Software Engineering

Studiengänge Informatik /Wirtschaftsinformatik
mediaProject

Herzlich Willkommen

Prof. Dr.-Ing. Anna Sabine Hauptmann

hauptman@informatik.htw-dresden.de

www.htw-dresden.de/~hauptman/mediaProject

Was erwarten Sie?/ Was wollen Sie lernen?

Verbindung praktischer Erfahrung mit theoretischem Hintergrund
Methodik, Bewährtes, Rahmenwerke
Weg bis zur ersten Zeile <<<<<<< Code ? → Anforderungen
Wie geht man nach Lehrmeinung vor?
Wie dokumentiert man gut?

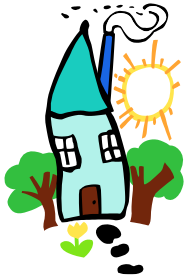
Wie werden Software-Systeme entwickelt?

Was ist „Software-Engineering“ ?

Was ist das Spezielle der Analyse in der (Software-)System-Entwicklung?

Warum haben Analyse und Definition von Anforderungen an das SW-System so große Bedeutung im SW-Entwicklungsprozess?

Ingenieurtechnische Produkt-Herstellung → Die „Bau-Methapher“ SW



Bauherr

Idee: Bau eines Hauses

Idee

Vision

Bau-Architekt

- Wohnhaus, Bürohaus, Garage...
- Reihenhaus, Doppelhaus, ...
- Flachdach, mehrere Etagen
- Anzahl und Nutzung der Räume
- ...

Was ?

Analyse

Bauingenieur

- Baumaterialien:
  Ziegel, Yton, Holz,
- Technologien:
  Bauteile oder Ortbeton verwenden
- ...

Wie ?

Entwurf

Systementwurf

Komponentenentwurf

Maurer,
Zimmermann ,
Klempner, ...

baggern, mauern, putzen, ...

bauen

implementieren

Vertreter der Bauaufsicht

Rohbauabnahme/Endabnahme

testen

testen

Bauherr

bezugsfertiges Haus

→ **GUI, DB, Logik, …**

Abstraktion

Ingenieurtechnische Produkt-Herstellung → Die „Bau-Methapher“ SW

Bauherr

Idee: Bau eines Hauses

Idee

Bau-Architekt

- Wohnhaus, Bürohaus, Garage...
- Reihenhaus, Doppelhaus, ...
- Flachdach, mehrere Etagen
- Anzahl und Nutzung der Räume
- ...

Was ?

Bauingenieur

- Baumaterialien:
  Ziegel, Yton, Holz,
- Technologien:
  Bauteile oder Ortbeton verwenden
- ...

Wie ?

Maurer,
Zimmermann ,
Klempner, ...

baggern, mauern, putzen, ...

bauen

Vertreter der Bauaufsicht

Rohbauabnahme/Endabnahme

testen

Bauherr

bezugsfertiges Haus

Vision

Analyse

Entwurf
Systementwurf
Komponentenentwurf

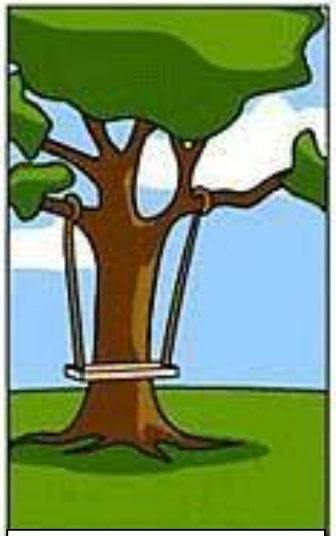
implementieren

testen
→ **GUI, DB,**
Logik, ...

Abstraktion

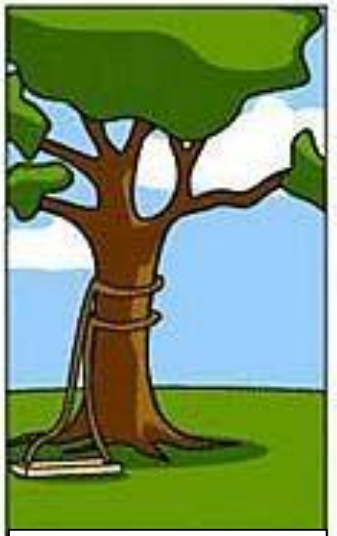
Software-Engineering

Was der Kunde erklärte



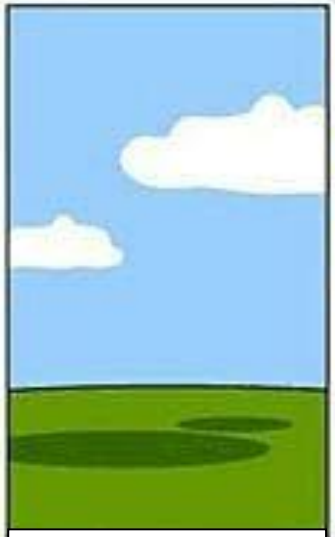
Was der Projekt-leiter verstand

Was der Analytiker beschrieb

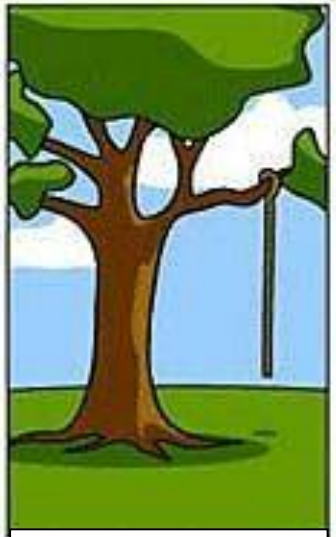


Was der Entwerfer plante

Was der Programmie-rer programmierte

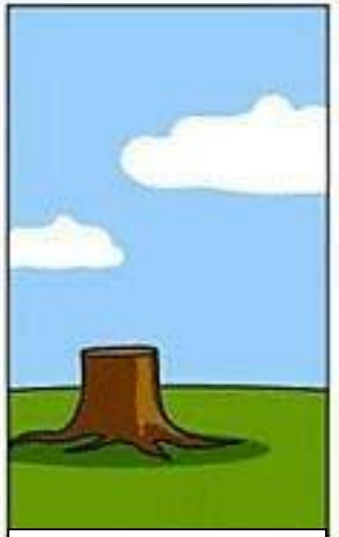


Wie das Projekt do-kumentiert wurde



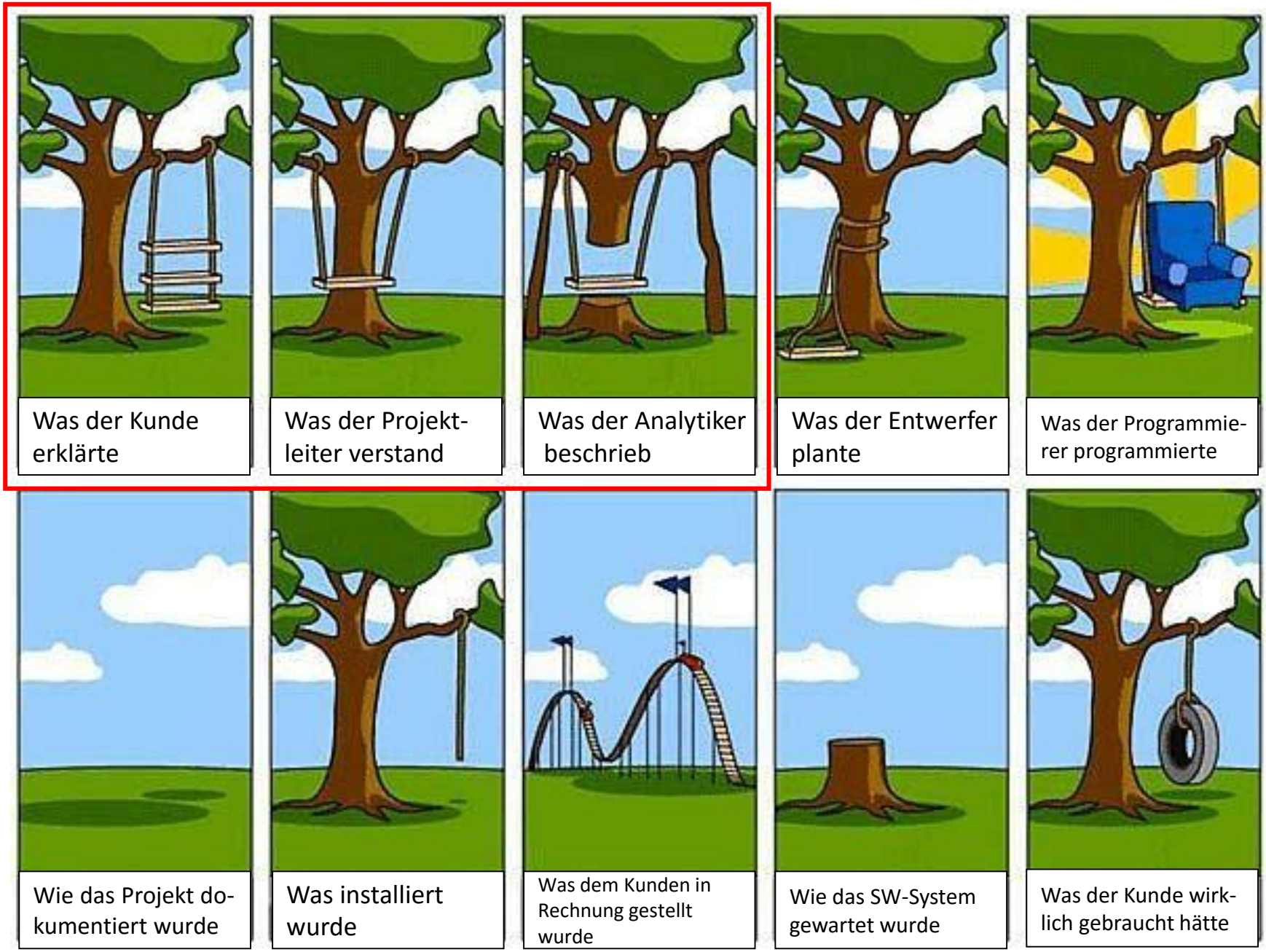
Was installiert wurde

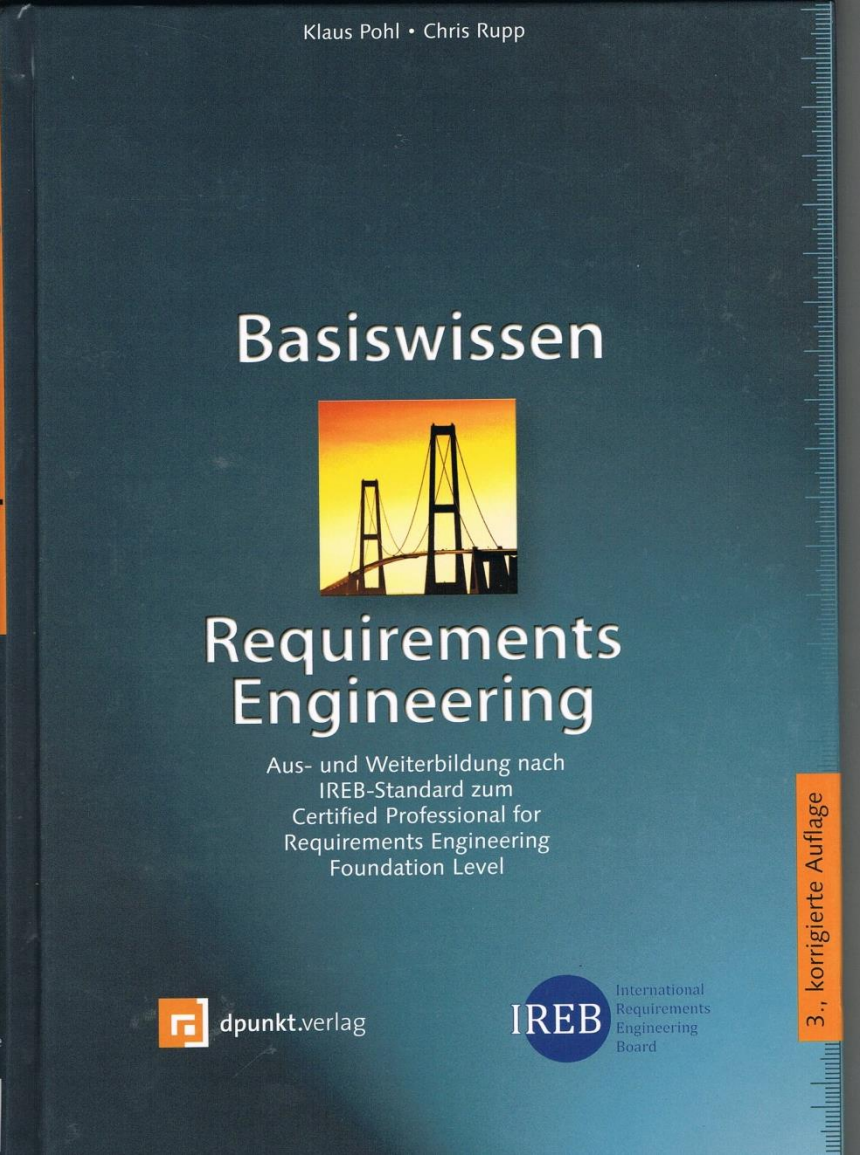
Was dem Kunden in Rechnung gestellt wurde



Wie das SW-System gewartet wurde

Was der Kunde wirk-lich gebraucht hätte


Was der Kunde erklärte
Was der Projekt-leiter verstand
Was der Analytiker beschrieb
Was der Entwerfer plante
Was der Programmie-rer programmierte
Wie das Projekt do-kumentiert wurde
Was installiert wurde
Was dem Kunden in Rechnung gestellt wurde
Wie das SW-System gewartet wurde
Was der Kunde wirk-lich gebraucht hätte



IREB e.V.:

International
Requirements
Engineering
Board

Zertifizierungsstelle

ISBN: 9 783 898 647 717

O Buch war (leicht) beschädigt

Thalia.de

Thalia Buchhandlung

Rugestr. 6 - 10
D-01069 Dresden
Tel. +49 351 4715078
Fax. +49 351 4715357
thalia.dresden-btu@thalia.de

Quittung

Basiswissen Requirements		
9783898647717		29,90 1
Positionsrabatt	10,00%	-2,99
SUMME (1)	EUR	26,91
EC-Karte	EUR	26,91

Kartenzahlung

electronic cash / girocard

Terminal-ID:	53505852
Datum/Uhrzeit:	15.09.2012/11:19
TA-Nr.:	012956
Belegnummer:	8083
Kartennummer:	xxxxxxxxxxxxxx7090
gültig bis:	12/16
Betrag:	26,91 EUR
VU-Nummer:	0515792300

Genehmigung erteilt

Betrag enthält 1,76 EUR MWSt:
1: 7,00%= 1,76 Netto: 25,15

St
Us

Buch & Kunst Gmbh
01069 Dresden VK 56395
Pohl,K.:Basiswissen Requirements En
KNV 30 194 356 978-3-89864-771-7 WG 9000
HTW, Fachbereich
29,90 EUR
LS 80274 vom 14.09.12 BZ 041250624300010
9 783898 647717

Ka 2
La

Vielen Dank für Ihren Besuch.
Auf Wiedersehen!
Im Internet: http://www.thalia.de

Erfolgsquote von Software-Projekten gestern und heute

Studie des Verteidigungsministeriums der USA
(Ende der 60-ziger Jahre)

80% werden nie abgeschlossen

15% laufen nur nach aufwendigen, teuren Nachbesserungen

nur 5 % aller SW-Projekte laufen wie geplant

Standish Group, CHAOS Report

	1995	2006
30 %	aller Software-Projekte scheitern	20 %
53 %	werden nicht wie geplant realisiert und/oder die Kunden sind unzufrieden	46 %

Ingenieurtechnische Produkt-Herstellung → Die „Bau-Methapher“ SW

Bauherr

Idee: Bau eines Hauses

Idee

Vision

Bau-Architekt

- Wohnhaus, Bürohaus, Garage...
- Reihenhaus, Doppelhaus, ...
- Flachdach, mehrere Etagen
- Anzahl und Nutzung der Räume
- ...

Was ?

Analyse

Bauingenieur

- Baumaterialien:
  Ziegel, Yton, Holz,
- Technologien:
  Bauteile oder Ortbeton verwenden
- ...

Wie ?

Entwurf

Systementwurf

Komponentenentwurf

Maurer,
Zimmermann ,
Klempner, ...

baggern, mauern, putzen, ...

bauen

implementieren

Vertreter der Bauaufsicht

Rohbauabnahme/Endabnahme

testen

testen

→ **GUI, DB, Logik, …**

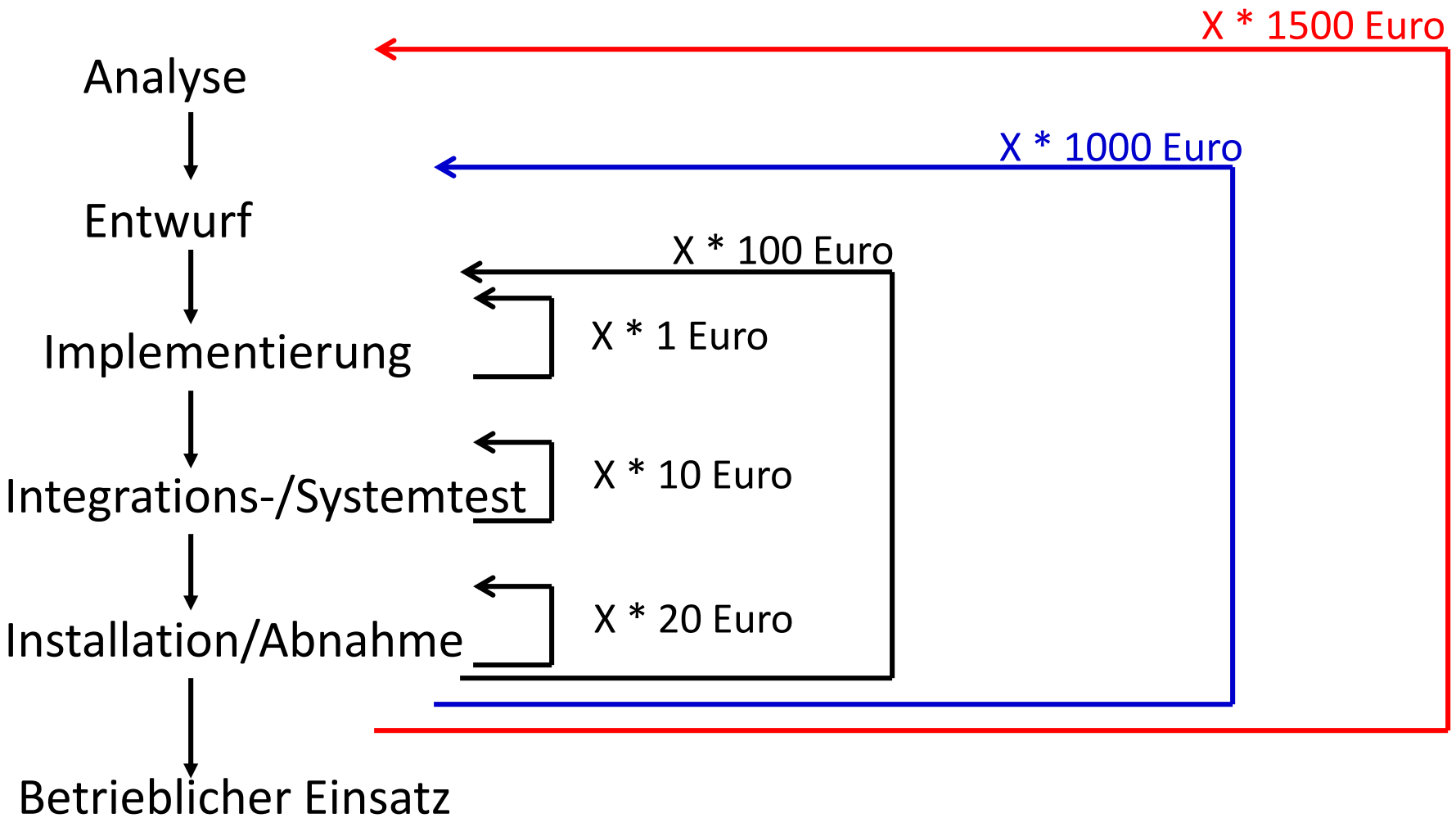
Bauherr

bezugsfertiges Haus

Systemanalyse im Kontext der SW-Entwicklung

60% der Fehler entstehen bereits in der Phase der Analyse

Relative Kosten der Fehlerbehebung



Systemanalyse im Kontext der SW-Entwicklung

60% der Fehler entstehen bereits in der Phase der Analyse.

Diese Fehler zu beheben,
verursacht die meisten Kosten im Rahmen der Fehlerbeseitigung.

Einige Definitionen des Begriffes „SW-Engineering“ –Teil 1

Naur, P. Randell, B. 1968 Report on a Conference, Garmisch	The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machine.
Parnas, D. L. 1974 SW-Engineering or Methodes for the Multi-Person Construction of Multi-Versions Programs Springer Verlag	SW-Engineering ist die Programmierung unter mindestens einer der zwei Bedingungen: - mehr als eine Person schreibt und benutzt das Programm - mehr als eine Fassung des Programmes wird erzeugt
Boehm, B. W. Classics in SW-Engineering 1979, Yourden Press	SW-Engineering ist die praktische Anwendung wissenschaftlicher Erkenntnisse auf den Entwurf und die Konstruktion von Computerprogrammen, verbunden mit der Dokumentation, die zur Entwicklung, Benutzung und Wartung der Programme erforderlich ist.

Einige Definitionen des Begriffes „SW-Engineering“ –Teil 2

Sommerville, I. Software Engineering 1985, Addison Wesley	Software-Engineering befasst sich mit dem Bau von SW-Systemen, die nicht von einem Entwickler allein hergestellt werden können. Software-Engineering beruht auf der Anwendung von ingenieurmäßigen Prinzipien und umfasst sowohl technische als auch nichttechnische Aspekte.
Sommerville, I. Software Engineering 2001, Addison Wesley	Das Software-Engineering ist eine technische Disziplin, die sich mit allen Aspekten der SW-Herstellung beschäftigt.
IEEE Std 610.12 (1990) Standard Glossary of Software Engineering Terminology 1990, IEEE New York	The Application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is the application of engineering to software.

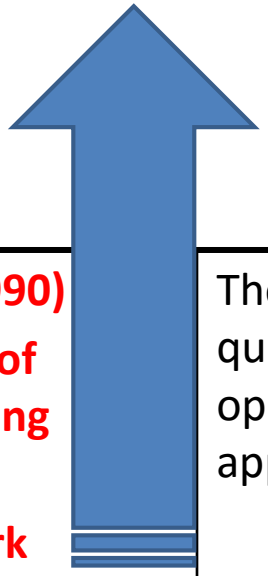
Einige Definitionen des Begriffes „SW-Engineering“ –Teil 2

siehe Unterlagen: **IEEE Std 610.12 (1990)**

Standard Glossary of Software Engineering Terminology

1990, IEEE New York

IEEE : **I**nstitute of **E**lectrical and **E**lectronics **E**ngineers



IEEE Std 610.12 (1990)

Standard Glossary of Software Engineering Terminology

1990, IEEE New York

The Application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is the application of engineering to software.

Einige Definitionen des Begriffes „SW-Engineering“ –Teil 3

Kahlbrandt, B. Software Engineering 1998, Springer	Software-Engineering ist • die Entwicklung, die Pflege und der Einsatz Qualitativ hochwertiger Software unter Einsatz von • wissenschaftlichen Methoden • wirtschaftlichen Prinzipien • geplanten Vorgehensmodellen • Werkzeugen • und quantifizierbaren Zielen
Ein Vorschlag:	Software-Engineering ist - die effektive und effiziente Entwicklung und Weiterentwicklung komplexer SW-Systeme - sowie begleitender Dokumente - in einem bewusst arbeitsteilig gestalteten Prozess - unter Anwendung bewährter Prinzipien, Methoden und Modellen.

Jochen Ludewig / Horst Lichter „Software Engineering“
Grundlagen, Menschen, Prozesse, Techniken

Auflage 1, dpunkt.verlag 2007 3-89864-268-2, S. 51f

Software-Engineering in der Informatik  Hygiene in der Medizin

Jochen Ludewig / Horst Lichter „Software Engineering“
Grundlagen, Menschen, Prozesse, Techniken

dpunkt.verlag 2007 3-89864-268-2, S. 51f

Software-Engineering in der Informatik  Hygiene in der Medizin

nützt nichts
sondern verhindert vielmehr Schäden
→ sollte generell beachtet werden

„Software Engineering ist – wie die Hygiene in der Medizin –
langweilig und frustrierend für Leute,
die die Abwehr von Fehlschlägen und Katastrophen
nicht als positive Leistung betrachten.“

Analyse versus Synthese in der (Software-) System-Entwicklung ?

Analyse

Auflösung

d.h. systematische Untersuchung

ausgehend von einer definierten Perspektive

WAS ?

analysieren
→ in Teile zerlegen,
um das Ganze zu verstehen

Synthese

Zusammensetzung

d.h. systematisches Implementieren

ausgehend von definierten Rahmenbedingungen

WIE?

synthetisieren
→ aus Teilen zusammensetzen,
um das Ganze zu bauen

Wie werden Software-Systeme entwickelt?

Bevor mit der Implementierung begonnen wird, werden die Anforderungen ermittelt und beschrieben. Danach wird ausgehend von der Beschreibung der Anforderungen eine Struktur festgelegt, nach der SW-System gebaut wird. Dabei begleiten ständig Test und Dokumentation die Arbeit.

Was ist Software-Engineering“ (siehe weitere Definitionen)?

Software-Engineering ist

- die effektive und effiziente Entwicklung und Weiterentwicklung komplexer SW-Systeme
- sowie begleitender Dokumente
- in einem bewusst arbeitsteilig gestalteten Prozess
- unter Anwendung bewährter Prinzipien, Methoden und Modellen.

Was ist das Spezielle der Analyse in der (Software-) System-Entwicklung ?
Analyse bedeutet nach DUDEN, Das große Fremdwörterbuch):
„Auflösung“, d.h. systematische Untersuchung eines Gegenstandes oder Sachverhaltes hinsichtlich seiner Eigenschaften aus einer definierten Perspektive.
Gegensatz: Synthese : Zusammensetzung, Verknüpfung, Verfahren zur Herstellung, …

Warum haben Analyse und Definition von Anforderungen an das SW-System große Bedeutung im Entwicklungsprozess?

Auch heute noch werden nur die Hälfte aller SW-Projekte wie geplant realisiert.

60% der Fehler im SW-system resultieren aus Fehlern in der Analysephase, d.h aus der Definition der Anforderungen.

Die Behebung von Fehlern aus der Analysephase sind vergleichsweise sehr teuer.

Was kennzeichnet ein System?

Warum sind Software-Systeme ganz spezielle Systeme?

Welche Grundprinzipien finden in der Software-Entwicklung Anwendung?

Welchen Unterschied gibt es zwischen Modellen und Prototypen bei der SW-Entwicklung?

→ VDI 3633 Blatt 1: System (VDI = Verein Deutscher Ingenieure)

Ein System ist eine abgegrenzte Anordnung von Komponenten, die miteinander in Beziehung stehen.

Es ist gekennzeichnet durch:

- die Festlegung seiner Grenze gegenüber der Umwelt (Systemgrenze), mit der es über Schnittstellen Materie, Energie, Informationen austauschen kann. (Systemein- und Systemausgangsgrößen)
- die Komponenten, die bei der Erhöhung der Auflösung selbst wiederum Systeme darstellen (Subsysteme) oder aber als nicht weiter zerlegbar angesehen werden (Elemente).
- die Ablaufstruktur in den Komponenten, die durch spezifische Regeln und konstante oder variable Attribute charakterisiert wird.
- die Relationen, die Systemkomponenten miteinander verbinden (Aufbaustruktur)
- die Zustände der Komponenten, die jeweils durch Angabe der Werte aller konstanten und variablen Attribute (Zustandsgrößen) beschrieben werden, von denen im allgemeinen nur ein kleiner Teil untersuchungsrelevant ist.
- die Zustandsübergänge der Komponenten als kontinuierliche oder diskrete Änderungen mindestens einer Systemvariablen aufgrund des in dem System ablaufenden Prozesses.

→ VDI 3633 Blatt 1: Prozess (VDI = Verein Deutscher Ingenieure)

Ein Prozess ist die Gesamtheit von aufeinander einwirkenden Vorgängen in einem System,
durch die Materie, Energie oder auch Informationen umgeformt, transportiert oder auch gespeichert werden.

→ VDI 3633 Blatt 1: Modell (VDI = Verein Deutscher Ingenieure)

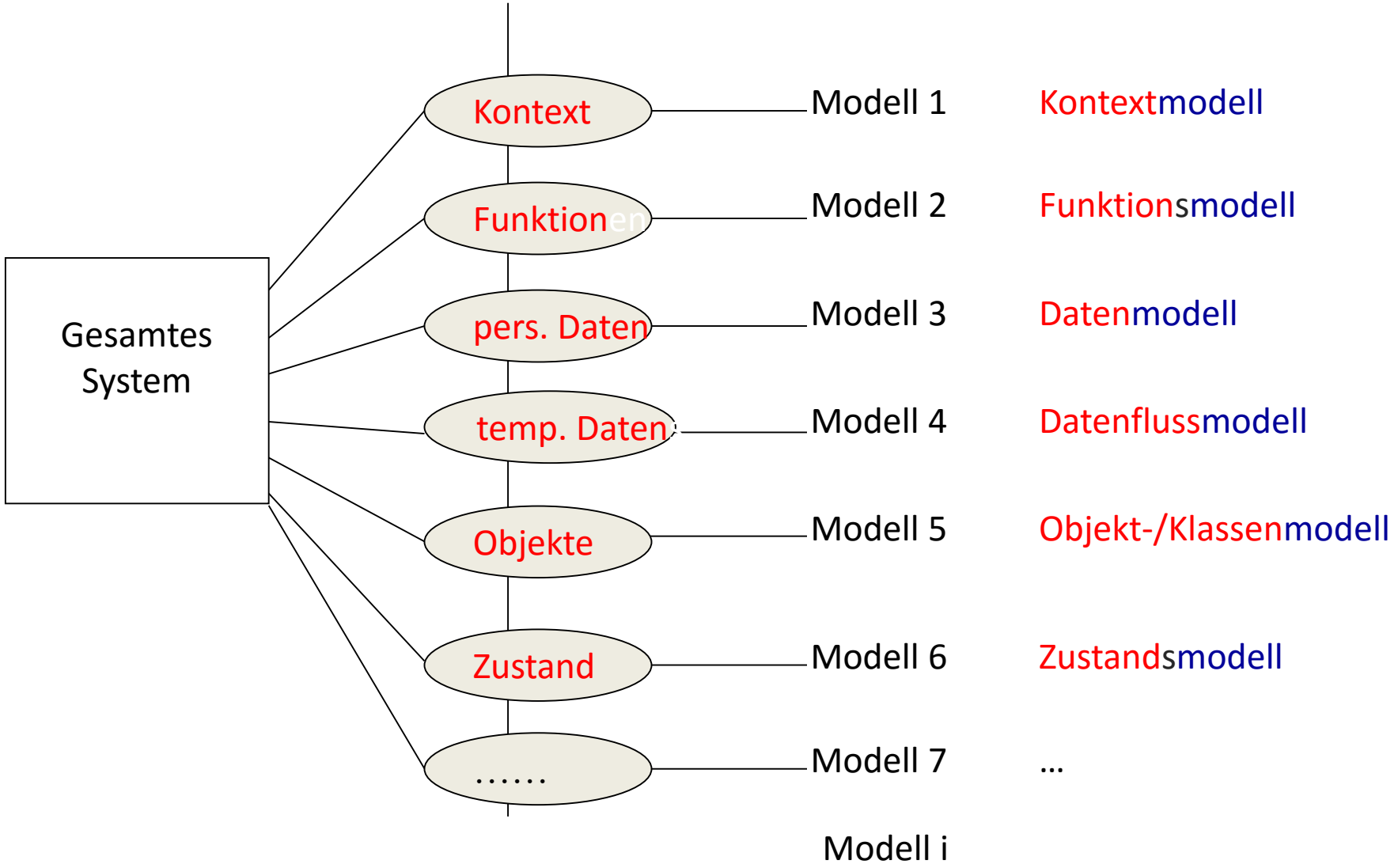
Ein Modell ist eine vereinfachte Nachbildung
eines geplanten oder real existierenden Originalsystems
mit seinen Prozessen
in einem anderen begrifflichen oder gegenständlichen System.

Es unterscheidet sich hinsichtlich der untersuchungsrelevanten Eigenschaften nur innerhalb eines vom Untersuchungsziel abhängigen Toleranzrahmens vom Vorbild.

begrifflicher Rahmen:
z.B. UML
Unified Modeling Language

siehe auch:
„Grundsätze des Modellierens“

Mögliche Untersuchungsziele bzw. Perspektiven führen zu speziellen Modellen.



pers. Daten : persistente Daten, gespeicherte Daten
temp. Daten: temporäre Daten, fließende Daten

Grundprinzipien der Systemanalyse

Abstraktion

Strukturierung

Zerlegung

Kapselung

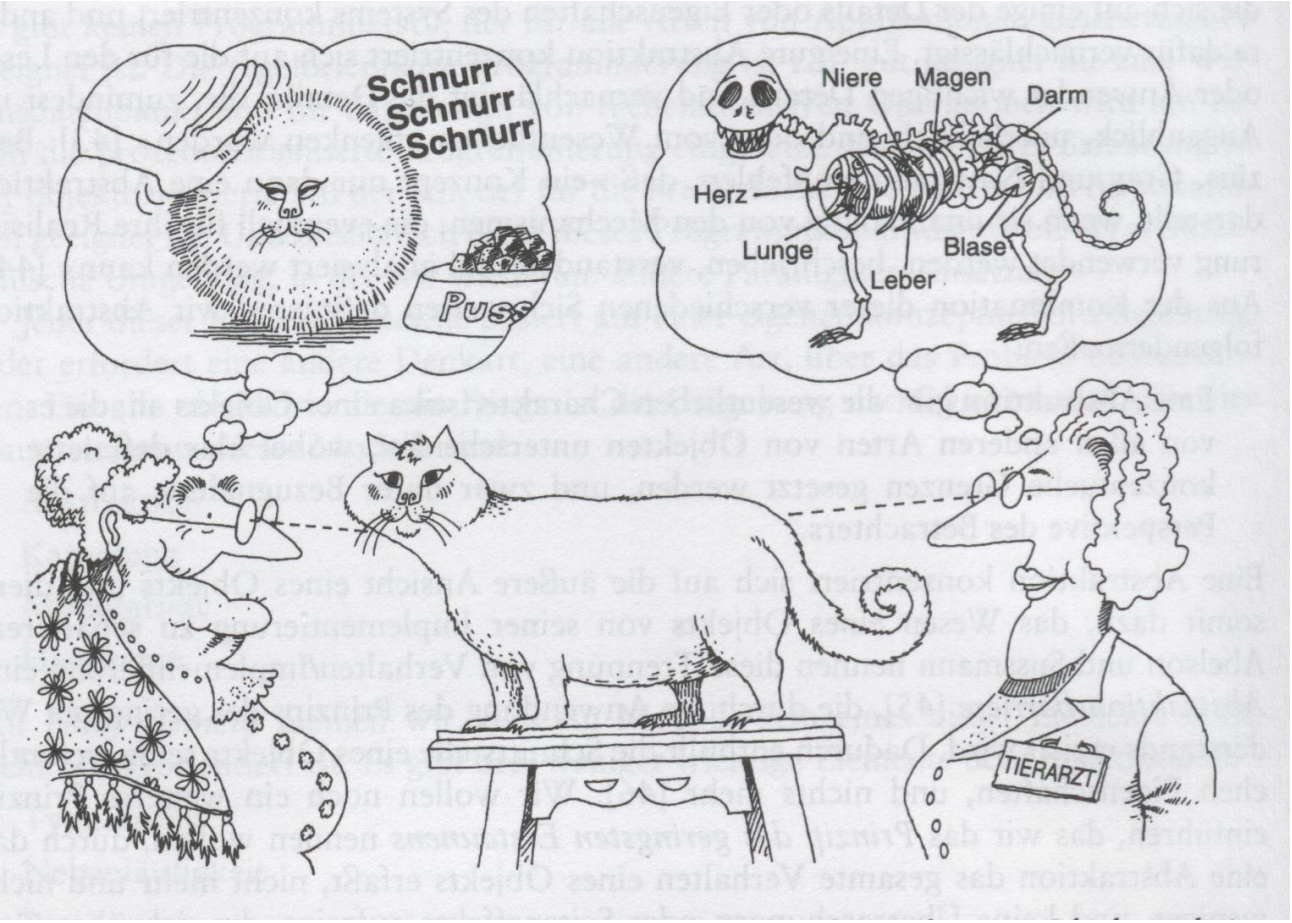
Hierarchisierung

Standardisierung

(integrierte) Dokumentation und Test

Diese Grundprinzipien finden auch im gesamten SW-Entwicklungsprozess Anwendung.

Abstraktion

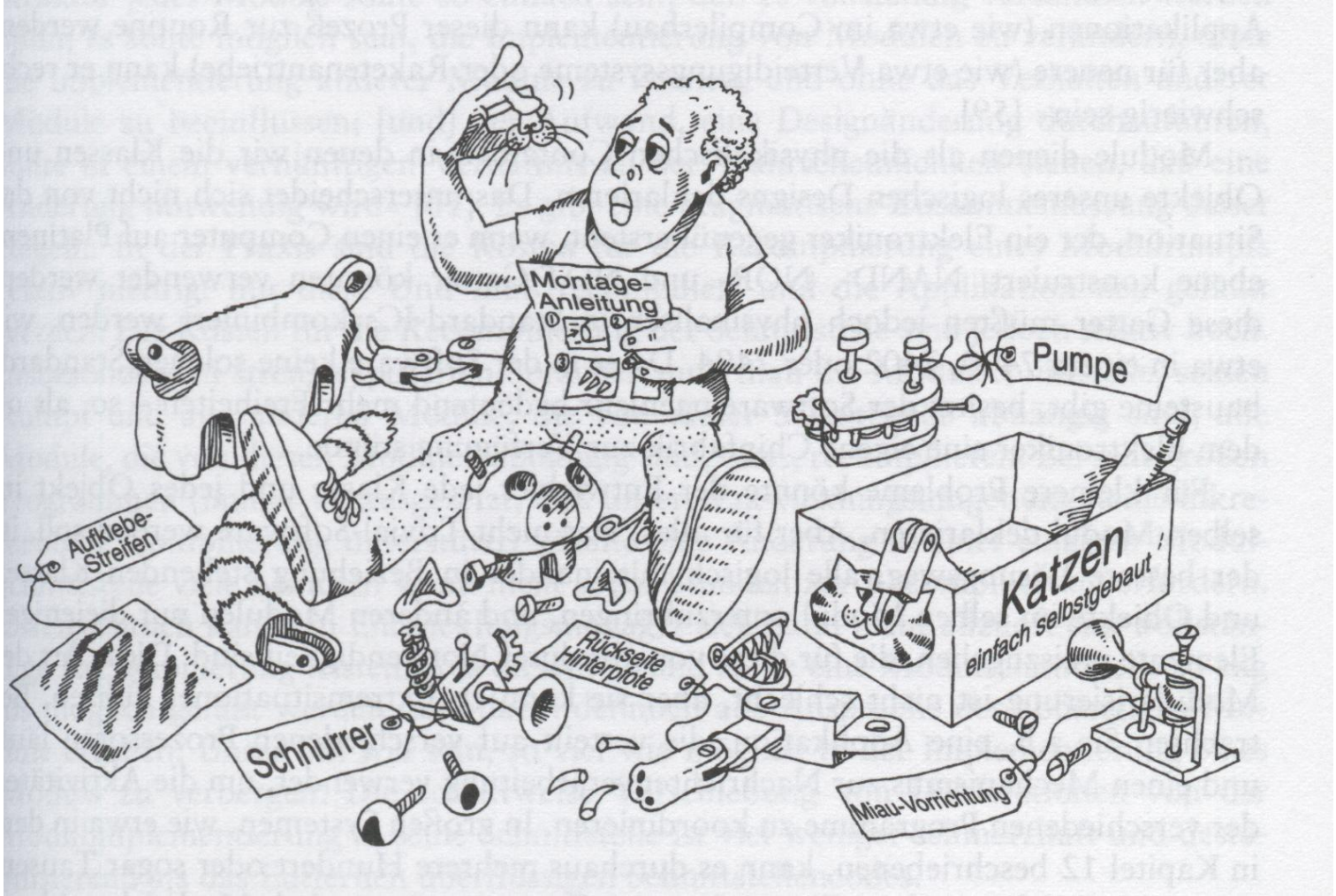


Booch, Grady „Objektorientierte Analyse und Design“ Addison Wesley, S. 62

Strukturierung

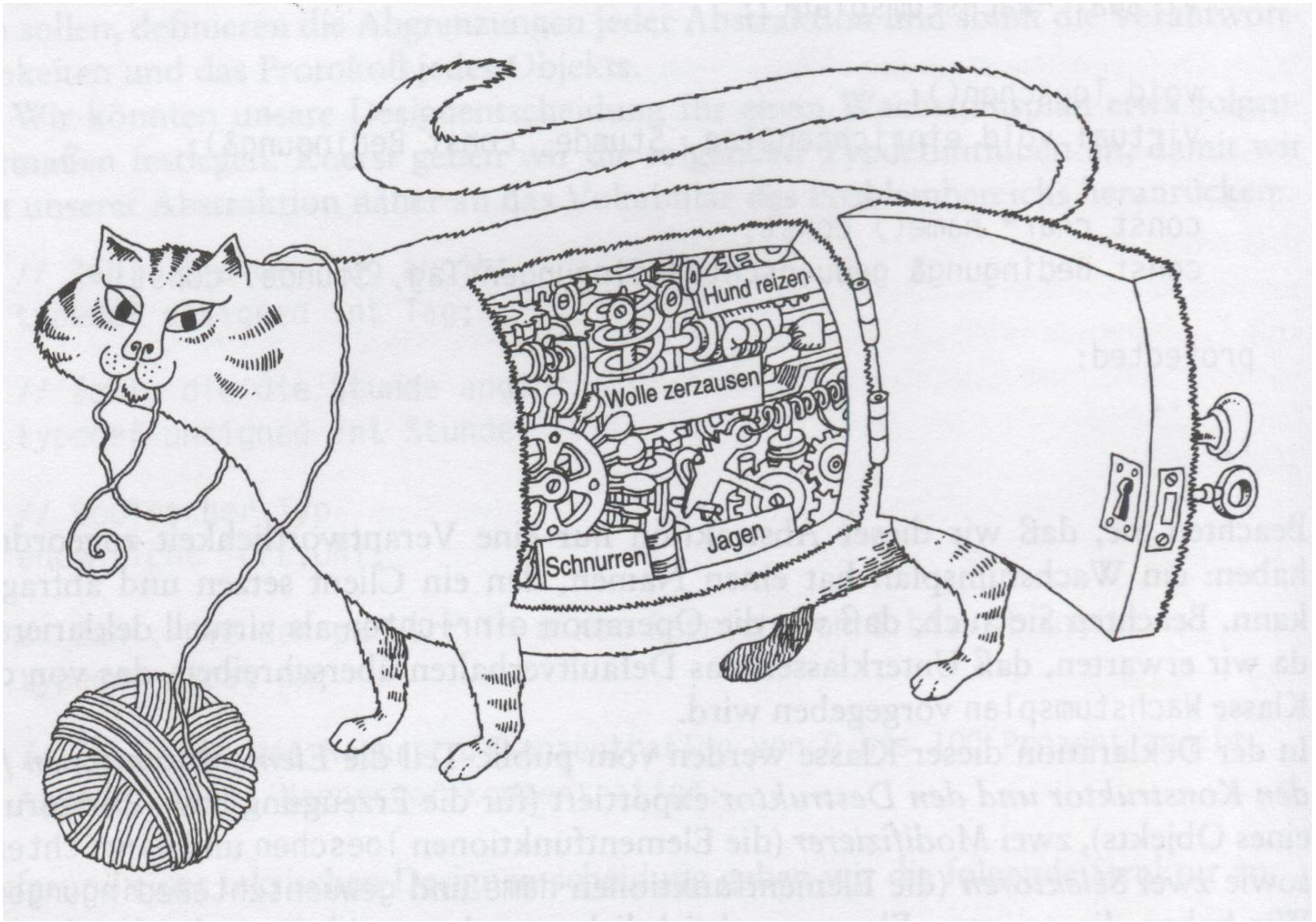
- Gliederung,
- Festlegung der Struktur,
  d.h. Festlegung der Komponenten und ihrer Beziehungen zueinander

Zerlegung - Zusammensetzung



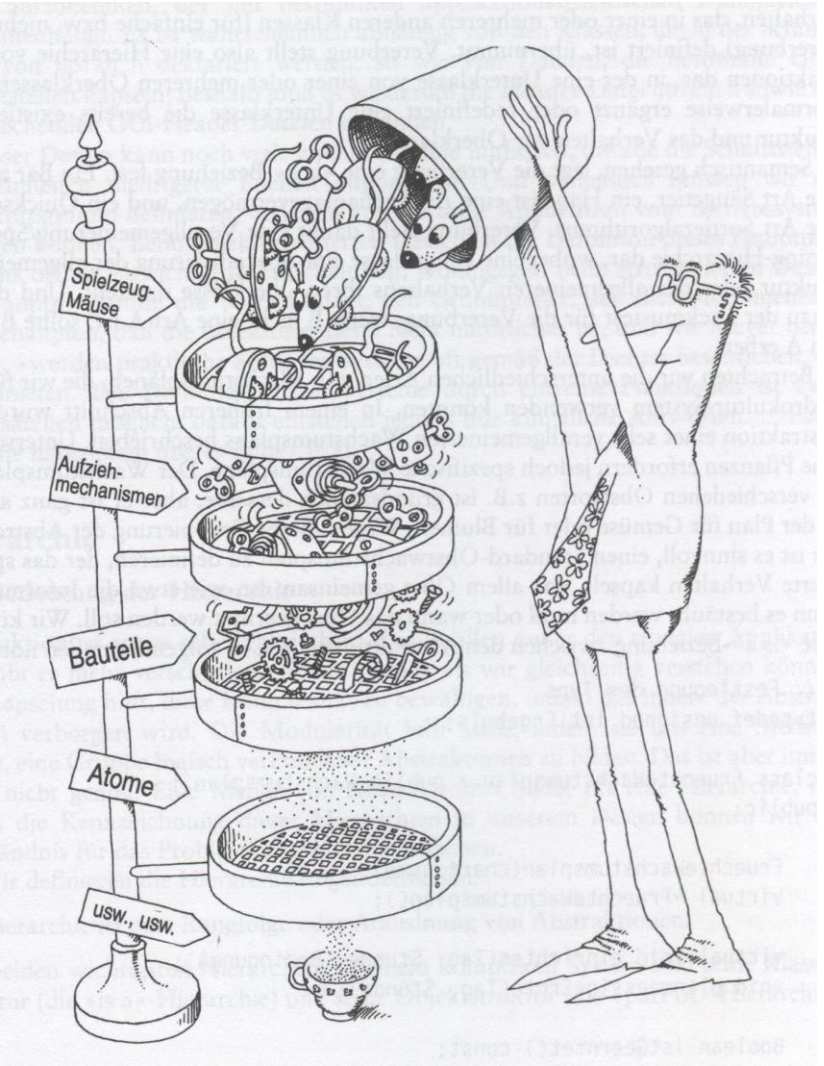
Booch, Grady „Objektorientierte Analyse und Design“ Addison Wesley, S. 77

Kapselung



Booch, Grady „Objektorientierte Analyse und Design“ Addison Wesley, S. 7

Hierarchisierung



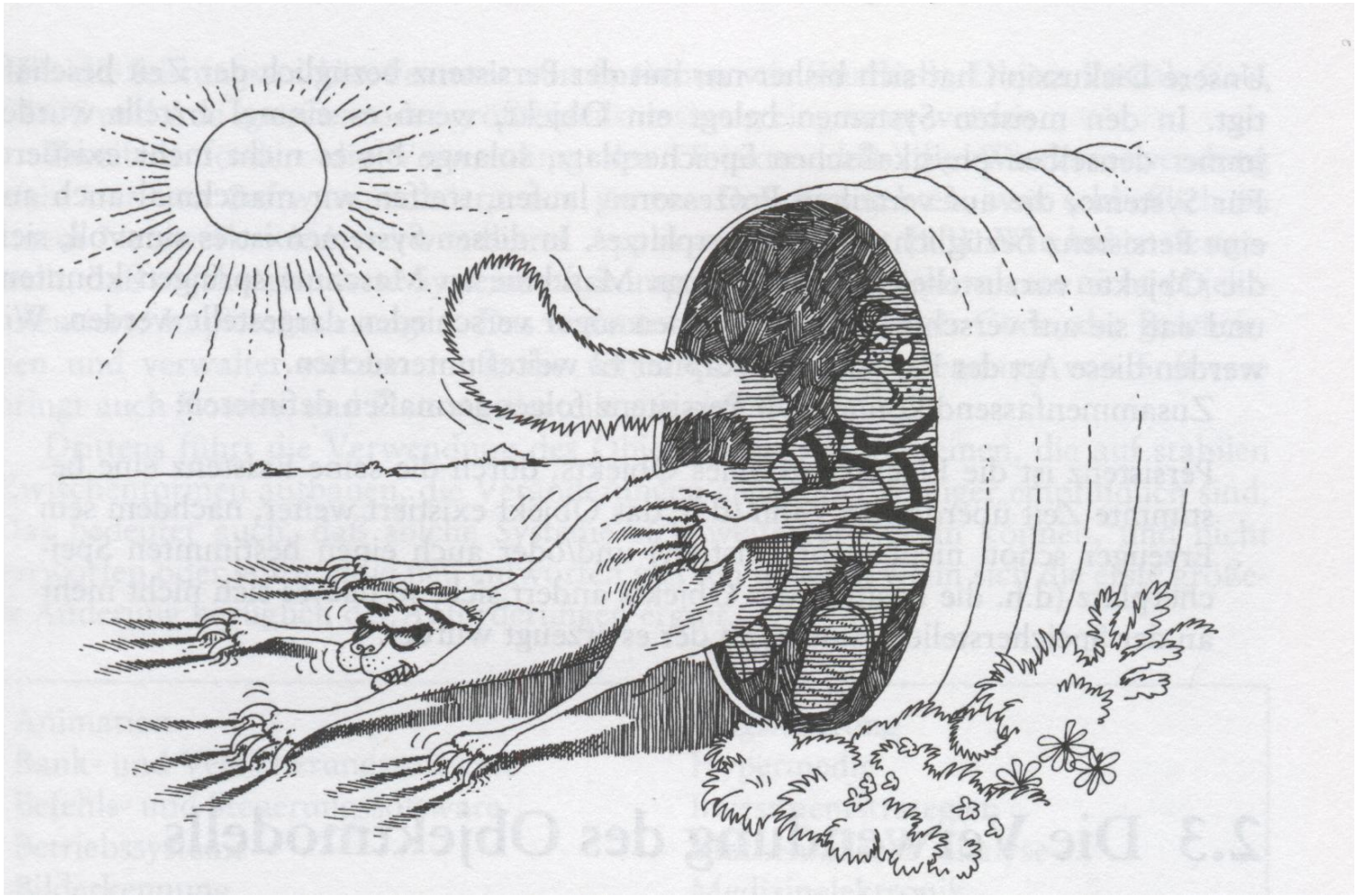
Booch, Grady „Objektorientierte Analyse und Design“ Addison Wesley, S. 84

Standardisierung

- verbindliche festgelegte Norm
  - z.B.:
    - ANSI
    - TCP/IP
    - …

- schwächere Form: Konvention
  - Funktionen werden mit Verben bezeichnet
  - Bezeichner für Interfaces (Schnittstellenklassen) beginnen mit I „Izustand"
  - Anforderungen werden mit Satzschablonen nach Chris Rupp beschrieben.
  - …

Persistenz



Booch, Grady „Objektorientierte Analyse und Design“ Addison Wesley, S. 103

integrierte Dokumentation und Test

Die Dokumentation zu den einzelnen Produkten sind in den Entwicklungsprozess integriert.

→ Jedes erarbeitete Ergebnis wird dokumentiert.

Beispiele:

- Lastenheft
- Pflichtenheft
- Architektur, Entwurf des SW-Systems
- Testplan
- Testprotokoll

→ Jedes erarbeitete Ergebnis wird überprüft, d.h. es wird getestet.

Beispiele:

- Sind die im Lastenheft formulierten Ziele die wirklichen Ziele?
- Sind die im Pflichtenheft formulierten Anforderungen die wirklichen Anforderungen?

→ Ziele messbar

Sind sie eindeutig und klar beschrieben? Sind sie vollständig, widerspruchsfrei?

- Ist die Struktur der Datenbank geeignet, um die erforderlichen Daten zu speichern?
- Welche Testfälle sind durchzuführen?
- Mit welchen Daten ist zu testen?
- Ist die Struktur des SW-Systems (Architektur/Entwurf) zur Aufgabenlösung geeignet?
- …

→ IEEE Std 610.12 (1990): Prototyp

Institute of **E**lectrical and **E**lectronics **E**ngineers

prototype.
A preliminary type, form, or instance of a system that serves as a model for later stages or for the final, complete version of the system.

prototyping.
A hardware and software development technique in which a preliminary version of part or all of the hardware or software is developed to permit user feedback, determine feasibility, or investigate timing or other issues in support of the development process.
See also: **rapid prototyping.**

rapid prototyping.
A type of prototyping in which emphasis is placed on developing prototypes early in the development process to permit early feedback and analysis in support of the development process.

→ IEEE Std 610.12 (1990): Prototyp

Quelle: Ludewig, Lichter „Software Engineering“, ISBN: 3-89864-268-2 Auflage 1 , S. 156:

- Ein Prototyp ist **lauffähig.**
- Ein Prototyp **realisiert ausgewählte Aspekte** des Zielsystems.
- Ein Prototyp wird von Klienten **geprüft und bewertet**.

→ IEEE Std 610.12 (1990): Prototyp

Quelle: Ludewig, Lichter „Software Engineering“, ISBN: 3-89864-268-2 Auflage 1 , S. 159:

Taxonomie des Prototyping von Floyd ((→ nach dem Ziel klassifiziert):

- exploratives Prototyping (explorativ – erforschen)
  → vorwiegend zur Unterstützung der Analyse

- experimentelles Prototyping
  → Überprüfung der technischen Realisierbarkeit unter den gegebenen Rahmenbedingungen (z.B. bzgl. der Antwortzeiten)

- evolutionäres Prototyping
  → Das SW-Projekt ist nie abgeschlossen, in einem Prozess entstehen in Zyklen an die neuen Anforderungen angepasste, weiter entwickelte SW-Systeme.

Was kennzeichnet ein System?

Ein System besteht aus Komponenten, die miteinander in Beziehung stehen.
Eine Grenze trennt das System von seiner Umgebung (auch Kontext genannt).
Schnittstellen (Verbindungsstellen) verbinden das System mit seiner Umwelt.
Ein System kann Subsysteme enthalten.

Warum sind Software-Systeme ganz spezielle Systeme?

SW-Systeme sind „immateriell“ und komplex; sie haben keine natürliche Lokalität.
SW-Systeme sind aus einem Werkstoff hergestellt, der von sich aus keine Strukturierung im „Großen“ erfordert.

Welchen Unterschied gibt es zwischen Modellen und Prototypen bei der SW-Entwicklung?

Modelle bilden ab; Prototypen sind lauffähig.

Welche Grundprinzipien finden in der Software-Entwicklung Anwendung?

Abstraktion, Strukturierung, Zerlegung, Kapselung, Hierarchisierung, Standardisierung, integrierte Dokumentation

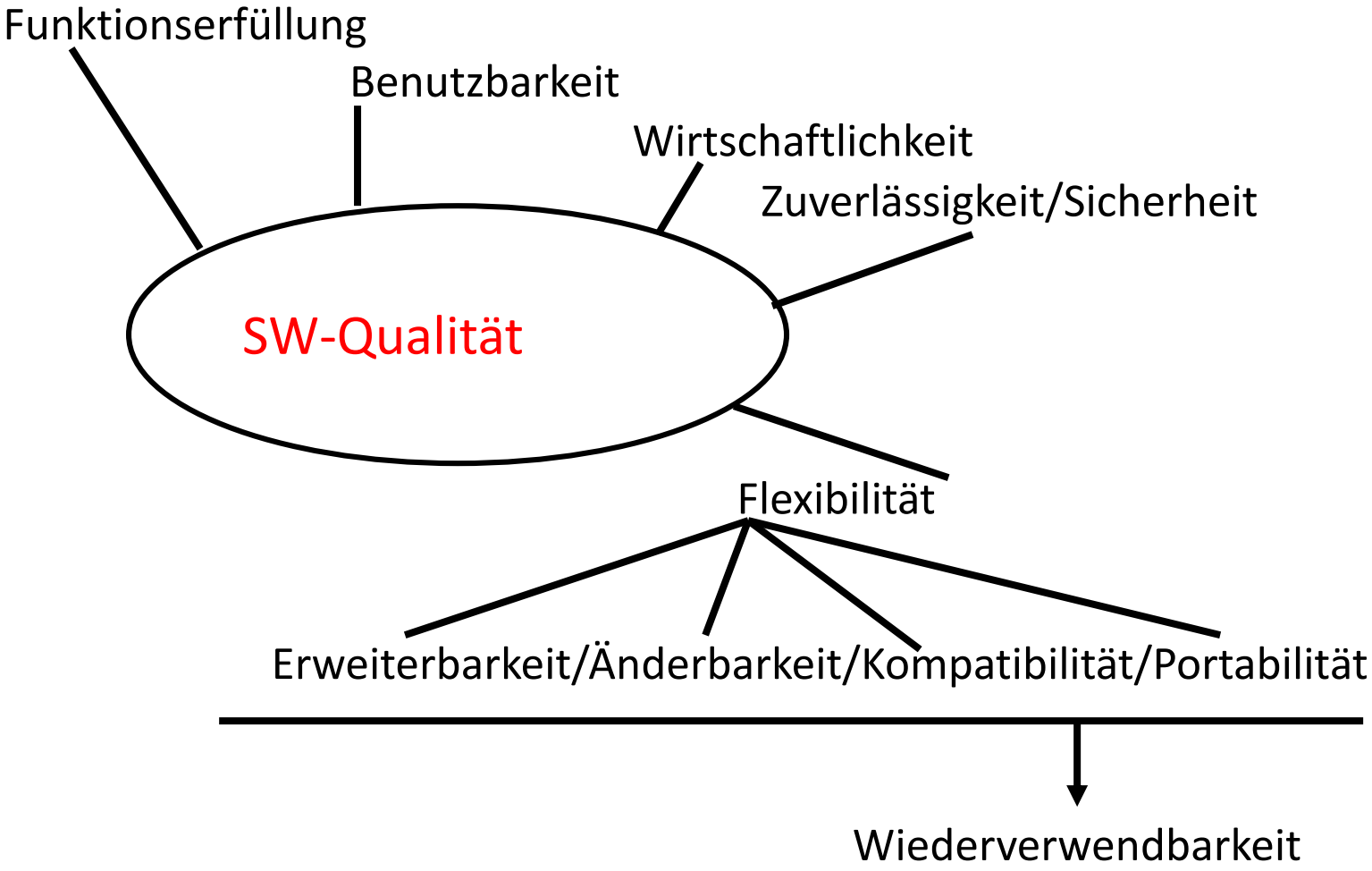
Welchen Unterschied gibt es zwischen Modellen und Prototypen bei der SW-Entwicklung?

Modelle bilden Sachverhalte statisch ab; Prototypen sind dynamisch, d.h. sie sind lauffähig.

Was kennzeichnet qualitativ hochwertige Software-Systeme?

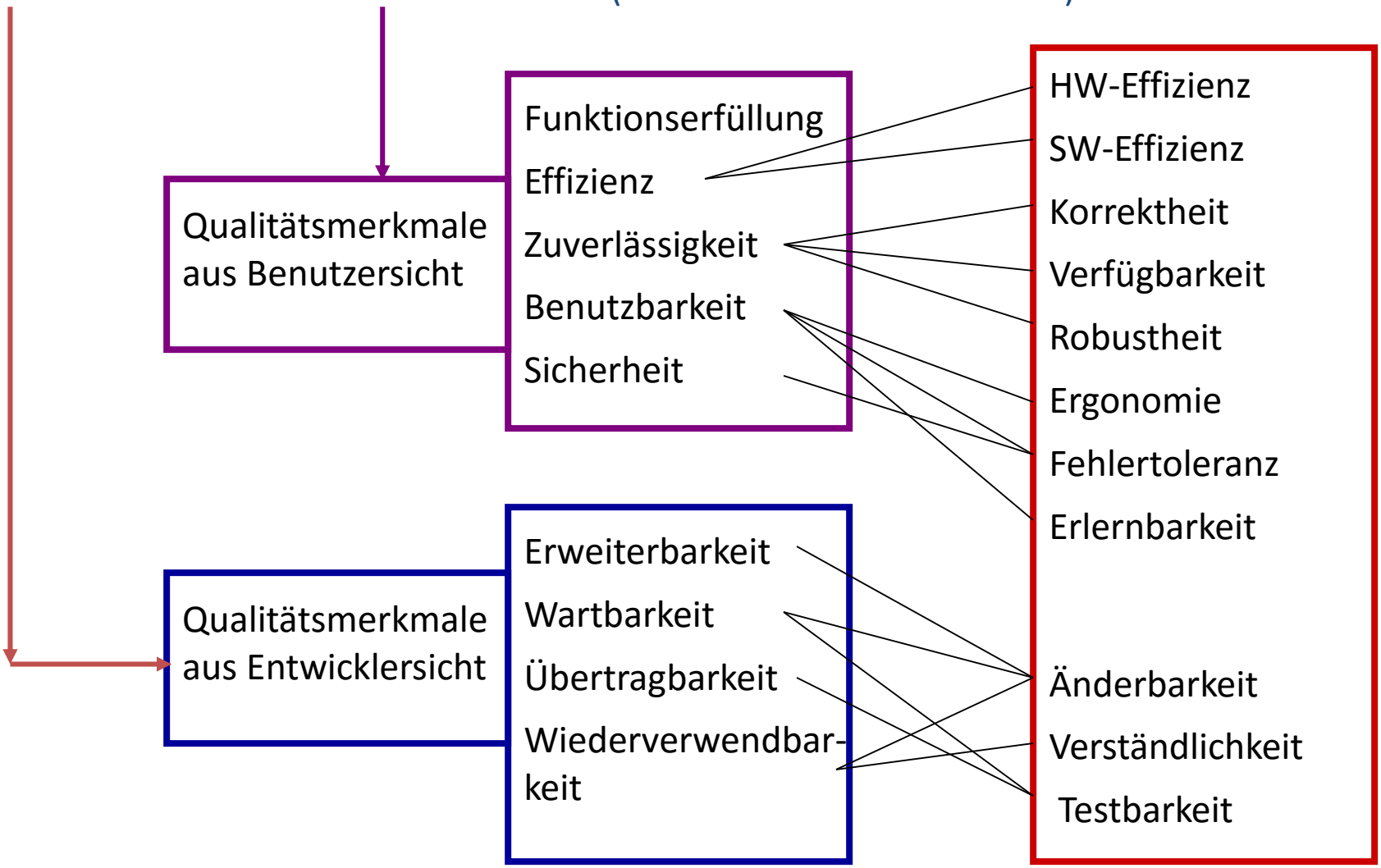
Wie kann hohe Software-Qualität erreicht werden?

Qualität ohne Trennung in Benutzersicht und Entwicklersicht



Qualität aus:

Benutzersicht und aus **Entwicklersicht** (z.B. nach Bernd Kahlbrandt)



äußere Qualitätskriterien und innere Qualitätskriterien

nach Bernd-Uwe Pagel, Hans-Werner Six: Software-Engineering Bd.1

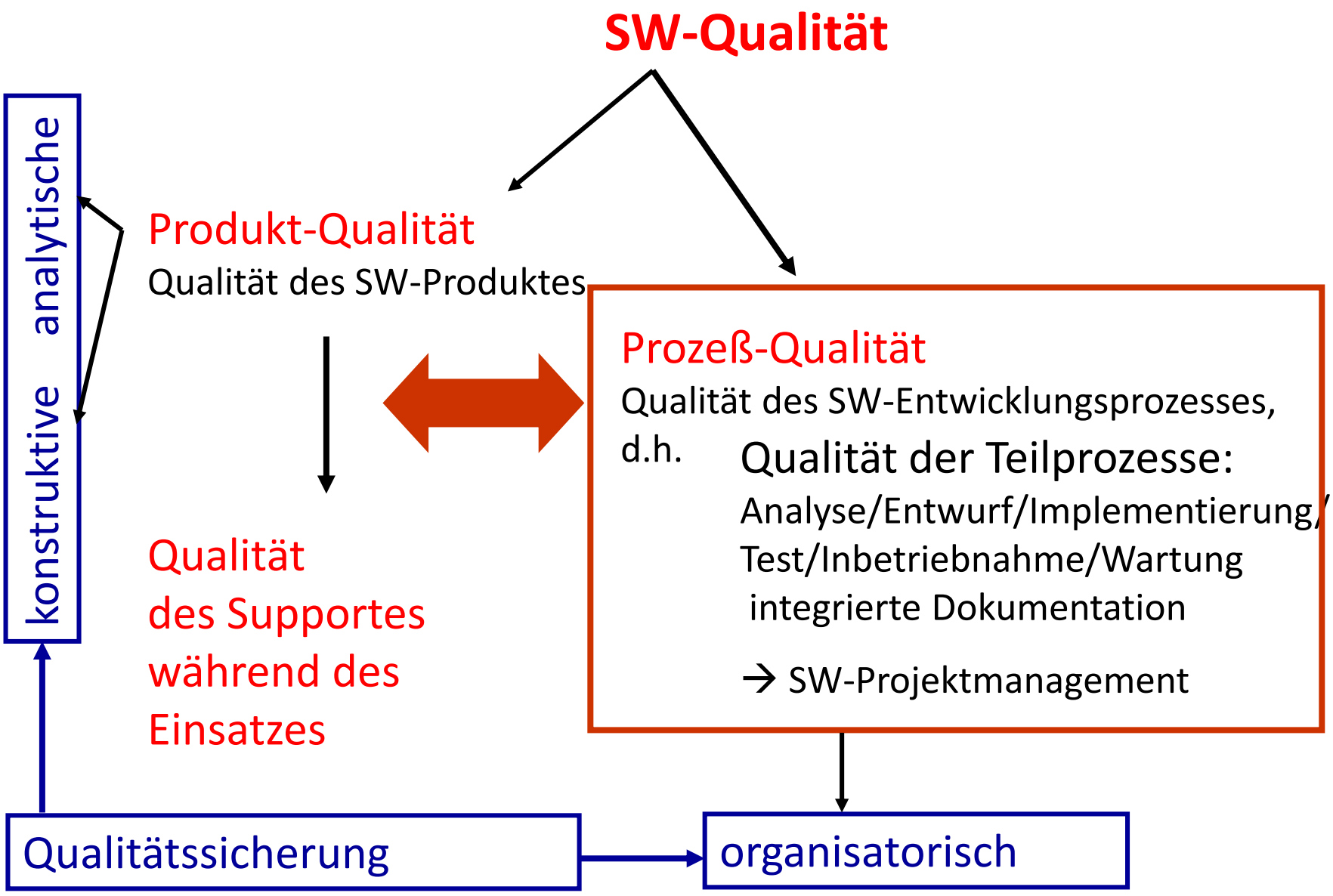
äußere Qualitätskriterien
(für Benutzer entscheidend):

- Korrektheit
- Benutzerfreundlichkeit
- Zuverlässigkeit
- Robustheit
- Effizienz

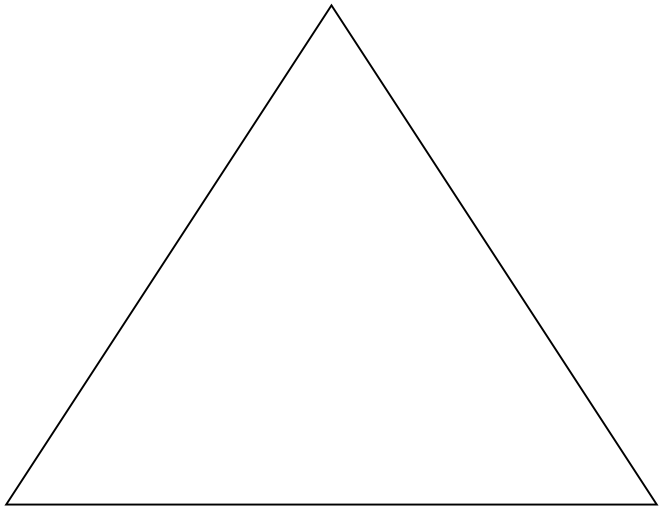
innere Qualitätskriterien
(für Entwickler entscheidend):

- Testbarkeit
- Wiederverwendbarkeit
- Änderbarkeit
- Erweiterbarkeit
- Portabilität/Kompatibilität

→ Wartbarkeit


SW-Qualität
konstruktive analytische
Produkt-Qualität
Qualität des SW-Produktes
Prozeß-Qualität
Qualität des SW-Entwicklungsprozesses,
d.h.
Qualität der Teilprozesse:
Analyse/Entwurf/Implementierung/
Test/Inbetriebnahme/Wartung
integrierte Dokumentation
→ SW-Projektmanagement
Qualität
des Supportes
während des
Einsatzes
Qualitätssicherung
organisatorisch

SW-Qualität

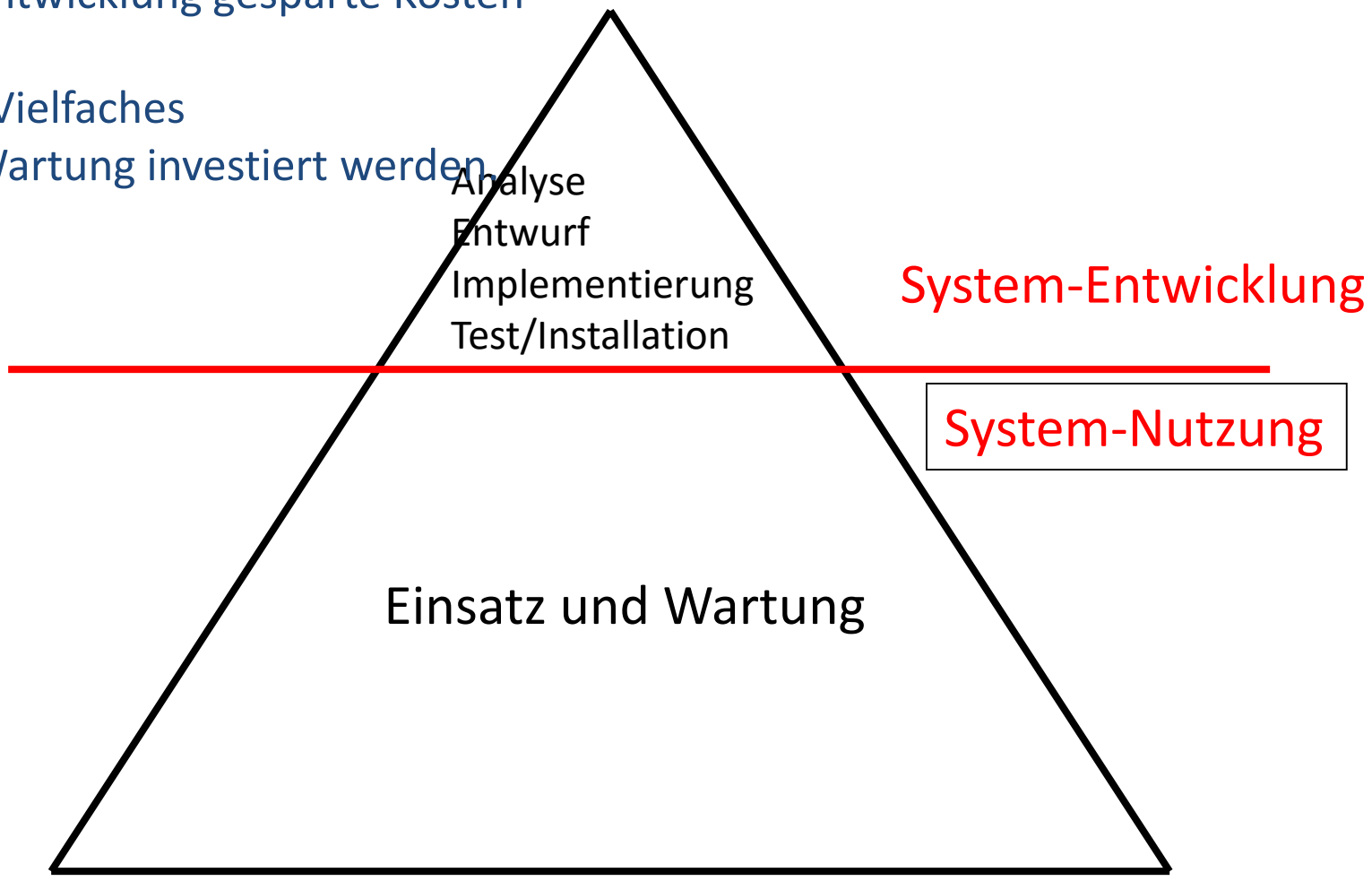


Kosten

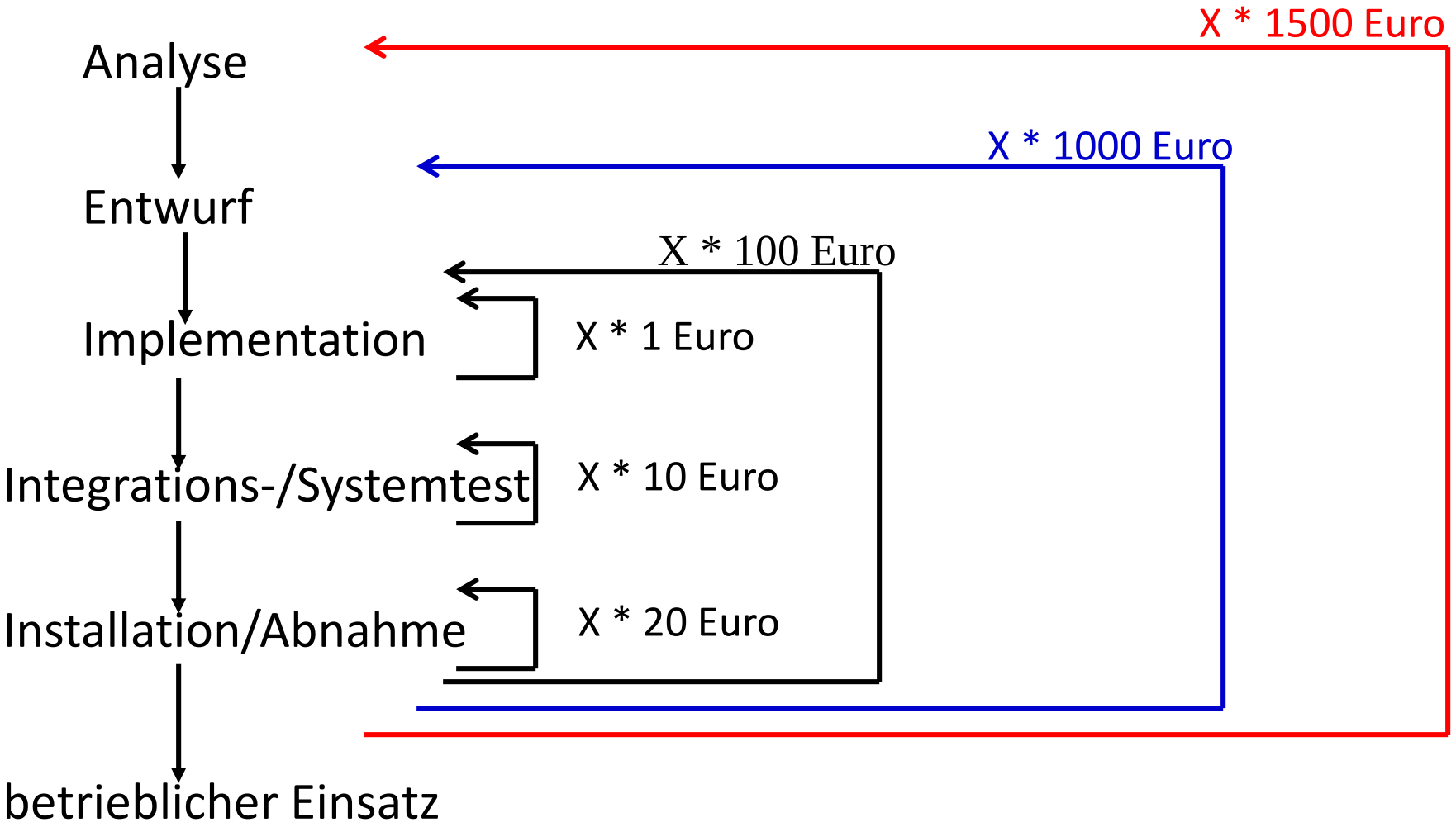
Zeit

nicht vergessen: SW-Qualität – erfordert Kosten und Zeit

„Eisberg-Effekt“:
In der Entwicklung gesparte Kosten
müssen
um ein Vielfaches
in der Wartung investiert werden.



zur Erinnerung: Relative Kosten zur Fehlerbehebung



Was kennzeichnet qualitativ hochwertige Software-Systeme?

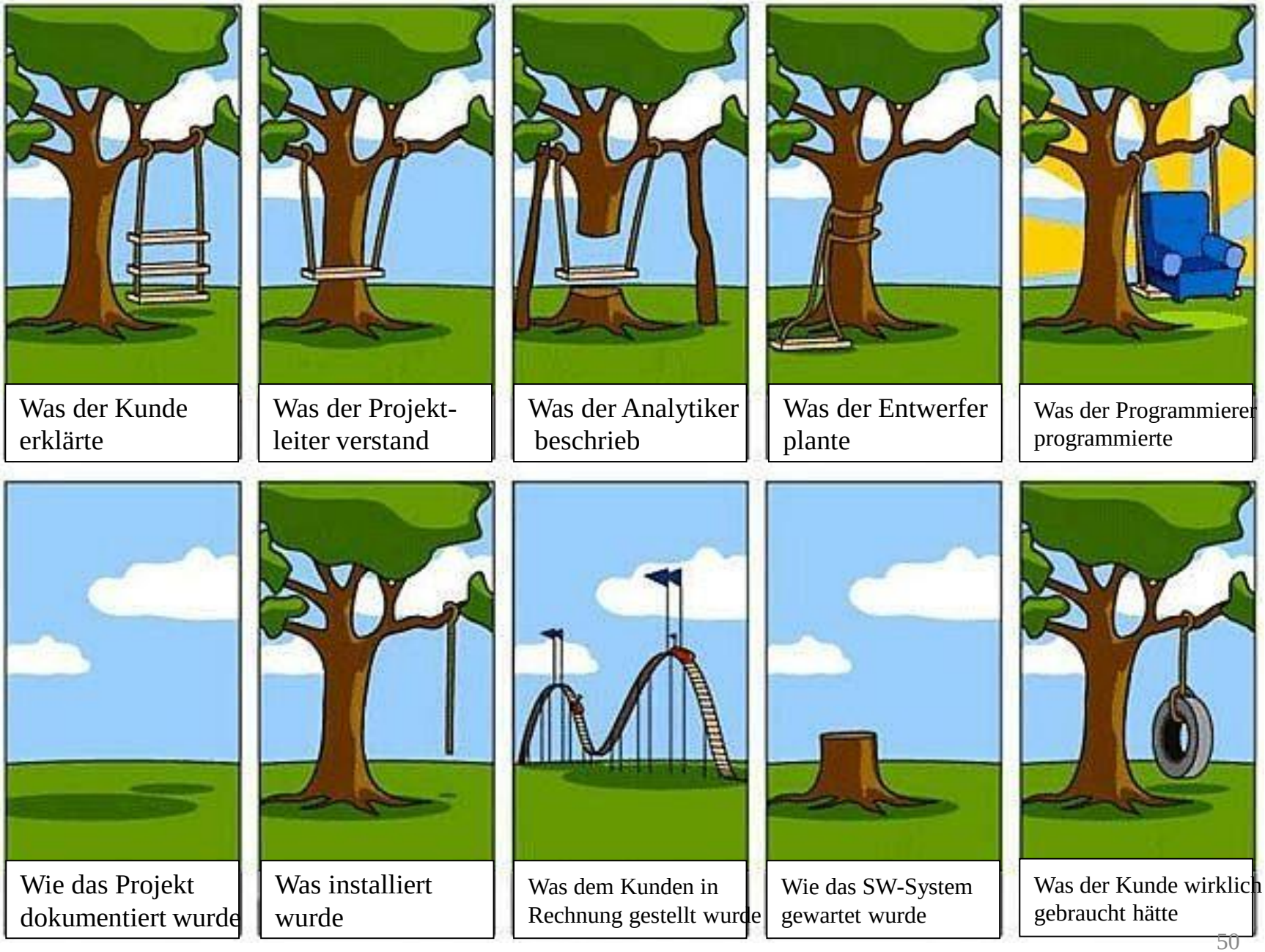
Funktionserfüllung, Benutzerfreundlichkeit,
Wirtschaftlichkeit
Je nach gesetzten Prioritäten:
Zuverlässigkeit, Sicherheit,
Flexibilität (Änderbarkeit, Erweiterbarkeit, Portabilität, Kompatibilität)
Wiederverwendbarkeit

Wie kann hohe Software-Qualität erreicht werden?

Durch gezielte Qualitätssicherung im
- organisatorischen Bereich
  (Vorgehen bei der Software-Entwicklung → Vorgehensmodelle)
- im Rahmen der Anforderungsmodellierung
- im konstruktiven Bereich
  (Entwurf, Implementierung → Muster, Schnittstellen, Werkzeuge…)
- im analytischen Bereich (Test)

Welche Rolle spielen Anforderungen des Kunden
im Rahmen der Software-Entwicklung?

Was eigentlich sind Anforderungen?


Was der Kunde erklärte
Was der Projekt-leiter verstand
Was der Analytiker beschrieb
Was der Entwerfer plante
Was der Programmierer programmierte
Wie das Projekt dokumentiert wurde
Was installiert wurde
Was dem Kunden in Rechnung gestellt wurde
Wie das SW-System gewartet wurde
Was der Kunde wirklich gebraucht hätte

Welche Rolle spielen Anforderungen des Kunden
im Rahmen der Software-Entwicklung?

- Anforderungen sind Ausgangspunkt der Entwicklung,
  daraus resultiert ihre große Bedeutung (sowohl inhaltlich als auch zeitlich).
- Die Anforderungen verbinden Kunden und Entwickler.
- Die Anforderungen sind Maß der Dinge bei der Übergabe des Produktes.

Was eigentlich sind Anforderungen?

IEEE Std 610.12 (1990)

Requirement

(1) A condition or capability needed by a user to solve a problem or achieve an objective.

(2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documets.

(3) A documented representation of a condition or capability as in (1) or (2).

Requirement analysis

(1) The process of studying user needs to arrive at a definition of system, hardware, or software requirements.

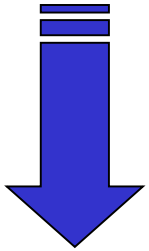
(2) The process of studying and refining system, hardware, or software requirements.

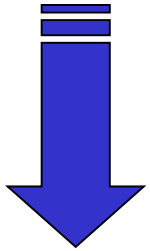
«A requirements is something that the product must do
or a quality that the product must have.»

Robertson, Suzanne/ Robertson, James
"Mastering the Requirements Process" – Seite 5f

Eine Anforderung ist (*aus der Sicht des Nutzers)* das,

was das Produkt tun soll oder die Qualität, die es haben soll



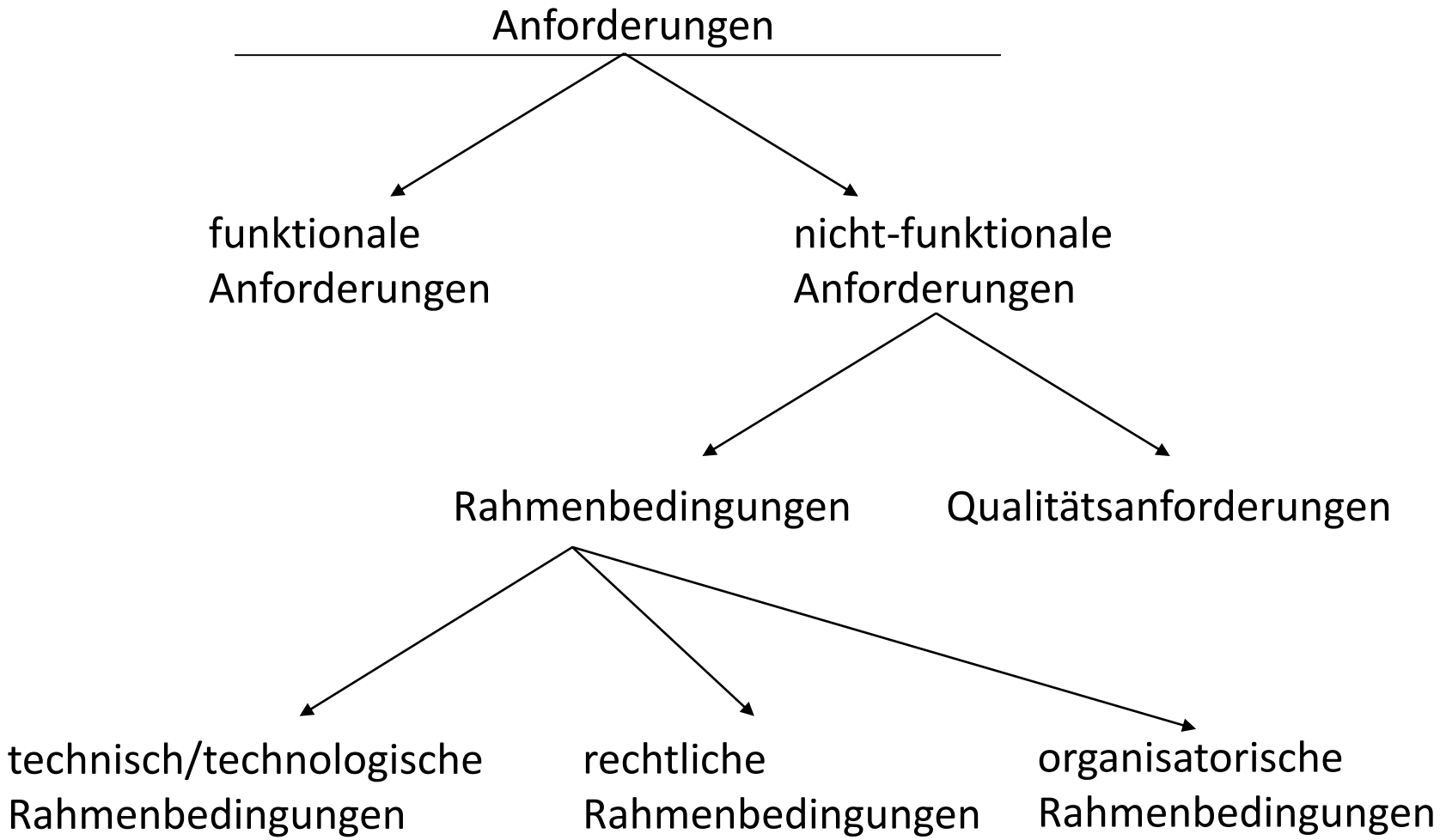


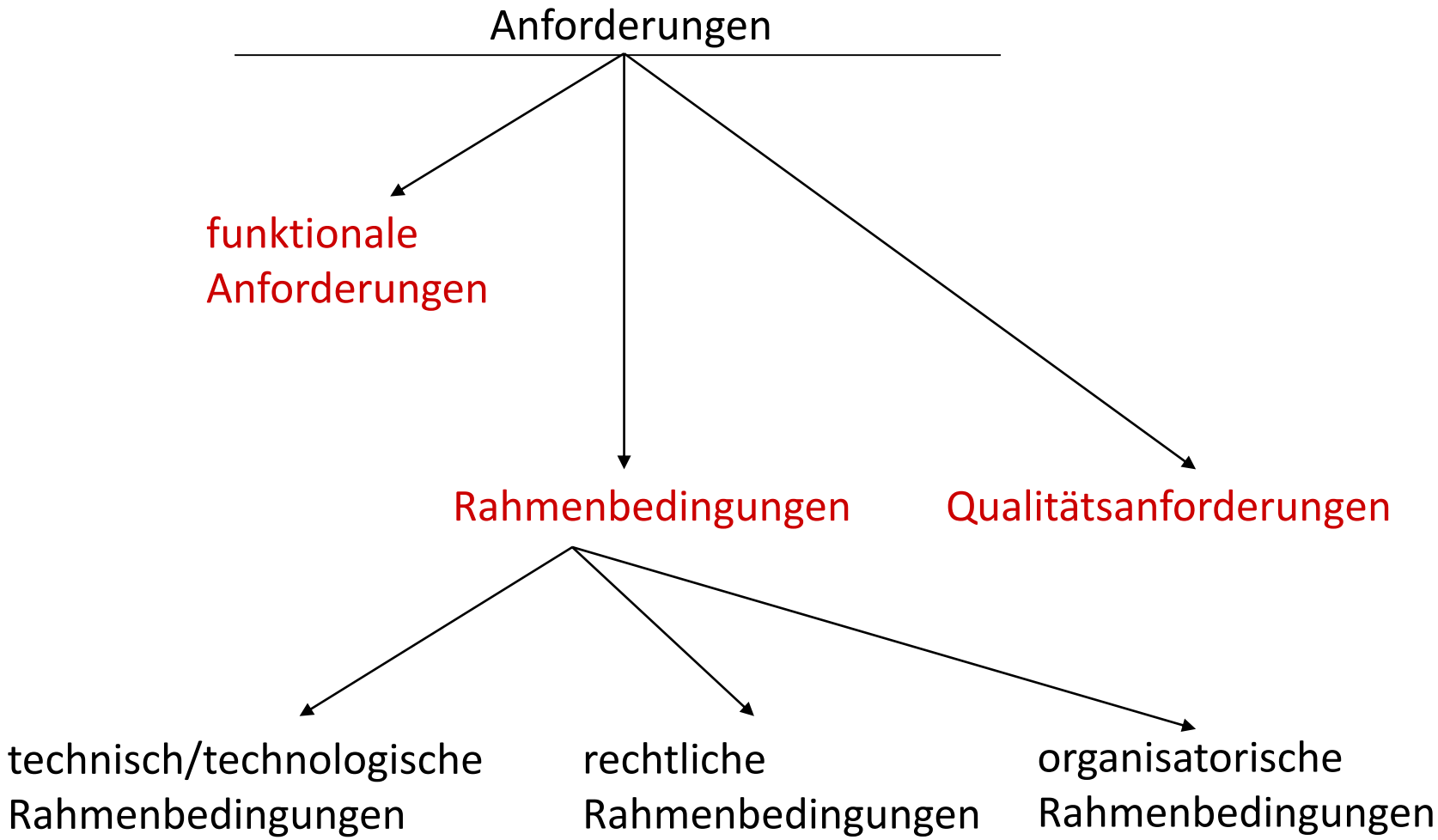
funktionale Anforderungen

nicht-funktionale Anforderungen

functional requirements

non-functional requirements


Anforderungen
funktionale Anforderungen
nicht-funktionale Anforderungen
Rahmenbedingungen
Qualitätsanforderungen
technisch/technologische Rahmenbedingungen
rechtliche Rahmenbedingungen
organisatorische Rahmenbedingungen



Nach: Pohl, Klaus „Requirements Engineering- Grundlagen, Prinzipien. Techniken“
ISBN 978-3-89864-342-9

Anforderung	Typ
Das SW-System muss es erlauben, Personen als Leser **anzumelden**.	F
Das SW-System muss es beliebigen Personen erlauben, Bücher im Katalog **zu suchen.**	F
Das SW-System muss es registrierten Benutzern erlauben, Bücher **auszuleihen.**	F
Das SW-System darf es *nicht* erlauben, eine Person als Benutzer **anzumelden**, **die bereits als Benutzer registriert ist .**	F/R-org
Das SW-System muss beim Suchen im Katalog zu einer Detaillierung der Suchanfrage **auffordern**, **wenn die Suchantwort mehr als 100 Treffer enthält.**	F/R-org
Das SW-System **hat** beim Suchen im Katalog **eine Antwortzeit von maximal 3 Sekunden.**	F/Q
Die **relationale Datenbank** zum Speichern der Daten ist bereits vorhanden; es **ist** eine **Oracle-Datenbank.**	R-tt
Das **Betriebssystem**, auf dem das SW-System zum Einsatz kommt, **ist LINUX.**	R-tt

Welche Rolle spielen Anforderungen des Kunden
im Rahmen der Software-Entwicklung?

- Anforderungen sind Ausgangspunkt der Entwicklung,
  daraus resultiert ihre große Bedeutung (sowohl inhaltlich als auch zeitlich).
- Die Anforderungen verbinden Kunden und Entwickler.
- Die Anforderungen sind Maß der Dinge bei der Übergabe des Produktes.

Was eigentlich sind Anforderungen?

Eine Anforderung ist eine Beschaffenheit oder Fähigkeit,
die von einem Benutzer zur Lösung eines Problems oder zur Erreichung
eines Zieles benötigt wird.
Es gibt funktionale und nicht-funktionale Anforderungen.
Diese nicht-funktionalen Anforderungen sind genau genommen

- Qualitätsanforderungen oder
- einschränkende Rahmenbedingungen
  (technisch/technologische, organisatorische, rechtliche Rahmenbedingungen).

Problematische Synonyme sind: Leistungsmerkmale, feature

Worin bestehen die Risiken der Anforderungsanalyse?

Wie kann den Risiken begegnet werden?

Diese Baumaßnahme
finanziert
die Deutsche
Kreditbank AG

Worin bestehen die Risiken der Anforderungsanalyse?

Verletzung der : Vollständigkeit, Korrektheit, Eindeutigkeit, Widerspruchsfreiheit, Realisierbarkeit, Angemessenheit, Minimalität

Wie kann den Risiken begegnet werden?

Durch

- bewusste Diskussion mit dem Kunden (→ soziale Kompetenz)
- klare, eindeutige Formulierung,
- Verwendung eines Glossars
- Ergänzung der Dokumente in natürlicher Sprache durch geeignete Modelle (→Dokumente mit grafischen Darstellungen)
- Verwendung von Werkzeugen, die die Konsistenz unterstützen
- Einbeziehung von Prototypen (experimentelle Prototypen, funktionaler Prototypen)

Kommunikation zwischen Menschen ....

.... also auch zwischen Kunde und Entwickler

Gedacht ist nicht zwingend gesagt !

Gesagt ist nicht zwingend gehört !

Gehört ist nicht zwingend verstanden !