

# Software Engineering

## SS 2005

Prof. Dr. Barbara Paech, Jürgen Rückert



Institut für Informatik  
Im Neuenheimer Feld 326  
69120 Heidelberg  
<http://www-swe.informatik.uni-heidelberg.de>  
[paech@informatik.uni-heidelberg.de](mailto:paech@informatik.uni-heidelberg.de)



RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG



### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

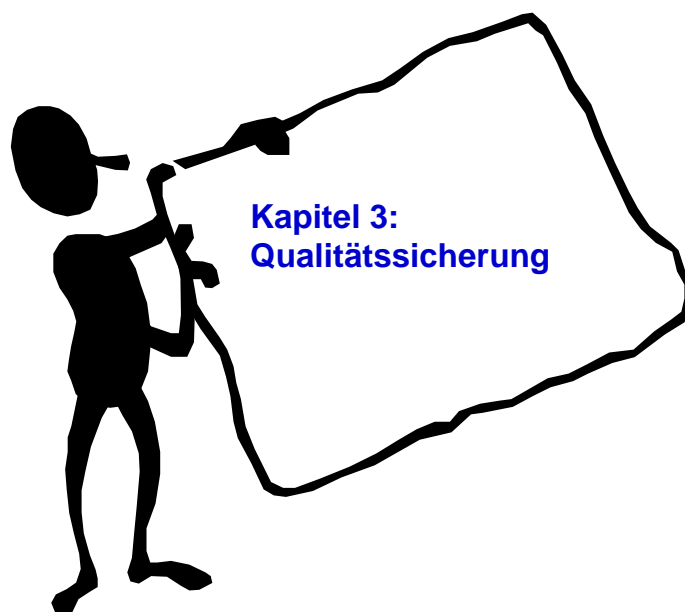
### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM



2. Beschreibungstechniken

- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ...
- 4.5. Vorgehen QM

## 3.1. Motivation

## 3.2. Testen

3.2.1. Komponententest

3.2.2. Systemtest und Akzeptanztest

### 3.2.3. Integrationstest

## 3.3. Dokumentenqualität

## 3.4. Inspektionen

## 3.5. Nutzungstest

## 3.6. Metriken

# 1.3.1. Aktivitäten und Ergebnisse der Entwicklung und Qualitätssicherung

2. Beschreibungstechniken

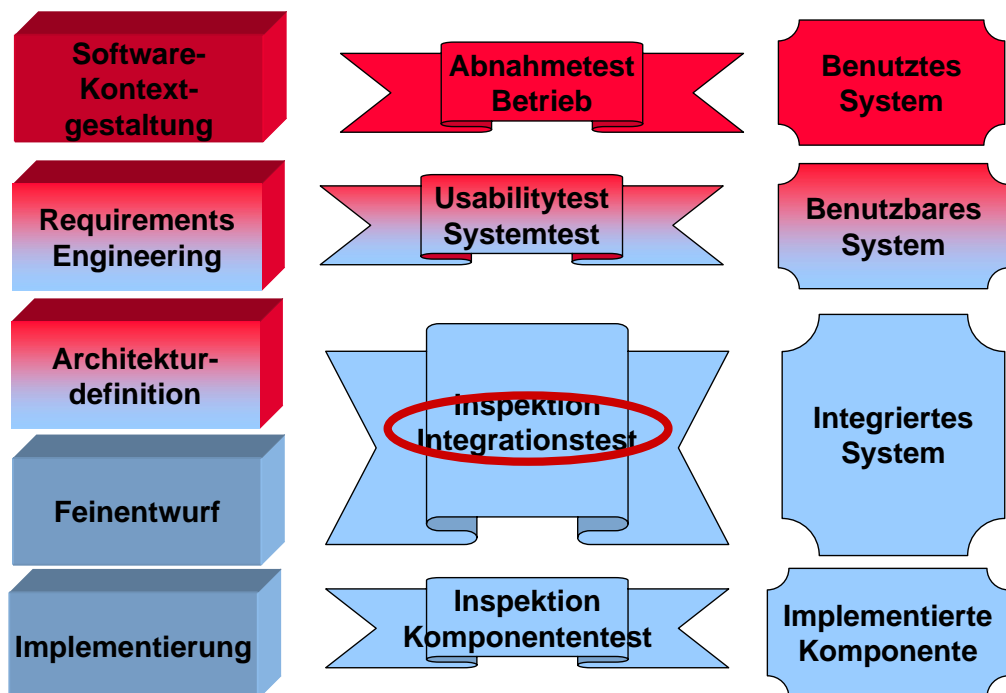
- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ...
- 4.5. Vorgehen QM



### 2. Beschreibungstechniken

- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

### 3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

### 4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ...
- 4.5. Vorgehen QM

- ◆ 3.2.3.1. Überblick
- ◆ 3.2.3.2. Integrationsteststrategien
- ◆ 3.2.3.3. Integrationstest eines objekt-orientierten Entwurfs

### 2. Beschreibungstechniken

- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

### 3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

### 4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ...
- 4.5. Vorgehen QM

- ◆ **Integrationstest** setzt voraus, dass Komponenten schon getestet sind (inkl. Fehlerkorrektur)
- ◆ Testet gegenüber einer technischen Spezifikation (Entwurf)
- ◆ Testet **Zusammenspiel der Komponenten**
- ◆ Benötigt Einheiten, die Schnittstellen zwischen Komponenten beobachten (**Monitore**)
- ◆ Komponenten werden **schrittweise** zusammengesetzt (abhängig von Teststrategie)

## 3.2.3.1. Integrationsprobleme (1)

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM

- ◆ Fehlende, überlappende oder widersprüchliche Funktionen

- ◆ Datenübertragung

- Übergebene Daten syntaktisch falsch
- Übergebene Daten verschieden interpretiert
- Übergebene Daten zum falschen Zeitpunkt vorliegend

- ◆ Falscher Empfänger einer Nachricht

- Falsches Objekt wird an eine Nachricht gebunden

- ◆ Verletzung der Vorbedingungen

Folie 7

## 3.2.3.1. Integrationsprobleme (2)

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM

- ◆ Verletzung der Nachrichtenreihenfolgen/-sequenzen (z.B. bei Client-Server, remote-procedure-call)

- ◆ Gemeinsamer Speicher von Subsystemen

- ◆ Betriebssystem- und Middlewareaufrufe (z.B. Datenbank, ORB)

- ◆ Fehler durch inkorrektes Speichermanagement

- (allocation/deallocation)

Folie 8

### 3.2.3.1. Gründe für Integrationsprobleme

- ◆ Oft Konfigurations- und Versionsinkonsistenzen (Probleme oft schon beim Binden)
- ◆ Falsche Verwendung einer Schnittstelle
- ◆ Synchronisation bei Echtzeit

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

4. Methode

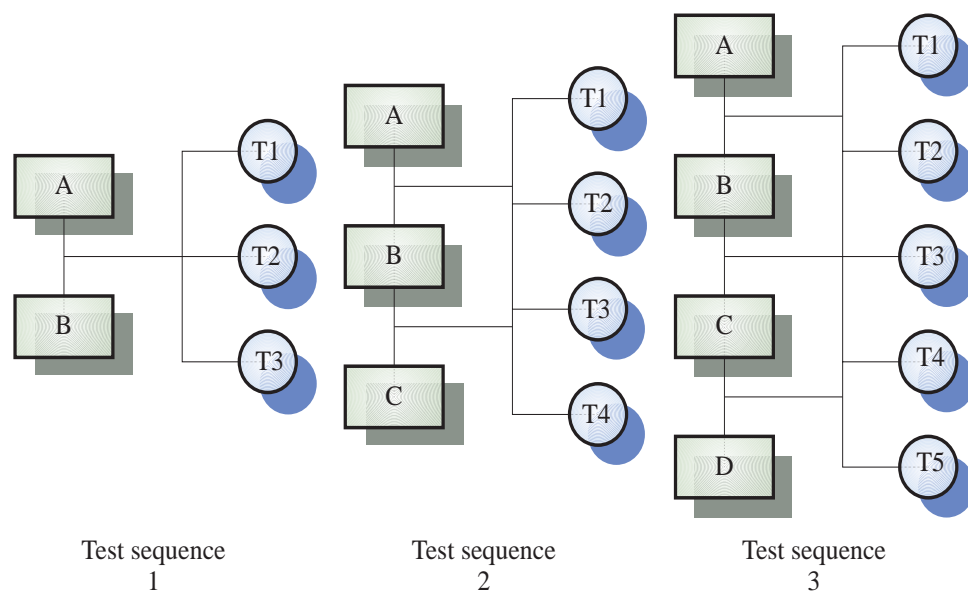
4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM

### 3.2.3.1. Inkrementelle Integration



Wiederholung stellt sicher, dass neue Komponente nicht bisheriges Zusammenspiel stört. Neue Testfälle testen neues Zusammenspiel.

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM

## 3.2.3.2. Integrations-Teststrategien

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM

- ◆ Legen die **Reihenfolge** der Integrationsschritte fest (abhängig von Architektur, Projektplan und Testkonzept)
- ◆ **Top-Down:**
  - Beginn mit Komponente A, die nicht von anderen aufgerufen wird. Ersetze von A benötigte Komponenten mit Stubs und ersetze Stubs inkrementell mit richtigen Komponenten.
- ◆ **Bottom-Up:**
  - Beginn mit Komponenten, die keine anderen aufrufen. Ersetze aufrufende Komponenten durch Testtreiber und ersetze Testtreiber inkrementell mit richtigen Komponenten.
- ◆ Typischerweise beides gemischt nötig!

Folie 11

## 3.2. Testrahmen

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

4. Methode

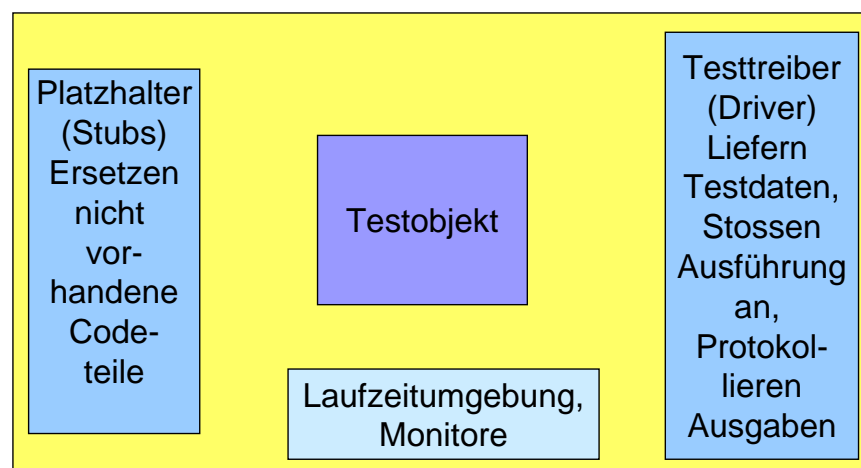
4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM

- ◆ Falls Code nicht allein ablauffähig (z.B. Funktion, Klasse), muss **Testrahmen** bereitgestellt werden



Folie 12

## 3.2.3.2. Top-Down Integrationstest

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

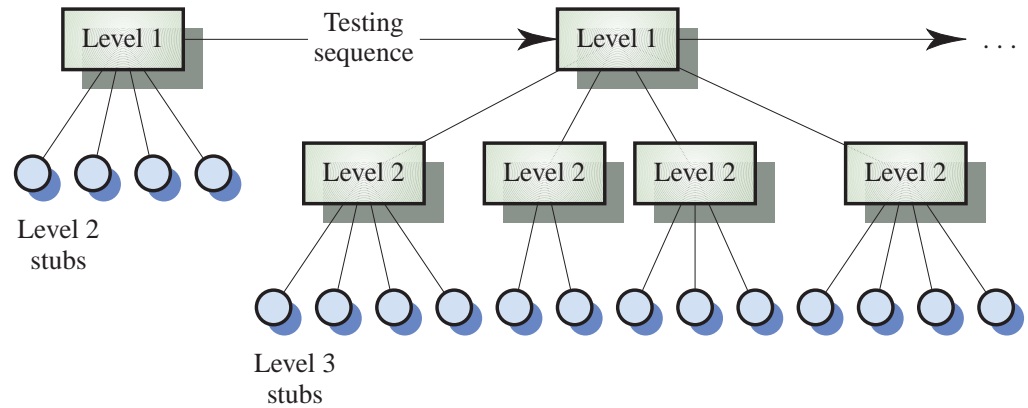
4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM



Benötigt viele Stubs, wenige Testtreiber

Folie 13

© 2004 Institut für Informatik  
Ruprecht-Karls-Universität Heidelberg

Barbara Paech

Vorlesung - Software Engineering - SS 2005

## 3.2.3.2. Bottom-Up Integrationstest

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

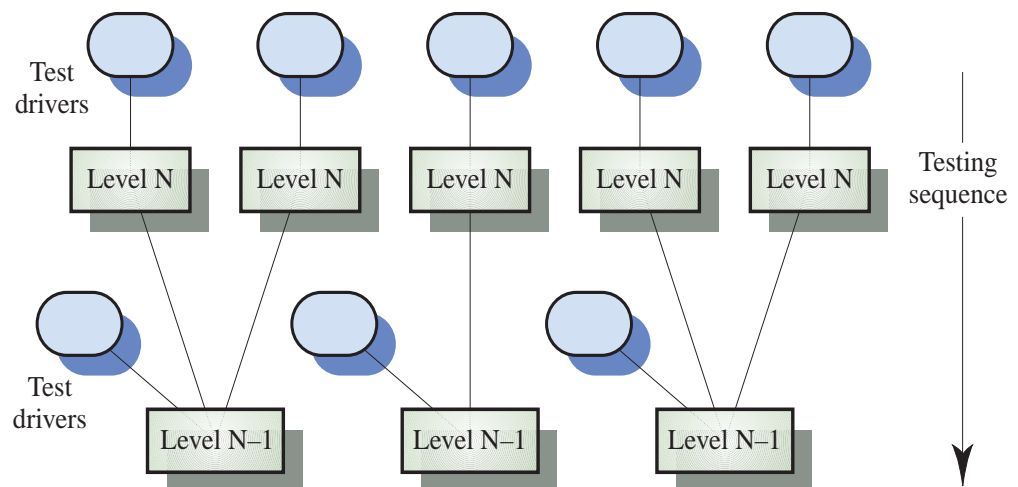
4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM



Benötigt viele Testtreiber, keine Stubs

Folie 14

© 2004 Institut für Informatik  
Ruprecht-Karls-Universität Heidelberg

Barbara Paech

Vorlesung - Software Engineering - SS 2005

## 3.2.3.2. Unterschied zu Komponententest

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM

- ◆ **Komponententest**
  - Aufruf einer Funktion durch Testtreiber mit durch Testmethode bestimmten Eingaben (Testen der inneren Logik der Methode, mglst. vollständige Abdeckung der Funktionsabläufe)
  
- ◆ **Integrationstest**
  - Aufruf einer Funktion aus einer anderen Funktion heraus (Testen des Zusammenspiels, Schnittstelle, mglst. vollständige Abdeckung des spezifischen Zusammenspiels)

## 3.2.3.3. Testen objektorientierter Systeme

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM

- ◆ Testen **einzelner Methoden** in den Klassen (Zusammenspiel der Methoden in einer Klasse)
  
- ◆ Testen **einzelner Klassen** (Einbezug von Vererbung)
  
- ◆ Testen von **Objekt-Clustern**
  - Aufgrund mangelnder Hierarchie ist hier Top-Down oder Bottom-Up nicht strikt durchführbar
  - Besser z.B. szenario-basiert (zielorientiert) oder thread-basiert (ereignisorientiert)

### 3.2.3.3. Abhängigkeiten zwischen Klassen

#### 2. Beschreibungstechniken

##### 2.1. Modellierung

...

##### 2.9. Zusammenfassung

#### 3. Qualitätssicherung

##### 3.1. Einführung

##### 3.2. Testen

##### 3.6. Metriken

#### 4. Methode

##### 4.1. Allgemeines

##### 4.2. Vorgehen RE

...

##### 4.5. Vorgehen QM

- ◆ Vererbung
- ◆ Komposition / Aggregation (Erzeugungsabhängigkeiten)
- ◆ Methodenaufrufe / Attributeverwendung
- ◆ Objekte oder Referenzen auf Objekte als Methodenparameter
- ◆ Gemeinsamer Namensraum
- ◆ Globale Variablen

Folie 17

### 3.2.3.3. Typische Fehler

#### 2. Beschreibungstechniken

##### 2.1. Modellierung

...

##### 2.9. Zusammenfassung

#### 3. Qualitätssicherung

##### 3.1. Einführung

##### 3.2. Testen

##### 3.6. Metriken

#### 4. Methode

##### 4.1. Allgemeines

##### 4.2. Vorgehen RE

...

##### 4.5. Vorgehen QM

- ◆ Methode nicht vorhanden (z.B. wg. Konfigurations- und Versionsproblemen)
- ◆ Falsche Methode/Objekt angesprochen (z.B. bei Bindung zur Laufzeit)
- ◆ Absturz/ Deadlock wegen Konflikten bei der Nutzung gemeinsamer Ressourcen (Prozesse, Benutzungsschnittstelle)

Folie 18

## 3.2. Zusammenfassung Testen

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

- ◆ Semantische Fehler am besten durch Tests zu entdecken
- ◆ Test sehr aufwändig. Deshalb ist eine **durchgängige Strategie über alle Teststufen** notwendig, um mit mglst. geringem Aufwand mglst. viele wichtige Abläufe testen zu können
- ◆ So eine Strategie ist heute vor allem noch erfahrungsbasiert

Folie 19

## 3.6. Metriken

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

- ◆ Erlauben die **Qualität von Dokumenten (insbesondere Code)** automatisch zu überprüfen
- ◆ Geben insbesondere Hinweise auf **Verständlichkeit, Wartbarkeit**
- ◆ Was sind typische Messgrößen im Code?

Folie 20

## 3.6. Software Metriken(1)

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM

<b>Code Länge (LOC)</b>	LOC misst die Programmgröße: <b>Hoher LOC</b> einer Komponente deutet auf Komplexität und Fehleranfälligkeit
<b>Fan in/Fan-out</b>	Fan-in misst die Anzahl von Funktionen, die eine bestimmte Funktion X aufrufen. Fan-out misst die Zahl, der Funktionen, die X aufruft. <b>Hoher Fan-in:</b> X eng gekoppelt mit vielen anderen Funktionen. Änderung hat große Auswirkung. <b>Hoher Fan-out:</b> X hat wahrscheinlich komplexe Kontrolllogik (um die vielen aufgerufenen Funktionen zu synchronisieren).
<b>Verzweigungstiefe</b>	Misst die Tiefe der ineinander geschachtelten IF-Anweisungen. <b>Hohe Tiefe</b> deutet auf schlechte Verständlichkeit und Fehleranfälligkeit.

Folie 21

## 3.6. Software Metriken (2)

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM

<b>Zyklomatische Komplexität V(G)</b>	V(G) misst die Kontrollflusskomplexität eines Programmes anhand des Kontrollflussgraphen G. $V(G) = \text{Anzahl der Kanten} - \text{Anzahl der Knoten} + 2 * \text{Anzahl der vollständig verbundenen Komponenten}$ V(G) sollte nach McCabe kleiner als 10 sein <b>Hoher V(G)</b> deutet auf schlechte Testbarkeit, Verständlichkeit und Wartbarkeit
<b>Länge der Identifikatoren</b>	Misst die durchschnittliche Länge. <b>Hohe Länge</b> deutet auf Verständlichkeit
<b>Fog Index</b>	Misst die durchschnittliche Länge von Wörtern und Sätzen und Dokumenten. <b>Hoher Fog index</b> deutet auf schlechte Verständlichkeit

Folie 22

## 3.6. Objekt-Orientierte Metriken

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

- ◆ Fan-in/Fan-out bezogen auf Methoden
  - Unterscheide Aufruf von anderen Klassen bzw. zu / von der eigenen Klasse
- ◆ Komplexität einer Klasse
  - Addition der Komplexität der Methoden
- ◆ Tiefe des Vererbungsbaumes
- ◆ Anzahl der überschreibenden Methoden einer Superklasse

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

Folie 23

© 2004 Institut für Informatik  
Ruprecht-Karls-Universität Heidelberg

Barbara Paech

Vorlesung - Software Engineering - SS 2005

## 3. Literatur

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

- ◆ A. Spillner, T. Linz: Basiswissen Softwaretest, dpunkt Verlag, 2002
- ◆ R. Binder: Testing object-oriented Systems, Addison-Wesley, 2000
- ◆ IEEE 829-1998, Standard for Software Test Documentation
- ◆ I. Sommerville, Software Engineering, Pearson Studium, 2001

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

Folie 24

© 2004 Institut für Informatik  
Ruprecht-Karls-Universität Heidelberg

Barbara Paech

Vorlesung - Software Engineering - SS 2005

2. Beschreibungstechniken

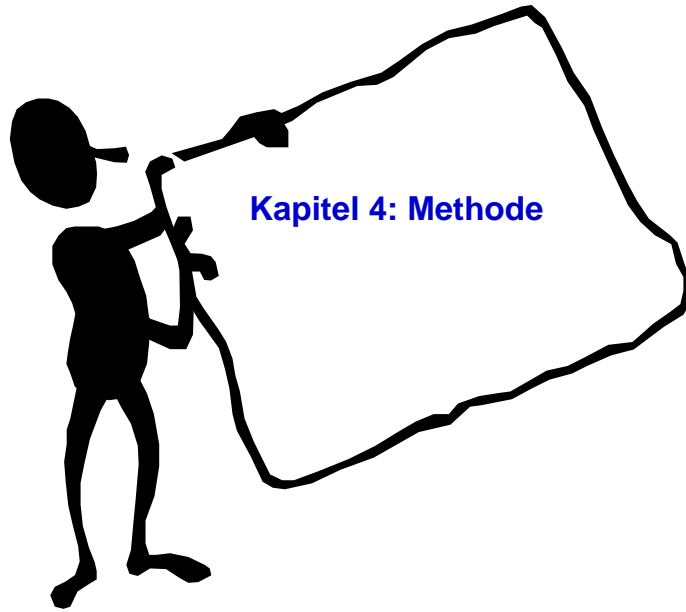
- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ...
- 4.5. Vorgehen QM



2. Beschreibungstechniken

- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

▶ 4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ...
- 4.5. Vorgehen QM

1. Einführung SWE
2. Beschreibungstechniken (Dokumentation)
3. Qualitätssicherung
4. **Methode (Entwicklung)**
5. Projektmanagement
6. Evolution

## Kapitel Vorlesung

2. Beschreibungstechniken

- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ...
- 4.5. Vorgehen QM

- ◆ 4.1. Allgemeines
- ◆ 4.2. Vorgehen Requirements Engineering
- ◆ 4.3. Vorgehen Architektur
- ◆ 4.4. Vorgehen Entwurf
- ◆ **4.5. Vorgehen Qualitätsmanagement**

## 1.3.2. Aktivitäten und Ergebnisse der Entwicklung

2. Beschreibungstechniken

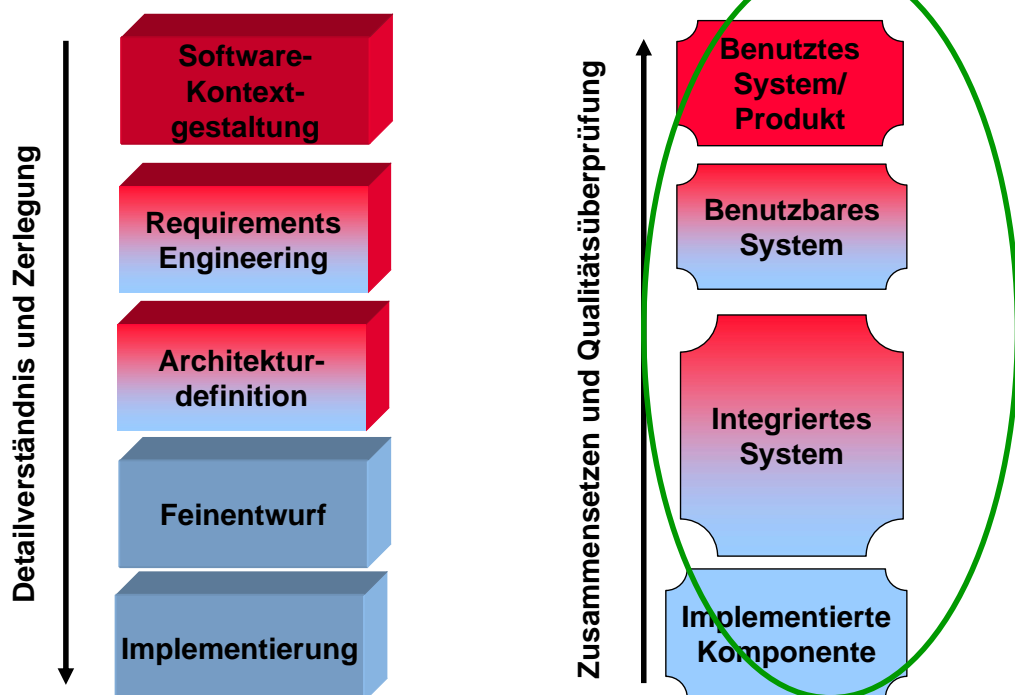
- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ...
- 4.5. Vorgehen QM



## 3.1. Was ist Qualität?

2. Beschreibungstechniken

- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ...
- 4.5. Vorgehen QM

- ◆ **Qualität:** Gesamtheit von Merkmalen und Merkmalswerten einer Einheit bzgl. ihrer Eignung, festgelegte und vorausgesetzte Erfordernisse zu erfüllen [ISO8402]
- ◆ **Achtung:** bezieht sich auf **Produkt und Prozess!**
- ◆ **Achtung:** Erfordernisse (Anforderungen) **ändern** sich über die Zeit!
- ◆ **Achtung:** **Qualität** muss immer mit **Kosten und Zeit** in Bezug gesetzt werden

## 4.5. Typische Qualitätsprobleme

2. Beschreibungstechniken

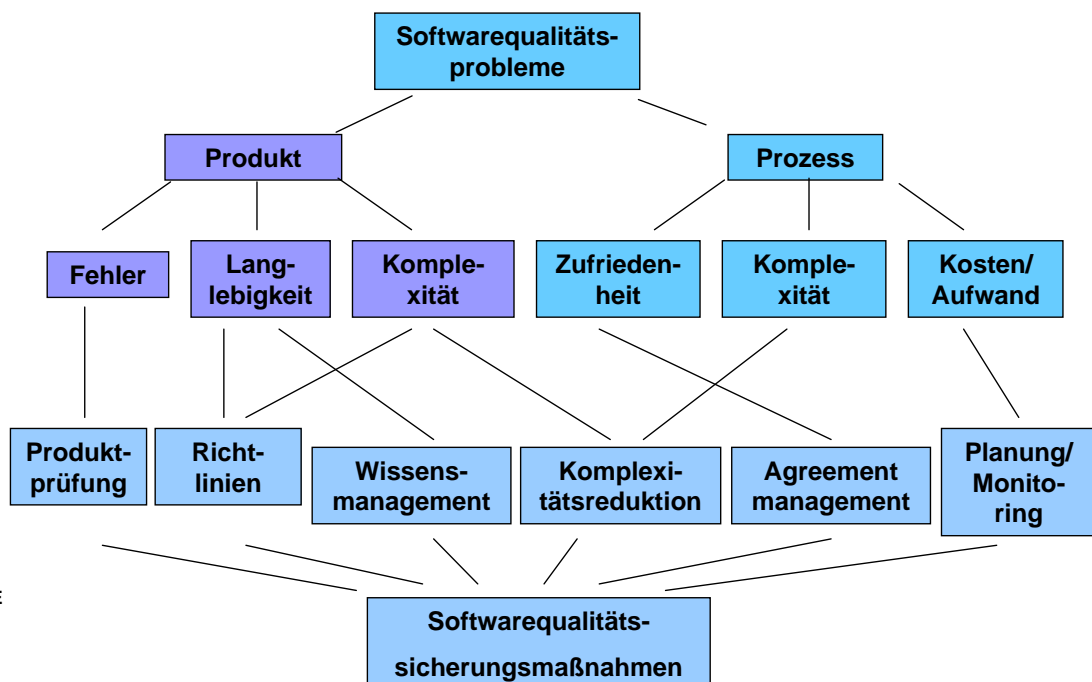
- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ...
- 4.5. Vorgehen QM



## 4.5. Wie kann ich Qualität messen?

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

- ◆ **Einfaches Maß:** bewertet eine unabhängige Variable (z.B. Größe = Lines of Code)
- ◆ **Komplexes Maß:** bewertet Kombination mehrerer unabhängiger Variablen (z.B. Produktivität = LOC/Zeiteinheit)
- ◆ **Maße für Prozess** (z.B. Produktivität, Anzahl der Beteiligten, Anzahl implementierte Anforderungen / Zeiteinheit) und **Produkt** (z.B. Größe, Anzahl Komponenten, gefundene Fehler)

## 4.5. Qualitätsmanagement

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

- ◆ 4.5.1. Aufgaben des Qualitätsmanagements
- ◆ 4.5.2. Qualitätssicherung

## 4.5.1. Aufgaben des Qualitätsmanagements

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

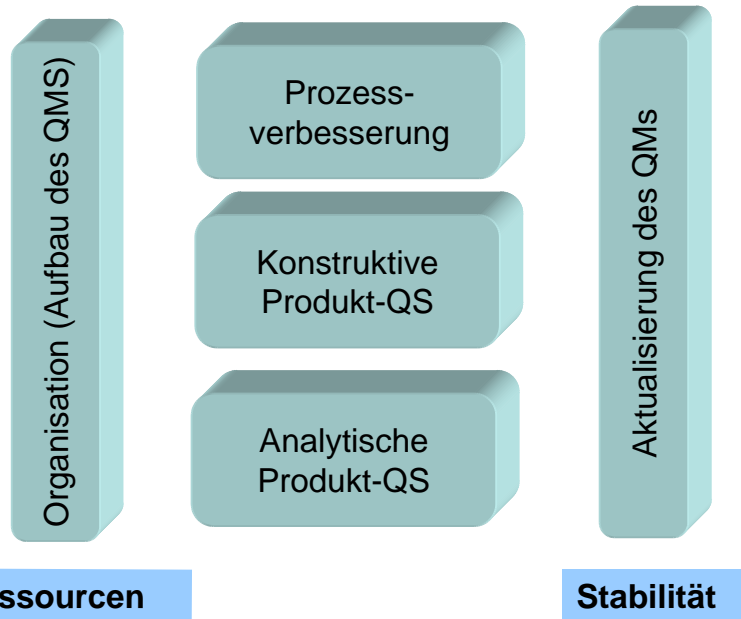
4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM



## 4.5.1. Qualitätsmanagement

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

4. Methode

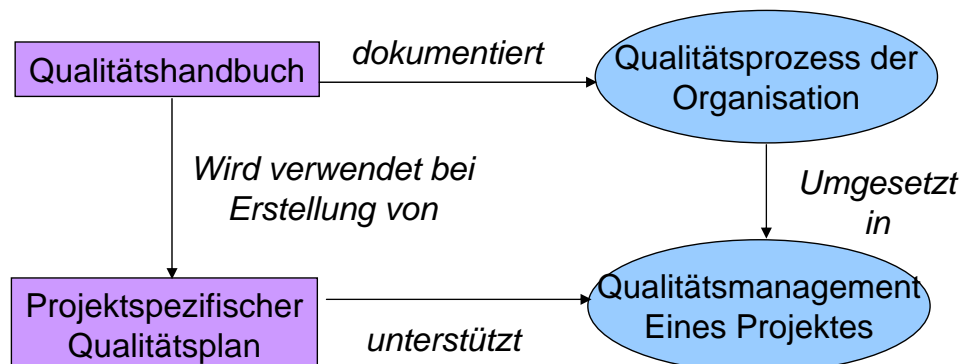
4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM

- ◆ Organisation der Qualitätssicherung
- ◆ ISO 9000 gibt einen Rahmen für QM-Systeme (insbesondere Qualitätshandbuch) vor (Ziel: Zertifizierung von QMS)



## 4.5.1. Bestandteile eines QM-Systems

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

- ◆ Ziele des QM-Systems (z.B. Fehlerdichte in Produkten, Kundenzufriedenheit)
- ◆ Festgelegte **Aufgaben und Zuständigkeiten** (z.B. Q-Manager, Projektleiter)
- ◆ Prozesse zur Planung und Durchführung von **Qualitätssicherungsmaßnahmen** (konstruktiv und analytisch)
- ◆ Prozesse zur **Kontrolle** (Sammeln und Analysieren von Daten aus Qualitätsmessungen)
- ◆ Prozesse zur **Überwachung des QM-Systems** (Audits)

Regelkreis

Folie 35

## 4.5.2. Qualitätssicherung

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

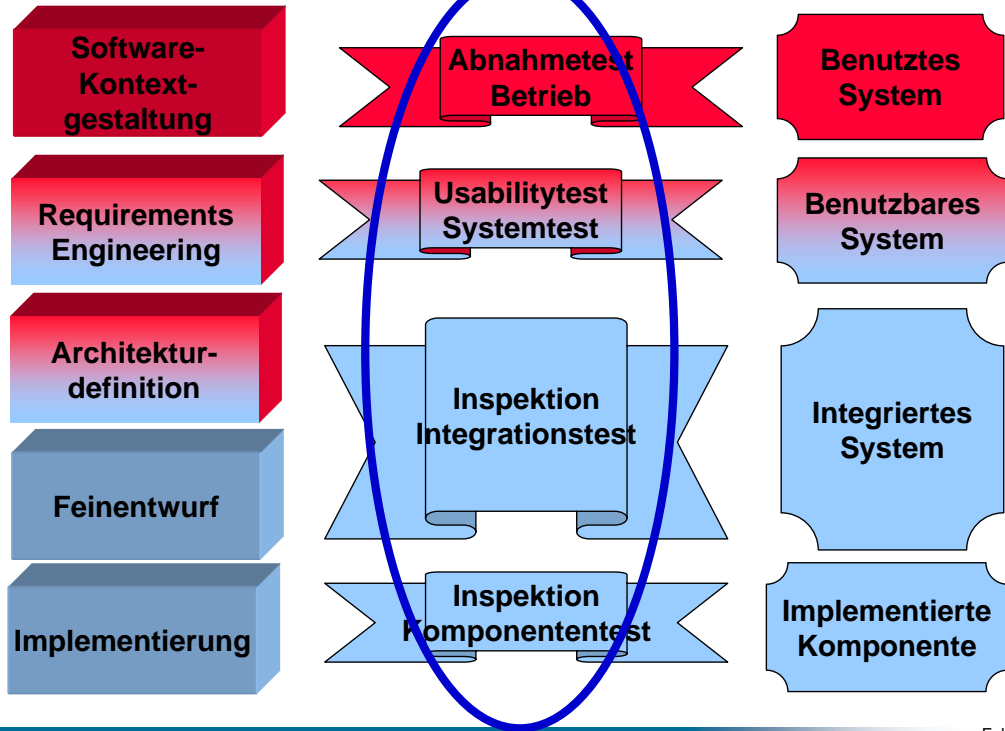
#### 4.5. Vorgehen QM

- ◆ Umfasst konstruktive und analytische Maßnahmen
- ◆ Wird meist nur als analytische Maßnahmen verstanden
- ◆ Im Folgenden nur auf analytische Maßnahmen bezogen

Folie 36

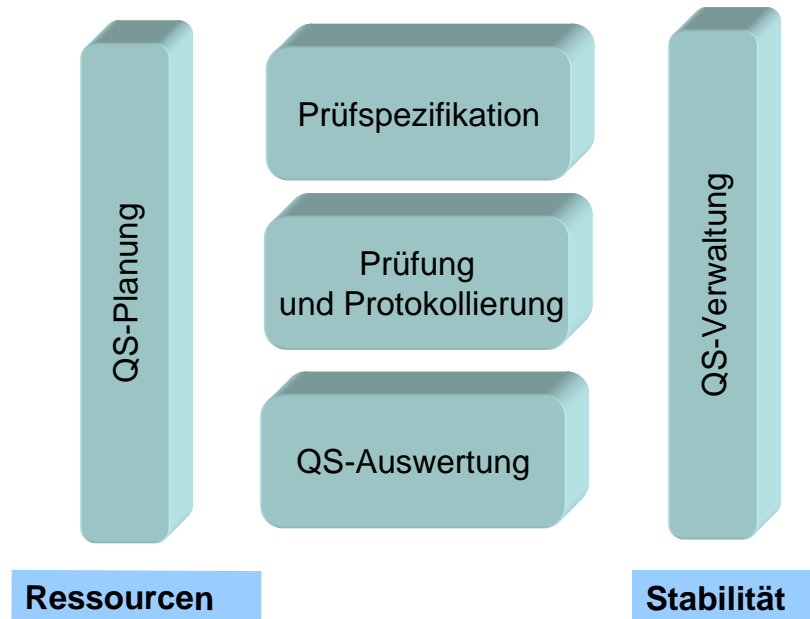
### 1.3.1. Aktivitäten und Ergebnisse der Qualitätssicherung

- 2. Beschreibungstechniken
  - 2.1. Modellierung
  - ...
  - 2.9. Zusammenfassung
- 3. Qualitätssicherung
  - 3.1. Einführung
  - 3.2. Testen
  - 3.6. Metriken
- 4. Methode
  - 4.1. Allgemeines
  - 4.2. Vorgehen RE
  - ...
  - 4.5. Vorgehen QM



### 4.5.2. QS-Aufgaben

- 2. Beschreibungstechniken
  - 2.1. Modellierung
  - ...
  - 2.9. Zusammenfassung
- 3. Qualitätssicherung
  - 3.1. Einführung
  - 3.2. Testen
  - 3.6. Metriken
- 4. Methode
  - 4.1. Allgemeines
  - 4.2. Vorgehen RE
  - ...
  - 4.5. Vorgehen QM



### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM

- ◆ **QS-Manager:** QS-Planung und -auswertung (Was wird wann wie intensiv geprüft?)
- ◆ **QS-Designer:** Prüfmethode und Prüfspezifikation (Wie wird geprüft?)
- ◆ **Prüfer:** Prüfung, Protokollierung (Was sind die Prüfergebnisse?)
- ◆ **Prüfautomatisierer:** Prüfwerkzeuge (Wie werden Werkzeuge eingesetzt?)
- ◆ **QS-Administrator:** QS-Verwaltung (Wie werden Prüfdaten und -ergebnisse verwaltet?)

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM

- ◆ Ziel: Festlegung einer **Prüfstrategie**
- ◆ Prüfstrategie definiert **Prüfmethoden** und **Prüfaufwand** aufbauend auf einer **Kosten/Risiko-**Abschätzung (relativ zu **Qualitätszielen**)
  - **Kosten:** **Direkte Fehlerkosten** (beim Kunden durch Fehlerwirkung entstehende Kosten) + **Indirekte Fehlerkosten** (beim Hersteller durch Kundenunzufriedenheit entstehende Kosten) + **Fehlerkorrekturkosten** (beim Hersteller durch Fehlerkorrektur entstehende Kosten)
  - **Risiken:** Reifegrad des Entwicklungsprozesses, Prüfbarkeit der Software (Dokumentation, Komplexität, Systemumgebung), Mitarbeiterqualifikation

## 4.5.2. Details der Prüfstrategie

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM

### ◆ Pro Dokument und Entwicklungsergebnis

- **Festlegung der Prüfmethode** (Inspektion, Metriken, Black-Box-Test, White-Box-Test...)
- **Festlegung des Abdeckungsgrades**

### ◆ Priorisierung der einzelnen Prüfungen

- **Eintrittswahrscheinlichkeit eines Fehlers** (z.B. häufig vom Kunden genutztes Prüfobjekt, intern kritisches Prüfobjekt, komplexes Prüfobjekt)
- **Fehlerschwere** (Für Kunden: Schaden beim Auftreten bzw. Zufriedenheit; Für Hersteller: Größe des Fehlerbehebungsaufwandes)
- **Priorität der Anforderungen**

Folie 41

## 4.5.2. QS-Planungsdokument

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM

### ◆ QS-Konzept [IEEE 829-1998]

- Einführung
- Qualitätsziele (z.B. zu erreichende Zuverlässigkeit)
- Prüfobjekte und Festlegung, welche Funktionen und Eigenschaften geprüft werden
- Prüfstrategien
- Kriterien für Prüfungsabnahme, Prüfungsabbruch, Prüfungsfortsetzung
- Prüfungsdokumentation
- Prüfumgebung
- Prüfaufgaben und Verantwortliche
- Zeitplanung, Risiken
- Freigabe des QS-Konzepts

Folie 42

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

- ◆ Unterstützen Prüfung, z.B.
  - Ermitteln Metriken automatisch
  - Erlauben räumlich verteilte Inspektionen
  - Unterstützen Testdurchführung
  - Erlauben Simulation für Zielumgebung, Bedienoberflächen, Datenein/ausgabe
  
- ◆ Unterstützen Protokollierung und Auswertung

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

- ◆ Testdurchführung
  - Testdatengeneratoren: Auswahl aus DB oder zufallsgesteuerte Erzeugung (basierend auf Code oder Schnittstellenbeschreibung), auch für Lasttest etc. möglich
  - Protokollgenerator: Erzeugung von Berichten aus Testergebnissen
  - Dynamische Analyse: erweitert Code für Ermitteln von Zusatzinformation (z.B. Logging der Anweisungsausführung)
  - Capture/Replay: zeichnet manuelle Testdurchführung zur Wiederholung auf, kann auch SOLL/IST-Vergleich durchführen

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

- ◆ Stellt durchgeführte Prüfungen für Wartungsprüfungen sowie Weiterentwicklungsprüfungen zur Verfügung
- ◆ Benötigt **QS-Management-Werkzeuge** sowie **Dateivergleichs-Werkzeuge** (um Prüfergebnisse zu vergleichen)
- ◆ Ermöglicht damit **Regressionsprüfungen**

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

- ◆ Qualität eines Produktes entsteht nicht von selbst während der Entwicklung
- ◆ **Qualitätssicherung** ermöglicht Sicherstellung und Verbesserung der Produktqualität
- ◆ **Qualitätsmanagement** ermöglicht Verbesserung der Qualitätssicherung

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

- ◆ A. Spillner, T. Linz: Basiswissen Softwaretest, dpunkt Verlag, 2002
- ◆ IEEE 829-1998, Standard for Software Test Documentation
- ◆ I. Sommerville, Software Engineering, Pearson Studium, 2001

Folie 47

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

...

#### 4.5. Vorgehen QM

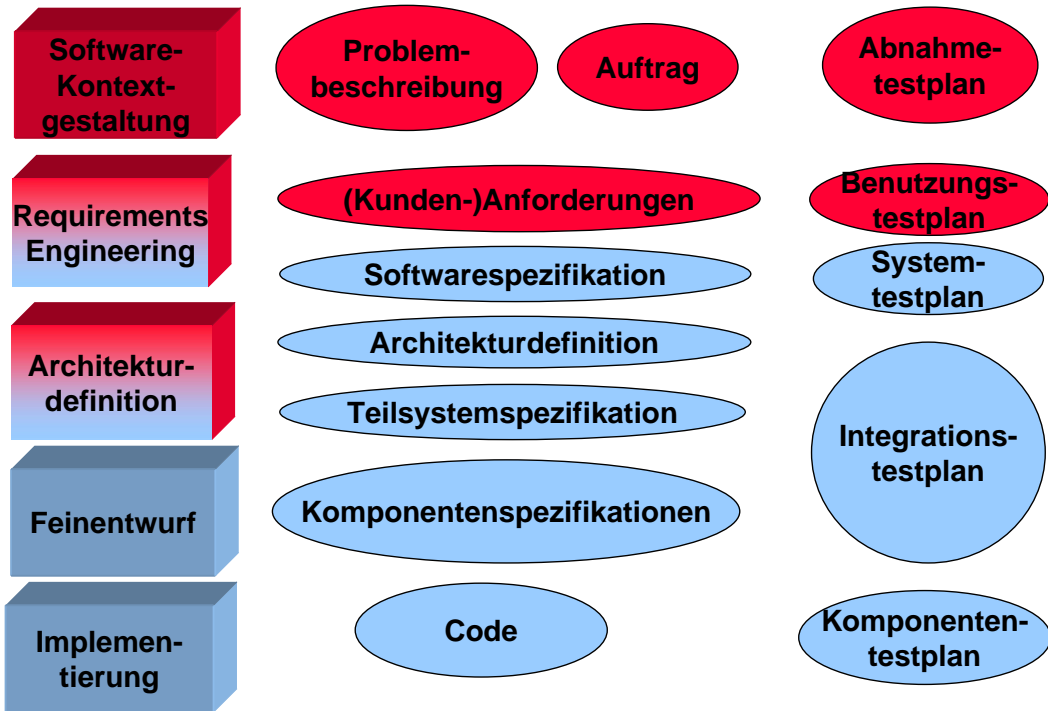


**Kapitel 2:  
Beschreibungstechniken  
[Folien teilweise von  
H. Hussmann und  
B. Rumpe]**

Folie 48

## 1.3.2. Entwicklungsdokumente / Abstraktionsebenen

- 2. Beschreibungstechniken
  - 2.1. Modellierung
  - ...
  - 2.9. Zusammenfassung
- 3. Qualitätssicherung
  - 3.1. Einführung
  - 3.2. Testen
  - 3.6. Metriken
- 4. Methode
  - 4.1. Allgemeines
  - 4.2. Vorgehen RE
  - ...
  - 4.5. Vorgehen QM



## Kapitel 2 Beschreibungstechniken

- 2. Beschreibungstechniken
  - 2.1. Modellierung
  - ...
  - 2.9. Zusammenfassung
- 3. Qualitätssicherung
  - 3.1. Einführung
  - 3.2. Testen
  - 3.6. Metriken
- 4. Methode
  - 4.1. Allgemeines
  - 4.2. Vorgehen RE
  - ...
  - 4.5. Vorgehen QM

- ◆ 2.1. Modellierung
- ◆ 2.2. Objektorientierung
- ◆ 2.3. Klassendiagramme
- ◆ 2.4. Zustandsübergangsdigramme
- ◆ 2.5. Nutzungsbeschreibung
- ◆ 2.6. NFR
- ◆ 2.7. Rationale
- ◆ 2.8. Interaktionsdiagramme
- ◆ **2.9. Zusammenfassung**

## 1.3.5. Beschreibungstechniken

2. Beschreibungstechniken

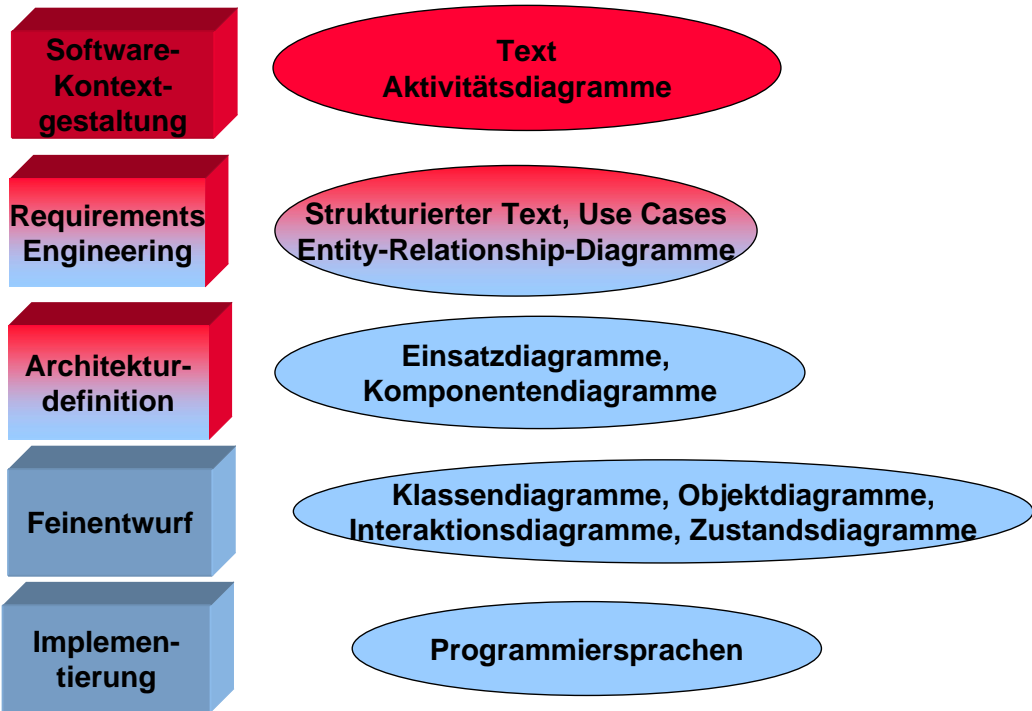
- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ....
- 4.5. Vorgehen QM



## 2.9. Zusammenfassung Beschreibungstechniken

2. Beschreibungstechniken

- 2.1. Modellierung
- ...
- 2.9. Zusammenfassung

3. Qualitätssicherung

- 3.1. Einführung
- 3.2. Testen
- 3.6. Metriken

4. Methode

- 4.1. Allgemeines
- 4.2. Vorgehen RE
- ....
- 4.5. Vorgehen QM

- ◆ Beschreibungstechniken stellen eine Notation zur Verfügung um Ausschnitte der (IST- oder SOLL-) Welt zu beschreiben.
- ◆ Die Notation repräsentiert eine bestimmte Auswahl von **Modellierungskonzepten**.
- ◆ Die Auswahl einer Beschreibungstechnik sollte **systematisch** erfolgen.

## 2.9. Grundlegende Systemkonzepte

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

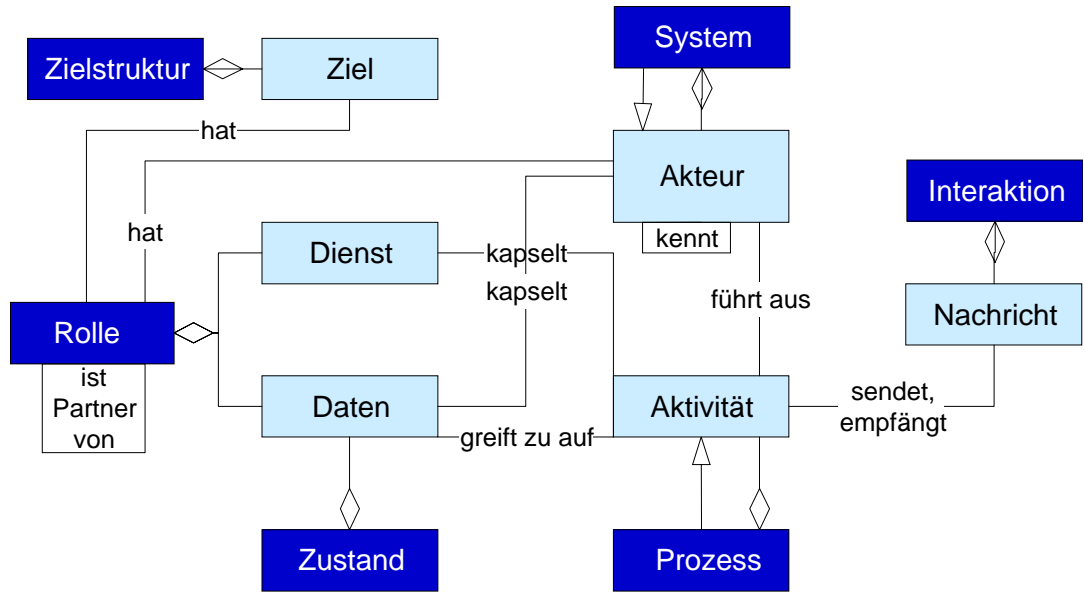
4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM



Folie 53

## 2.9. Fokus der Modellierungstechniken

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM

<b>Daten / Zustand</b>	<b>Struktur Änderungen</b>	
<b>Aktivität / Prozeß</b>	<b>Struktur Folgen</b>	
<b>Dienst</b>	<b>Struktur Verhalten</b>	
<b>Ziel</b>	<b>Struktur</b>	
<b>Rollen</b>	<b>Definition Kontrollzustands- änderungen</b>	
<b>Nachricht / Interaktion</b>	<b>Kommunikations- struktur Folgen</b>	

Folie 54

## 1.3.2. Gestaltungsbereiche der SW-Entwicklung

### 2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

### 3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

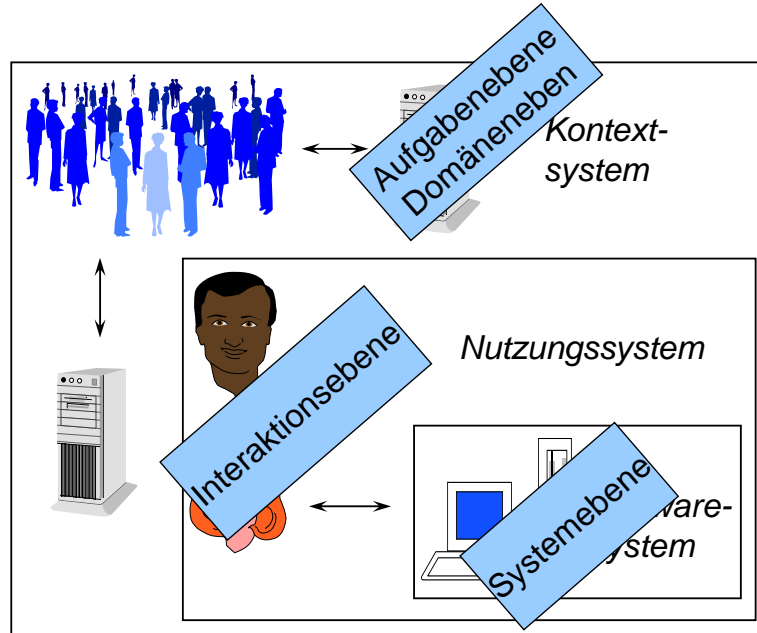
### 4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM



## 2.9. Beispiele für die Systemkonzepte

### 2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

### 3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

### 4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM

	Akteur/Rolle	Dienst	Aktivität	Daten	Nachricht	Ziel
Aufgabenebene						
Interaktions-ebene						
Systemebene						

## 2.9. Aufgabenorientierte Systemmodellierung

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

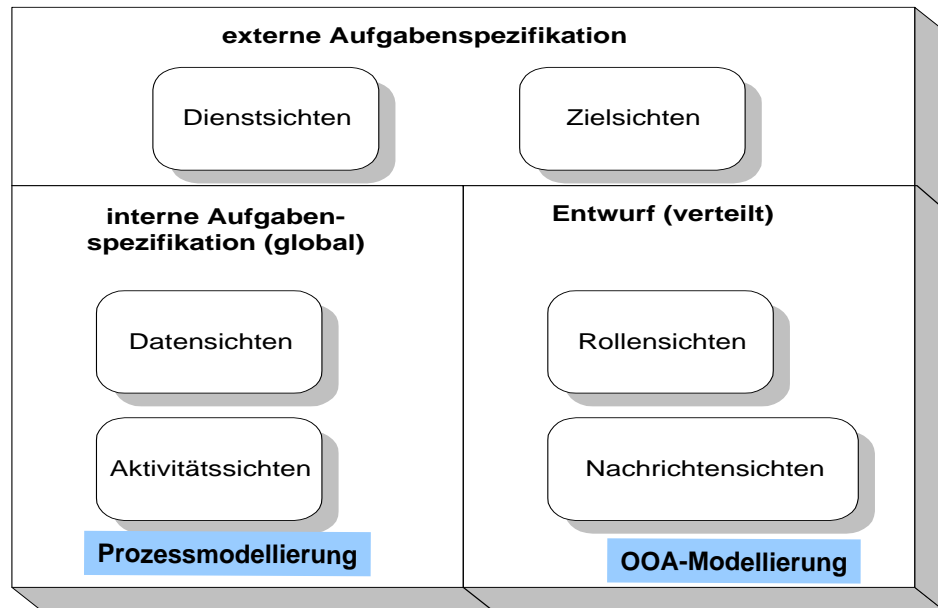
4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM



Folie 57

## 2.9. Modelle in Systementwicklung

2. Beschreibungstechniken

2.1. Modellierung

...

2.9. Zusammenfassung

3. Qualitätssicherung

3.1. Einführung

3.2. Testen

3.6. Metriken

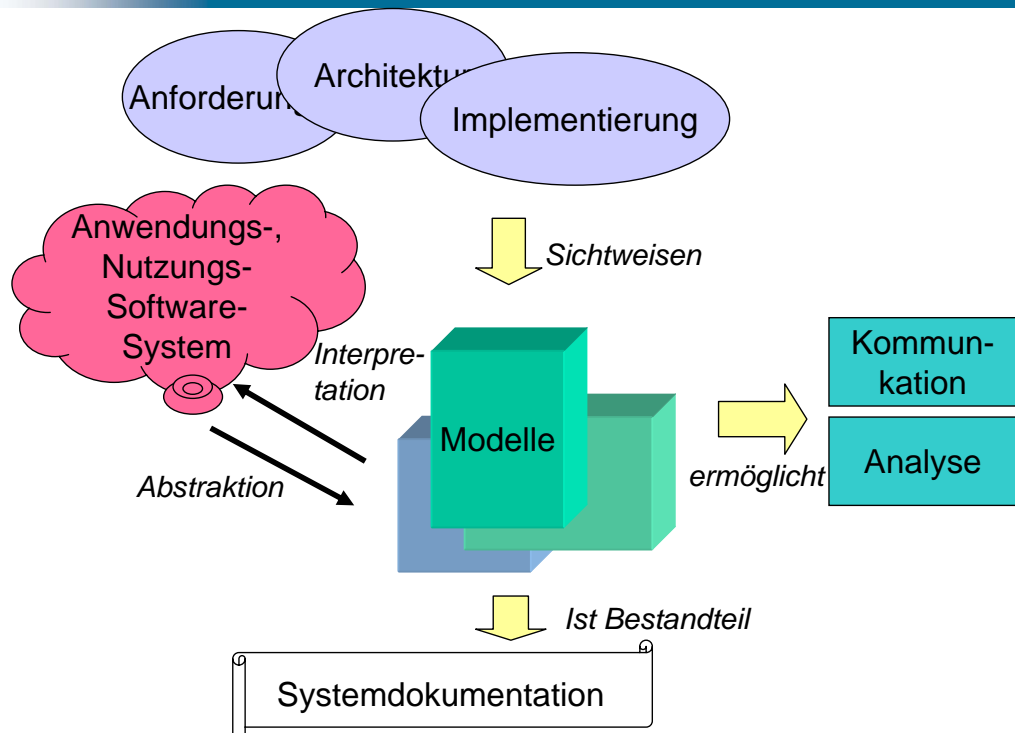
4. Methode

4.1. Allgemeines

4.2. Vorgehen RE

....

4.5. Vorgehen QM



Folie 58



## 2. Beschreibungstechniken

## 2.1. Modellierung

...

## 2.9. Zusammenfassung

## 3. Qualitätssicherung

## 3.1. Einführung

## 3.2. Testen

## 3.6. Metriken

## 4. Methode

## 4.1. Allgemeines

## 4.2. Vorgehen RE

....

## 4.5. Vorgehen QM

## 2.9. Wie hängen in einem Projekt verwendete Modelle zusammen?

- ◆ Vielfältige Beziehungen zwischen Modellen müssen während der SWE verwaltet bzw. berechnet werden können
  - **Strukturelle Abhängigkeit**
    - ist Teil/Erweiterung von
    - ist Ergänzung zu
    - importiert oder nutzt Elemente von
  - **Kausale Abhängigkeit**
    - wird benötigt für/stützt sich auf
    - ist Vorversion von
    - ist Beispiel für
    - ist Prüfergebnis von
  - **Semantische Beziehung**
    - ist Übersetzung von/ist Source von
    - ist Spezifikation von/ist Implementierung von
    - ist Abstraktion von/ist Detaillierung von
    - ist durch Transformation entstanden aus



## 2. Beschreibungstechniken

## 2.1. Modellierung

...

## 2.9. Zusammenfassung

## 3. Qualitätssicherung

## 3.1. Einführung

## 3.2. Testen

## 3.6. Metriken

## 4. Methode

## 4.1. Allgemeines

## 4.2. Vorgehen RE

....

## 4.5. Vorgehen QM

## 2.9. Was hat Einfluss auf die Wahl eines Modells?

- ◆ **Abdeckung** der zu beschreibenden Entscheidung /Systemkonzept
- ◆ **Verständlichkeit** für Beteiligte
- ◆ **Überprüfbarkeit** (z.B. automatisch)
- ◆ **Erstellbarkeit** (z.B. mit Tool, Erfahrung)
- ◆ **Änderbarkeit** des Modells
- ◆ **Kombinierbarkeit** mit anderen Modellen
- ◆ **Verfolgbarkeit** zu anderen Modellen

## 2.9. Wie gehe ich bei der Modellierung vor?

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM

## 1. Klärung des Zwecks des Modells

- **Vorgabe (Spezifikation)**
  - Z.B. Anforderungsspezifikation, Entwurfsmodell, Testspezifikation, Implementierungsmodell
  - Zielt auf Vollständigkeit
- **oder Verständnis/ Kommunikation**
  - Z.B. Problembeschreibung mit Modellen
  - Oft exemplarisch

## 2. Erstellung des Modells

## 3. Ggf. Optimierung des Modells

## 4. Validierung und Verifikation des Modells

- Review
- Simulation

## 2.9. Zusammenfassung

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM

- ◆ Beschreibungstechniken sind notwendig um **komplexe Zusammenhänge** übersichtlich darzustellen
- ◆ Notation muss dem **Zweck** angemessen sein
- ◆ Modelle sind Entwicklungsergebnisse und sind damit (genau wie Code) **systematisch zu entwickeln und weiterzuverarbeiten**

### 2. Beschreibungstechniken

#### 2.1. Modellierung

...

#### 2.9. Zusammenfassung

### 3. Qualitätssicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.6. Metriken

### 4. Methode

#### 4.1. Allgemeines

#### 4.2. Vorgehen RE

....

#### 4.5. Vorgehen QM

- ◆ M. Jeckle, Ch. Rupp, J. Hahn, B. Zengler, S. Queins, „UML 2 glasklar“, Hanser Verlag 2004
- ◆ B. Paech, „Aufgabenorientierte Softwareentwicklung“, Springer Verlag 2000
- ◆ H. Störrle, UML 2 für Studenten, Pearson Studium, 2005