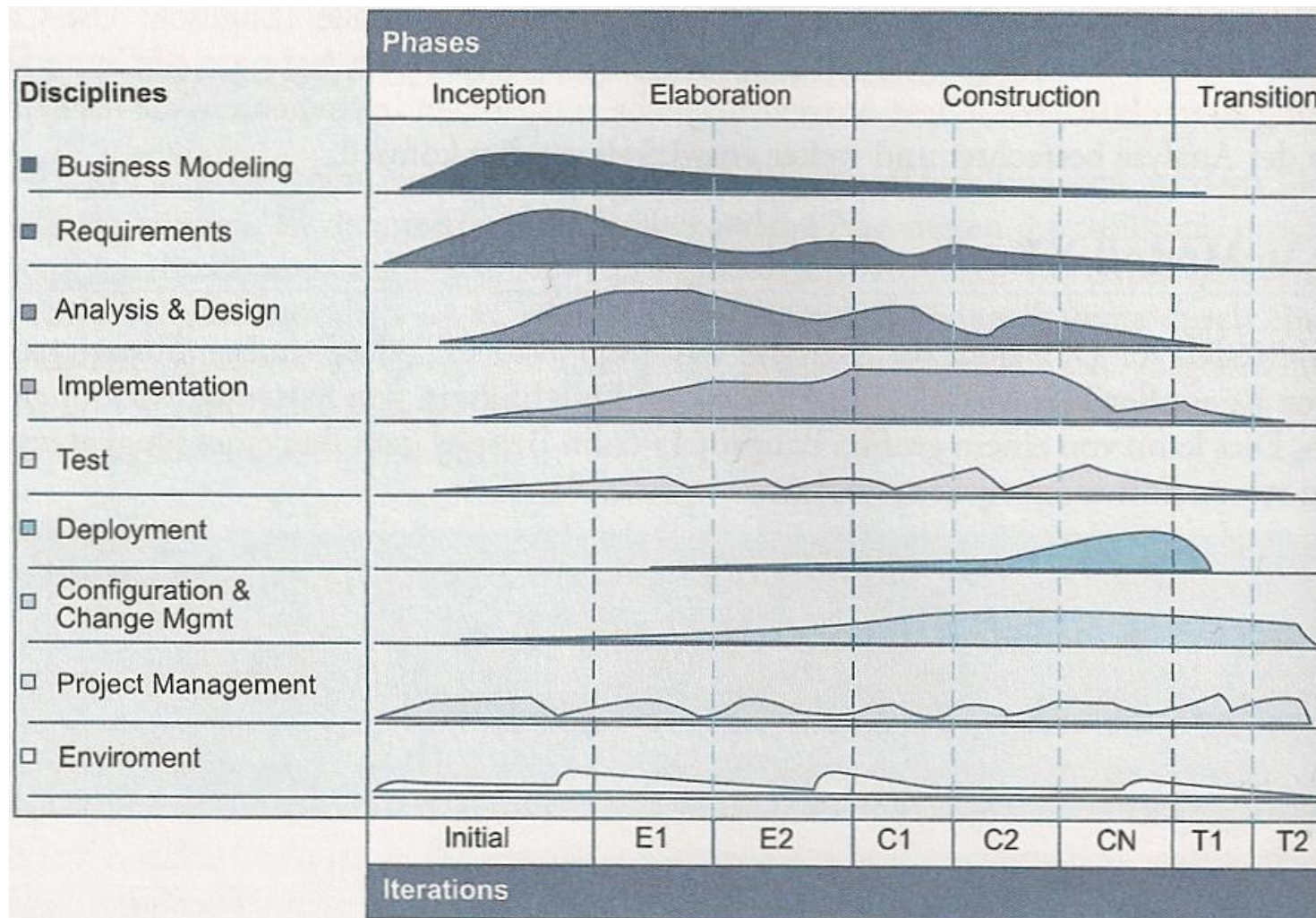


# V-Modell

- > + Weiterentwicklung des Wasserfallmodells
- > + Qualitätssicherung hinzugefügt
- > + Verifikation (Arbeitsprodukt entspricht Anforderungen)
- > + Validierung (Arbeitsprodukt entspricht beabsichtigter Nutzung)
- > - viele Derivate
- > - oft komplexe Vorgehensweisen



# Rational Unified Process



Quelle: Rupp, 2009, S. 38



# Rational Unified Process

- > + Von Rational (aufgekauft von IBM) entwickelt
- > + Für UML entwickelt
- > + Alle Phasen bestehen aus mehreren Iterationen
- > + Hohe Flexibilität
- > - Hoher Managementaufwand
- > - Hohe Komplexität

# RUP: 9 Kern-Workflows

1. Business Modeling
2. Requirements
3. Analysis & Design
4. Implementation
5. Test
6. Deployment
7. Configuration & Change Management
8. Project Management
9. Environment

# RUP: 6 „Best Practices“

- > Anforderungsmanagement
- > Iterative Entwicklung
- > Architekturorientierung
- > Visuelle Modellierung
- > Qualitätskontrolle
- > Change- und Konfigurationsmanagement

# Agile Methoden / SCRUM

- > Ausgangslage: Entwicklungsprozesse von IT zu komplex => nicht abbildbar
- > Großen äußeren Rahmen vorgeben
- > Selbstorganisation des Teams
- > => Soll produktiver sein
- > => Gemeinsame Verantwortung zur Zielerreichung
- > Hoch konzentrierte Aktivitäten an Teilaspekten
- > Rasche, funktionierende Ergebnisse

# Ziele und mehr...

Ziele im Requirements  
Engineering

Anforderungsarten

Anforderungs-  
quellen

# Begriffsdefinition „Ziel“

## Definition nach Pohl

*Ein Ziel ist die intentionale Beschreibung eines charakteristischen Merkmals des zu entwickelnden Systems bzw. des zugehörigen Entwicklungsprozesses.*

# Vorteile vom Ziel-Einsatz

- > Systemverständnis und Akzeptanz des Systems
- > Gewinnung von Anforderungen
- > Identifikation und Bewertung von Lösungsalternativen
- > Aufdeckung irrelevanter Anforderungen
- > Ziele begründen Anforderungen
- > Nachweis der Vollständigkeit
- > Identifikation und Auflösung von Konflikten
- > Stabilität von Zielen

# Anforderungsartefakte nach Pohl

## > Ziele

- Dokumentation der Intentionen der Stakeholder
- Abstrahierung

## > Szenarien

- Exemplarische konkrete Beispiele für Interaktionsfolgen
- Unterschiedliche Abstraktionsebenen

## > Lösungsorientierte Anforderungen

- Definieren Daten-, Struktur-, Funktions- und Verhaltenssicht des Systems
- Qualitätsanforderungen



# Und-Dekomposition eines Ziels

„Bei einer Und-Dekomposition eines Ziels  $Z$  wird dieses Ziel durch eine Menge von Teilzielen  $Z_1$  bis  $Z_n$  verfeinert. Die Dekomposition eines Ziels  $Z$  in die Teilziele  $Z_1, \dots, Z_n$  mit  $n \geq 2$  ist genau dann eine Und-Dekomposition, wenn alle Teilziele  $Z_1, \dots, Z_n$  erfüllt werden müssen, um das Ziel  $Z$  zu erfüllen.“

Vgl. Pohl

# Oder-Komposition eines Ziels

„Die Dekomposition eines Ziels  $Z$  in die Teilziele  $Z_1, \dots, Z_n$  mit  $n \geq 2$  ist genau dann eine Oder-Dekomposition, wenn die Erfüllung eines der Teilziele  $Z_1, \dots, Z_n$  ausreicht, um das Ziel  $Z$  zu erfüllen. Im Unterschied zur Und-Dekomposition muss bei einer Oder-Dekomposition zur Erfüllung des Ziels  $Z$  daher nur mindestens ein Teilziel von  $Z$  erfüllt werden.“

Vgl. Pohl

# Beispiel Navigationssystem

„Komfortable und schnelle Navigation zum Ziel“

- > Und-Dekomposition
- >  $Z_1$ : Komfortable Eingabe des Zielorts
- >  $Z_2$ : Automatische Auswahl der Route gemäß nutzerspezifischen Parametern
- >  $Z_3$ : Anzeigen und Umfahren von Verkehrsbedingungen

Vgl. Pohl

# Beispiel Navigationssystem

„Lokalisierbarkeit des Fahrzeugstandorts“

- > Oder-Dekomposition
- >  $Z_1$ : Ortung des Fahrzeugs über Mobilfunk
- >  $Z_2$ : Lokalisierung des Fahrzeugs über GPS

Vgl. Pohl

# Weitere Aspekte

## > Zielunterstützung

*„Ein Ziel  $Z_1$  unterstützt ein Ziel  $Z_2$ , wenn die Erfüllung von  $Z_1$  zur Erfüllung von  $Z_2$  beiträgt.“*

## > Zielbehinderung

*„Ein Ziel  $Z_2$  behindert ein Ziel  $Z_1$ , wenn die Erfüllung von  $Z_2$  die Erfüllung von  $Z_1$  erschwert.“*

Vgl. Pohl

# Weitere Aspekte

## > Zielkonflikt

*„Zwischen einem Ziel  $Z_1$  und einem Ziel  $Z_2$  besteht ein Zielkonflikt, wenn*

*(1) die Erfüllung von  $Z_1$  die Erfüllung des Ziels  $Z_2$  ausschließt und*

*(2) die Erfüllung von  $Z_2$  die Erfüllung des Ziels  $Z_1$  ausschließt.“*

Vgl. Pohl

# Weitere Aspekte

## > Zieläquivalenz

*„Zwischen zwei Zielen  $Z_1$  und  $Z_2$  besteht eine Zieläquivalenz, wenn*

*(1) die Erfüllung des Ziels  $Z_1$  die Erfüllung des Ziels  $Z_2$  bedingt und*

*(2) die Erfüllung des Ziels  $Z_2$  die Erfüllung des Ziels  $Z_1$  bedingt.“*

Vgl. Pohl

# Qualitätskriterien für Ziele I

- > Vollständig
- > Korrekt
- > Konsistent gegenüber anderen Zielen und in sich
- > Testbar
- > Verständlich für alle
- > Umsetzbar – realisierbar
- > Notwendig

Vgl. Rupp



# Qualitätskriterien für Ziele II

- > Eindeutig und
- > Positiv formuliert
- > Gültig und
- > Aktuell sein

Vgl. Rupp

# Essenzielle Bestandteile von Zielen

- > Gewinnen Sie möglichst alle relevanten Ziele
- > Vermeiden Sie zu Beginn eine vollständige Erfassung aller Attribute für ein Ziel
- > Definieren Sie für jedes Ziel zunächst die Basisattribute (Bezeichner, Name, Quelle, Verantwortliche Person, etc.)
- > Definieren Sie die Beziehungen zu anderen Zielen

Vgl. Pohl

# Essenzielle Bestandteile von Zielen II

- > Analyse der Ziele auf Vollständigkeit und korrekte Beziehungen
- > Ergänzen Sie die fehlenden Teile
- > Füllen Sie die Zielschablone aus.

Vgl. Pohl

# Zielschablone nach Rupp

## **Ziel:**

Welches Ziel soll durch das neue System erreicht werden?

## **Stakeholder:**

Wer profitiert von der Erreichung dieses Ziels, wer fördert die Erreichung dieses Ziels?

## **Auswirkung auf den/die Stakeholder:**

Welche Arbeitsprozesse sind von der Zielerreichung betroffen?

## **Einschränkungen:**

Welche Faktoren können eine Lösungsauswahl einschränken?

## **Sonstiges:**

Gibt es weitere Anmerkungen, die für das Verständnis des Ziels wichtig sind?

# Zielschablone nach Pohl

Zielschablone		
Nr.	Abschnitt	Inhalt / Erläuterung
1	Bezeichner	Eindeutiger Bezeichner des Ziels
2	Name	Eindeutiger Name für das Ziel
3	Autoren	Namen der Autoren, die an dieser Zielbeschreibung mitgearbeitet haben
4	Version	Aktuelle Versionsnummer des Ziels
5	Änderungshistorie	Liste der verschiedenen Änderungsstände inklusive Datum der Änderung, Versionsnummer, Autor und ggf. Grund und Gegenstand der Änderung
6	Priorität	Angabe der Wichtigkeit des beschriebenen Ziels gemäß der verwendeten Priorisierungstechnik
7	Kritikalität	Kritikalität des Ziels z. B. für den Erfolg des Systems
8	Quelle	Bezeichnung der Quelle ([Stakeholder   Dokument   System]), von der das Ziel stammt
9	Verantwortlicher	Der für dieses Ziel verantwortliche Stakeholder
10	Nutzende Stakeholder	Stakeholder, die einen Nutzen aus der Erfüllung dieses Ziels ziehen
11	Zielebene	Bezeichner der Detaillierungsebene des Ziels
12	Zielbeschreibung	Detaillierte Beschreibung des Ziels
13	Übergeordnete Ziele	Angabe der übergeordneten Ziele zusammen mit dem jeweiligen Typ der Verfeinerung (Und/Oder)
14	Untergeordnete Ziele	Subziele dieses Ziels und die Art der Verfeinerung
15	Weitere Zielbeziehungen	Weitere Beziehungen zu anderen Zielen (z. B. Konflikt, Unterstützung)
16	Assoziierte Szenarien	Querbezüge zu Szenarien, die dem Ziel zugeordnet sind
17	Ergänzende Informationen	Weitere Informationen zum Ziel

# Wann welche Zielschablone?

- > Für Kleinprojekte: => Rupp
  - > Für Großprojekte: => Pohl
  - > Für einen ersten Entwurf: => Rupp
  - > Für Systemmodellierung: => Pohl
- 
- > Je feiner und besser Ziele ermittelt wurden, desto einfacher kann man sie nachher modellieren!

## 2 Arten von Anforderungen

- > Funktionale Anforderungen
- > Nicht-Funktionale Anforderungen

### **Korrekt: 3 Arten von Anforderungen!**

- > Funktionale Anforderung
- > Qualitätsanforderung
- > Randbedingung

## **Funktionale Anforderung:**

„Eine funktionale Anforderung ist eine Anforderung bezüglich des Ergebnisses eines Verhaltens, das von einer Funktion des Systems bereitgestellt werden soll.“



# Qualitätsanforderung

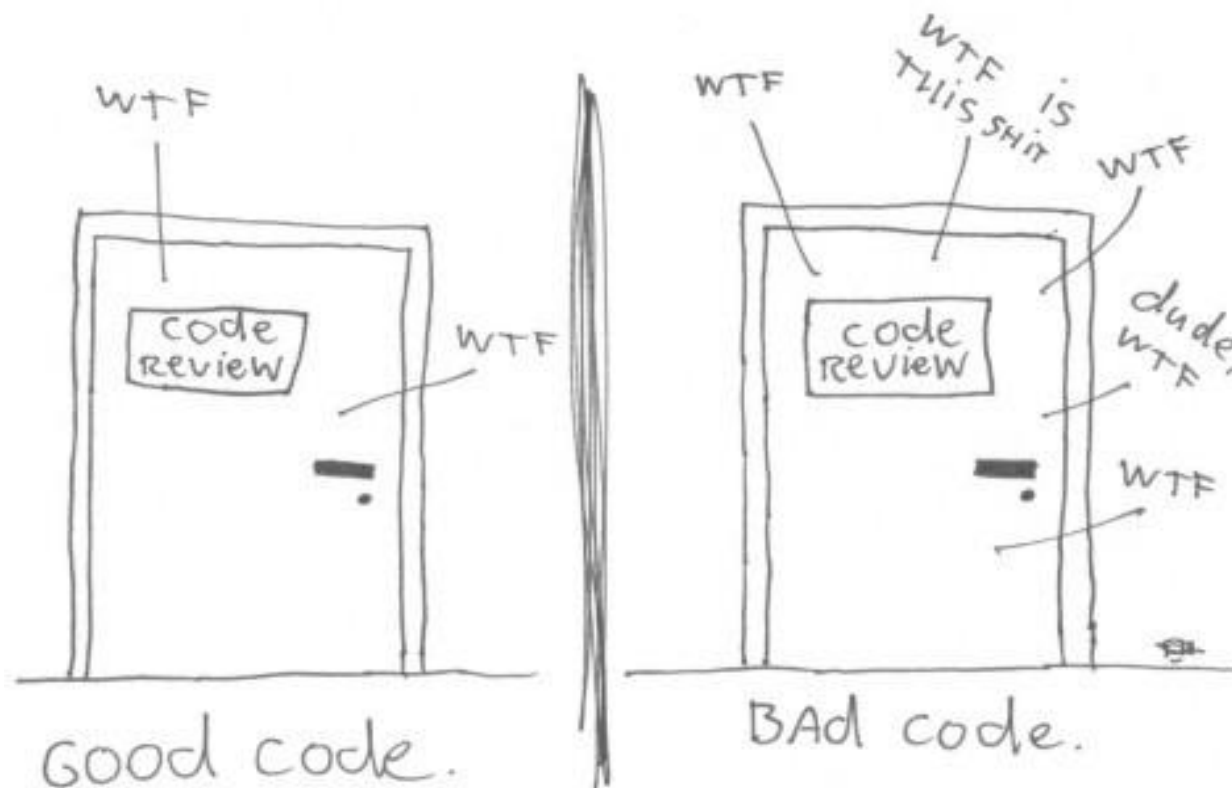
„Eine Qualitätsanforderung ist eine Anforderung, die sich auf ein Qualitätsmerkmal bezieht, das nicht durch funktionale Anforderungen abgedeckt wird.“

# Qualitätsanforderungen

- > Qualität der Funktionalität  
*(Angemessenheit, Sicherheit, Genauigkeit der Berechnung, Interoperabilität und Konformität)*
- > Zuverlässigkeit der Funktionalität
- > Benutzbarkeit des Systems
- > Effizienz des Systems
- > Änderbarkeit des Systems
- > Übertragbarkeit des Systems

# Qualität messen?

The ONLY VALID MEASUREMENT  
OF CODE QUALITY: WTFs/MINUTE



(c) 2008 Focus Shift/OSNews/Thom Holwerda - <http://www.osnews.com/comics>

# Probleme bei nicht-funktionalen Anforderungen

- > Zur Nachvollziehbarkeit, ob eine nichtfunktionale A. erfüllt ist, muss diese auch messbar sein.
  - Metrik definieren: z.B. max. page-hits / sec.
  - akzeptierte Limits in der SRS angeben
  - nichtfunktionalen A. ohne messbare Bewertung sind wertlos und ein potentieller Auslöser für Diskussionen, da subjektiv (z.B.: „Das System muss schnell sein“, „Die Website muss cool aussehen“).

## Randbedingung

„Eine Randbedingung ist eine Anforderung, die den Lösungsraum jenseits dessen einschränkt, was notwendig ist, um die funktionalen Anforderungen und die Qualitätsanforderungen zu erfüllen.“

# User Stories

- > Anforderungen aus Sicht des/r Anwenders\_in
- > Bestehen aus
  - Name
  - Kurztext (Beschreibung)
  - Akzeptanzkriterien (erfüllbar!)
- > 3C-Kriterien
  - Card (Story muss auf eine Karte passen)
  - Conversation (Dokumentation des Ergebnisses der Anforderungserhebung)
  - Confirmation (überprüfbar und testbar)

# Zusammenfassung I

- > Wichtige Begriffe
- > Anforderung
- > Ziel
- > Requirements Engineering
- > Systemmodellierung

# Zusammenfassung II

- > Wasserfallmodell
- > V-Modell
- > Rational Unified Process
- > Agile Methoden



# Zusammenfassung III

- > Ziele im RE
- > 3 Arten von Anforderungen

## Ausblick

VO – 02	Mo., 21.10.2013	17:30 – 21:00	4	Anforderungen ermitteln Rollen, Faktoren, Techniken Anforderungen formulieren Vorbereitungen zur guten Dokumentation
---------	-----------------	------------------	---	--

## Literatur

- > Klaus Pohl; Requirements-Engineering: Grundlagen, Prinzipien, Techniken; dpunkt.verlag; 2008; 2. Auflage
- > Christof Ebert; Systematisches Requirements Engineering: Anforderungen ermitteln, spezifizieren, analysieren und verwalten; dpunkt.verlag; 2010; 3. Auflage
- > Tim Weilkiens; Systems Engineering mit SysML/UML: Modellierung, Analyse, Design; dpunkt.verlag; 2008; 2. Auflage
- > Chris Rupp; Requirements-Engineering und -Management: Professionelle, Iterative Anforderungsanalyse für die Praxis; Hanser Verlag; 2009; 5. Auflage
- > Jutta Eckstein; Agile Softwareentwicklung mit verteilten Teams; dpunkt.verlag; 2009
- > Uwe Vigerschow, Björn Schneider; Soft Skills für Softwareentwickler, Fragetechniken, Konfliktmanagement, Kommunikationstypen und -modelle; dpunkt.verlag; 2007
- > Uwe Vigerschow, ...; Soft Skills für IT-Führungskräfte und Projektleiter – Softwareentwickler führen und coachen, Hochleistungsteams aufbauen; dpunkt.verlag; 2009
- > Marcus Grande; 100 Minuten für Anforderungsmanagement; Vieweg & Teubner, Springer Fachmedien; 2011
- > McConnell S.; Aufwandschätzung bei Softwareprojekten; Microsoft Press Deutschland; 2006
- > Klaus Pohl & Chris Rupp; Basiswissen Requirements Engineering; dpunkt.verlag; 3. korrigierte Auflage; 2011
- > Helmut Balzert; Softwaretechnik: Basiskonzepte und Requirements Engineering; Spektrum Akademischer Verlag; 2009; 3. Auflage