

# 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)

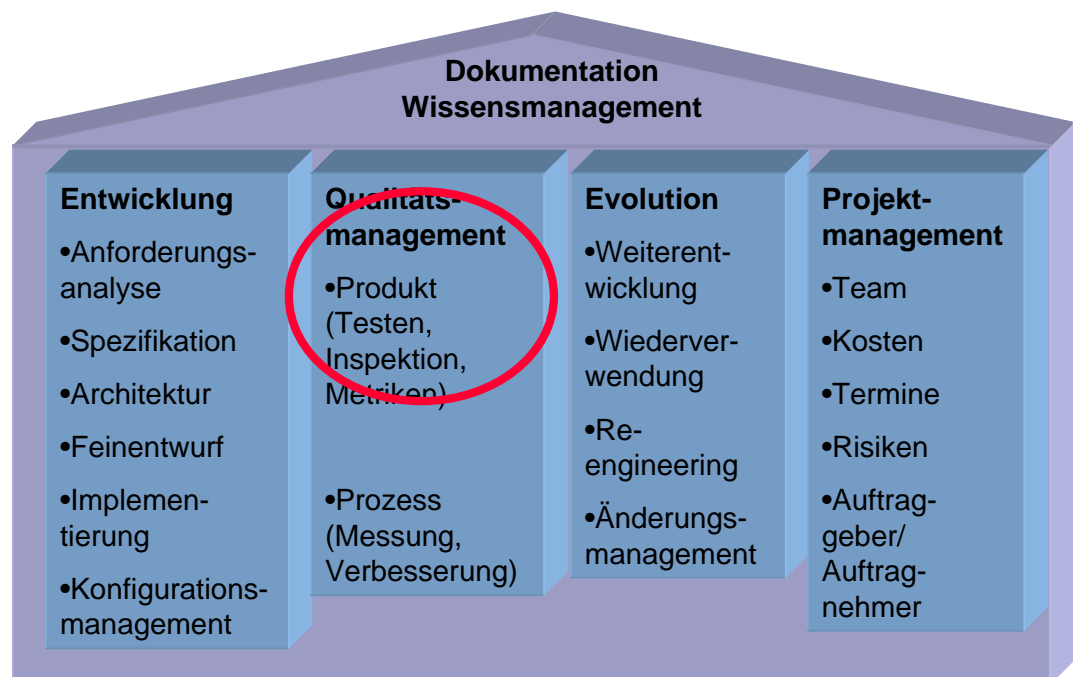


RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG



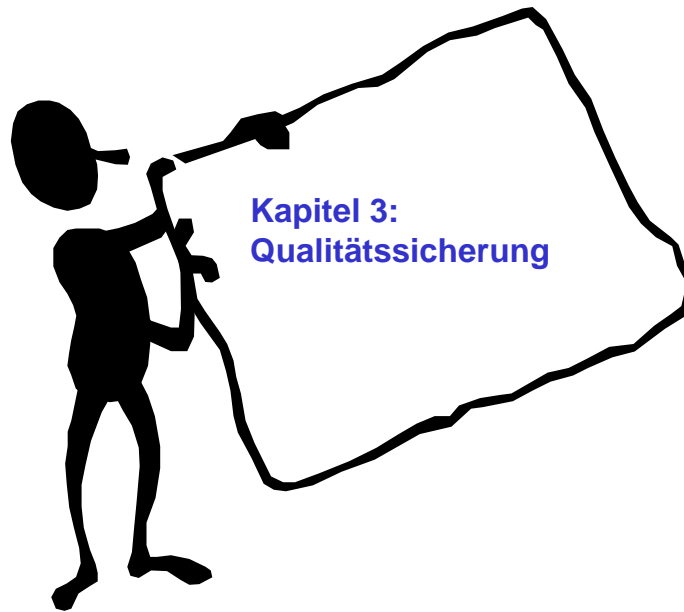
### 1.3.1. Aufgabenbereiche des Engineering

- ▶ 3. Qualitäts-sicherung
  - 3.1. Einführung
  - 3.2. Testen
  - 3.3. Dokumenten-qualität
  - 3.4. Inspektion



▶ 3. Qualitäts-  
sicherung

- 3.1. Einführung
- 3.2. Testen
- 3.3. Dokumenten-  
qualität
- 3.4. Inspektion



Kapitel 3:  
Qualitätssicherung

▶ 3. Qualitäts-  
sicherung

- 3.1. Einführung
- 3.2. Testen
- 3.3. Dokumenten-  
qualität
- 3.4. Inspektion

## 3. Qualitätssicherung

### 3.1. Motivation

### 3.2. Testen

#### 3.2.1. Komponententest

#### 3.2.2. Systemtest und Akzeptanztest

### 3.3. Dokumentenqualität

### 3.4. Inspektionen

### 3. Qualitäts- sicherung

- 3.1. Einführung
- 3.2. Testen
- ▶ 3.3. Dokumenten-  
qualität
- 3.4. Inspektion

### 3.3.1. Motivation

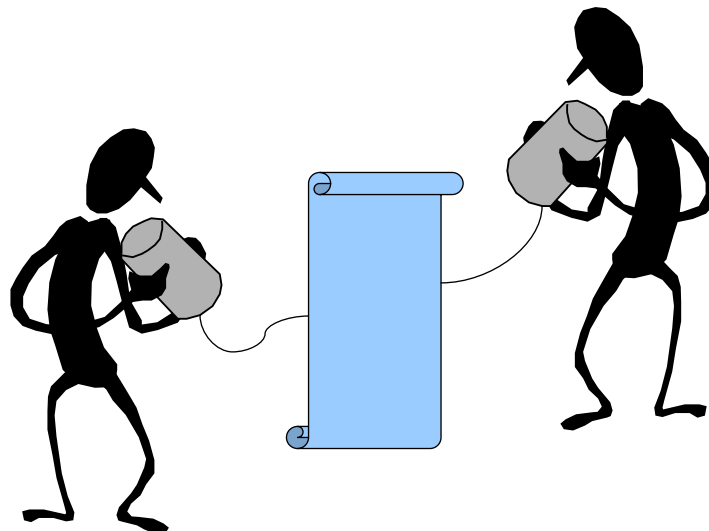
### 3.3.2. Dokumenteninhalt

### 3.3.3. Qualitätsmerkmale

### 3.3.4. Stilregeln

### 3. Qualitäts- sicherung

- 3.1. Einführung
- 3.2. Testen
- ▶ 3.3. Dokumenten-  
qualität
- 3.4. Inspektion



### 3.3.1. Motivation: Wozu sind Dokumente nötig?

3. Qualitäts-  
sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

- ◆ Dokumente sind sehr wichtig als **Kommunikationsmittel**
- ◆ Insbesondere bei
  - **Langlebigkeit** der Inhalte
  - **Rechtlicher Relevanz** der Inhalte
  - **Komplexität** der Inhalte
  - **Zugänglichkeit** für viele Beteiligte
- ◆ Dokumente enthalten im SWE meist Text und Modelle

### 3.3.2. Was gehört in ein technisches Dokument ? (1)

3. Qualitäts-  
sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

1. Einleitung
  - 1.1. Zweck
    - Wer hat das Dokument wie erstellt?
    - Wer soll das Dokument wozu lesen?
    - Für wen ist es wie verbindlich (Geltungsbereich)?
  - 1.2. Zusammenfassung
    - Kerninhalte
  - 1.3. Definitionen und Abkürzungen
    - Definition von Begriffen und Abkürzungen (inkl. Glossar)

## 3.3.2. Was gehört in ein technisches Dokument ? (2)

### 3. Qualitäts- sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

## 1.4. Referenzen, Standards und Vorschriften

Liste der referenzierten Dokumente, insbesondere Standards und Vorschriften

## 1.5. Überblick

Inhalt und Aufbau des Dokuments

## 2. – X. Kerninhalte

## X+1. Zusammenfassung

## X+2. Anhang

## 3.3.2. Beispiel Anforderungsdokument (1)

### 3. Qualitäts- sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

## 2. Beschreibung des Produktkontexts

### 2.1. Zweck des Produktes

zu erreichende Geschäftsziele

### 2.2. Stakeholder

Von dem Produkt Betroffene, daran Beteiligte und Interessierte

### 2.3. Abläufe im Kontext

Geschäftsprozesse, in denen das Produkt benötigt wird

## 3.3.2. Beispiel Anforderungsdokument ? (2)

### 3. Qualitäts- sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

## 3. Systemanforderungen

### 3.1. Systemkurzbeschreibung

### 3.2. Architekturideen

### 3.3. Funktionale Anforderungen

Ablauf und Verhaltensbeschreibung

Datenbeschreibungen

### 3.4. Nicht-funktionale Anforderungen

## 4. Projektanforderungen

### 4.1. Annahmen und Abhängigkeiten (inkl. Risiken)

### 4.2. Abnahme

Folie 11

## 3.3.2. Standards für Dokumente

### 3. Qualitäts- sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

- ◆ Standards geben im Wesentlichen eine **Kapitelgliederung** des Dokuments und eine **kurze Inhaltsbeschreibung** der einzelnen Kapitel vor
- ◆ Z.B. Anforderungen (<http://www.ieee.org>):
  - IEEE Standard 1362-1998 Guide for Information Technology - System Definition - Concept of Open Document (Software-Lastenheft)
  - IEEE Standard 830-1998 Recommended Practice for Software Requirements Specifications (Software-Pflichtenheft)
  - IEEE Standard 1233-1998 Guide for Developing System Requirements Specifications (Systemlasten und -pflichtenheft)
  - speziell: V-Modell (D), Dod-MIL-Std. 498 Department of Defense (USA), European Space Agency (EU), PSS-05 Ministry of Defense (GB), INCOSE, VDI-VDE,...

Gut als Checkliste

Folie 12

## 3.3.2. Umfang und Detailgrad eines Dokuments

### 3. Qualitäts- sicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.3. Dokumenten- qualität

#### 3.4. Inspektion

.... sind abhängig von der Beantwortung der Frage: Wie hoch ist das **Risiko**, wenn die LeserInnen nicht die benötigten Information finden

- ◆ Hoher **Aufwand zur Klärung von Fragen** für die LeserInnen
  - Z.B. wenn AutorInnen zeitlich, räumlich schwer zu erreichen
- ◆ Hohes **Risiko bei Auftreten von Fehlern** aufgrund von fehlender oder missverständlicher Information
  - hoher **Qualitätsanspruch** an die Software?
- ◆ Hohe **Änderungswahrscheinlichkeit**
  - Information wichtig, aber noch nicht fest

## 3.3.3. Merkmale von guten Dokumenten

### 3. Qualitäts- sicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.3. Dokumenten- qualität

#### 3.4. Inspektion

- ◆ **Verständlich für alle!**
  - **Eindeutig, vollständig, konsistent**
- ◆ **Verständlich für die Nutzer des Dokuments z.B. Anforderungen**
  - **Kunde:**
    - **korrekt:** Ist „Auftrag“ richtig wiedergegeben?
  - **RE-Ingenieur:** interne Abstimmung bei Änderungen, Stand des RE-Prozesses
    - **Gewichtet, nachvollziehbar, änderbar**
  - **Entwickler:**
    - **Umsetzbar:** Klare Vorgabe?
  - **Tester:**
    - **verifizierbar:** Testfälle ableitbar?

### 3.3.3. Merkmale von guten Anforderungen

3. Qualitäts-  
sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

- ◆ Nach IEEE Std. 830-1998
  - **eindeutig** (d.h. nur eine mögliche Interpretation)
  - **vollständig** (d.h. alle nicht-funktionalen Anforderungen, alle Systemreaktionen - auch auf ungültige Eingaben, explizite Kennzeichnungen von offenen Punkten)
  - **konsistent** (keine Widersprüche, konsistente Terminologie)
  - **korrekt** (d.h. nur "richtige" Anforderungen)
  - **verifizierbar** (d.h. durch ein Verfahren kann Erfüllung geprüft werden)
  - **gewichtet** bzgl. Wichtigkeit und Stabilität
  - **änderbar** (d.h. klare Struktur, gute Überblickstabellen, keine Redundanz)
  - **nachvollziehbar** (d.h. Quelle der Anforderungen festhalten und eindeutige Identifikatoren)

### 3.3.4. Stilratgeber für die Formulierung von Dokumenten

3. Qualitäts-  
sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

- ◆ Ziel: Dokumente sind **leichter zu lesen** und somit leichter zu verstehen
- ◆ Unser Stilratgeber behandelt die **häufigsten Probleme**, projektspezifisch sind Ergänzungen sinnvoll
- ◆ Regeln sollten zu einem firmenspezifischen Stilratgeber zusammengefasst werden

## 3.3.4. Stilratgeber (1)

### 3. Qualitäts- sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

- ◆ **Kurze Sätze** und **kurze Absätze** (max. ca. 7 Sätze), da das menschliche Kurzzeitgedächtnis begrenzt ist
- ◆ **Nur eine Aussage pro Satz** formulieren: 'und' vermeiden, Verschachtelte logische Zusammenhänge vermeiden
  - Wenn X oder Y und in diesem Fall Z gegeben ist, jedoch nicht wenn ...)
  - Pseudocode oder Entscheidungstabellen verwenden
- ◆ **Konsistente Terminologie:** Jargon vermeiden, Abkürzungen sparsam verwenden, Wortwiederholungen sind erwünscht!

## 3.3.4. Stilratgeber (2)

### 3. Qualitäts- sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

- ◆ **Offene Punkte kennzeichnen:** „TBD“
- ◆ **Generalität vermeiden:** klare Referenzen, Wörter wie „alle“ vermeiden
- ◆ **Verbindlichkeit klar formulieren:** „Muss, kann, soll“ etc. mit Bedacht verwenden
- ◆ **Aktiv formulieren:** „Das System ....“

## 3.3.4. Beispiele für Anforderungen (1)

### 3. Qualitäts- sicherung

- 3.1. Einführung
- 3.2. Testen
- ▶ 3.3. Dokumenten-  
qualität
- 3.4. Inspektion

#### Negativ

Wird ein Flugzeug identifiziert, das feindlich ist und eine unbekannte Mission hat, oder den geschützten Flugraum in weniger als 5 Minuten erreichen kann, dann soll ein Alarm ausgelöst werden.

#### Positiv?

## 3.3.4. Beispiele für Anforderungen (2)

### 3. Qualitäts- sicherung

- 3.1. Einführung
- 3.2. Testen
- ▶ 3.3. Dokumenten-  
qualität
- 3.4. Inspektion

#### Negativ

Beispiel Herd: "Mit der R-Taste wird die gewählte Funktion abgebrochen."

#### Positiv?

## 3.3.4. Beispiele für Anforderungen (3)

### 3. Qualitäts- sicherung

- 3.1. Einführung
- 3.2. Testen
- ▶ 3.3. Dokumenten-  
qualität
- 3.4. Inspektion

**Negativ**

Das System ermöglicht die Eingabe von Punkten. Es werden dann die Gesamtbewertungen ermittelt.

**Positiv?**

## 3.3.4. Stilratgeber für UC

- ◆ Siehe Word-Dokument

### 3. Qualitäts- sicherung

- 3.1. Einführung
- 3.2. Testen
- ▶ 3.3. Dokumenten-  
qualität
- 3.4. Inspektion

## 3.3. Zusammenfassung Dokumentenqualität

### 3. Qualitäts- sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

- ◆ Qualität der Dokumente ist wichtig für die Dokumentennutzer
- ◆ Es sollten klare Vorgaben für Gliederung und Inhalte existieren
- ◆ Auf die Formulierung kommt es an!

## 3.2. Testen

### 3. Qualitäts- sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

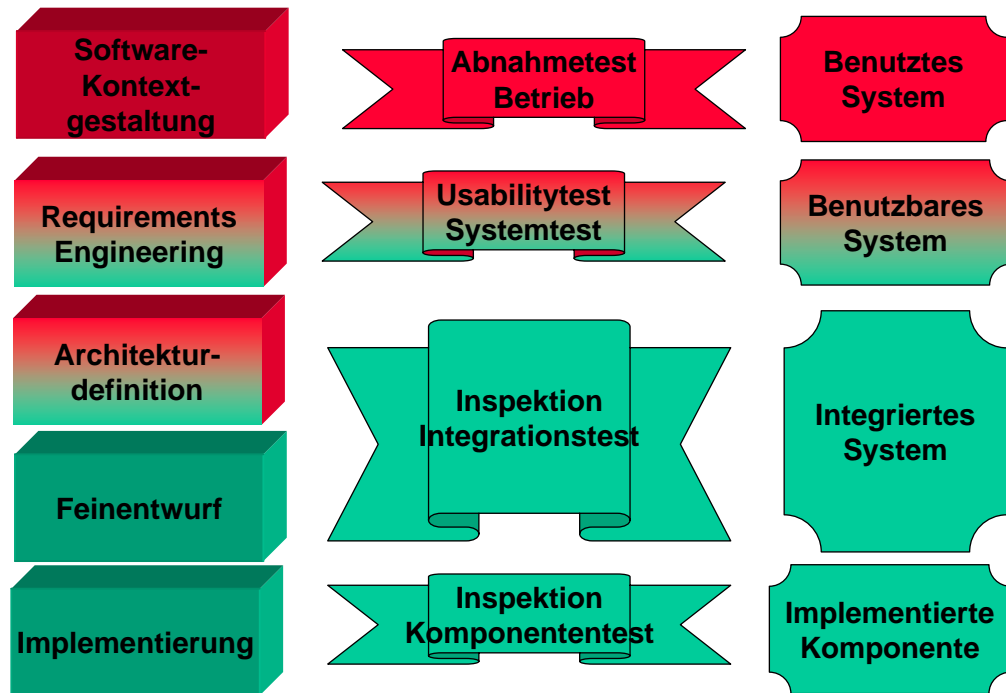
3.4. Inspektion

- 3.2.1. Komponententest
- 3.2.2. Systemtest und Akzeptanztest

## 1.3.1. Aktivitäten und Ergebnisse der Entwicklung und Qualitätssicherung

### 3. Qualitäts-sicherung

- 3.1. Einführung
- 3.2. Testen
- 3.3. Dokumenten-qualität
- 3.4. Inspektion



## 3.2.2. Systemtest

### 3. Qualitäts-sicherung

- 3.1. Einführung
- 3.2. Testen
- 3.3. Dokumenten-qualität
- 3.4. Inspektion

- ◆ Testet ob **Kunden-Anforderungen richtig umgesetzt** wurden
- ◆ Testumgebung sollte **Produktiv-Umgebung mglst. Nahe** kommen (also keine Stubs und Testtreiber)
- ◆ Produktiv-Umgebung oft selbst nicht geeignet, wegen Schadensrisiko und mangelnder Kontrolle
- ◆ Kann **einzelne Funktionen**, aber auch **Funktionssequenzen** (für Geschäftsprozesse) testen
- ◆ Sollten mit Test **nicht-funktionaler Leistungsmerkmale** verbunden werden

## 3.2. Testfallspezifikation in REQuest

### 3. Qualitäts- sicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.3. Dokumenten- qualität

#### 3.4. Inspektion

- ◆ Einleitung
- ◆ Endekriterien für Gesamttest bzw. Testobjekt
- ◆ Ressourcen
- ◆ Testfallbeschreibung
  - Typ (Komponenten, Integration, System...)
  - Ziel (Testobjekt, Problem, Testmethode)
  - Vorbedingung (Testumgebung einrichten)
  - Schritte (Eingabe, erwartete Ausgabe, erwartete Ausnahmen, Testabbruch)
  - Nachbedingung (Testumgebung abbauen)
  - Ressourcen (Werkzeuge)
  - Testendekriterium für Testfall

## 3.2.2. Abdeckung für Systemtest

### 3. Qualitäts- sicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.3. Dokumenten- qualität

#### 3.4. Inspektion

- ◆ Abdeckung der Systemfunktionen und Abdeckung der Use Cases
- ◆ Abdeckung der Systemfunktionen
  - White-Box und/oder Blackbox
  - White-Box lässt sich auch auf Beschreibung anwenden
- ◆ Abdeckung der Use Cases
  - White-Box angewendet auf Beschreibung

3. Qualitäts-  
sicherung

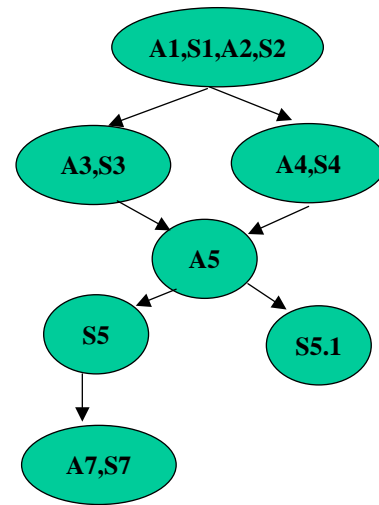
3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

- ◆ A1,S1,A2,S2
- ◆ If A3 then S3  
If A4 then S4
- ◆ A5,S5 [Ausnahme:  
S5.1]
- ◆ A7,S7



Typische Abdeckung:  
alle/ein gültiger Pfad, alle ungültigen Pfade

3. Qualitäts-  
sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

- ◆ **Funktionalität**
  - Angemessenheit
  - Sicherheit
  - Genauigkeit der Berechnung
  - Interoperabilität
  - Konformanz zu Standards
- ◆ **Zuverlässigkeit**
  - Reife
  - Fehlertoleranz
  - Wiederherstellbarkeit
- ◆ **Benutzbarkeit**
  - Verständlichkeit
  - Erlernbarkeit
  - Bedienbarkeit
- ◆ **Effizienz**
  - Zeitverhalten
  - Verbrauchsverhalten
- ◆ **Änderbarkeit**
  - Analysierbarkeit
  - Modifizierbarkeit
  - Stabilität
  - Prüfbarkeit
- ◆ **Übertragbarkeit**
  - Anpassbarkeit
  - Installierbarkeit
  - Konformanz zu Standards
  - Austauschbarkeit

## 3.2.2. Testen von Produktqualität (1)

### 3. Qualitäts- sicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.3. Dokumenten- qualität

#### 3.4. Inspektion

- ◆ **Funktionalität**
  - Angemessenheit
  - Sicherheit
  - Genauigkeit der Berechnung
  - Interoperabilität => **Kompatibilitätstest** (Datenaustausch mit vorhandenen Systemen)
  - Konformanz zu Standards
- ◆ **Zuverlässigkeit**
  - Reife => **Stresstest** (bei Überlastung), **Stabilität/Zuverlässigkeitstest** (Dauerbetrieb)
  - Fehlertoleranz => **Robustheitstest** (gegenüber Fehlbedienung, HW-Ausfall...)
  - Wiederherstellbarkeit
- ◆ **Benutzbarkeit => Usability Test**

## 3.2.2. Testen von Produktqualität (2)

### 3. Qualitäts- sicherung

#### 3.1. Einführung

#### 3.2. Testen

#### 3.3. Dokumenten- qualität

#### 3.4. Inspektion

- ◆ **Effizienz => Lasttest** (# Anwender, Transaktionen) und **Volumen/Massentest** (# Daten)
- ◆ **Änderbarkeit => Wartbarkeitstest, Dokumentationstest**
- ◆ **Übertragbarkeit**
  - Anpassbarkeit
  - Installierbarkeit => **Konfigurationstest** (z.B. versch. Sprachen, BS)
  - Konformanz zu Standards
  - Austauschbarkeit

### 3. Qualitäts- sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

- ◆ Setzt Systemtest voraus (inkl. Fehlerkorrektur)
- ◆ Testet ob **Kunden-Anforderungen richtig umgesetzt** wurden. Insbesondere auch Benutzerakzeptanztests
- ◆ Testet in **Produktiv-Umgebung**
- ◆ Bei vielen verschiedenen Produktiv-Umgebungen systematischer **Feldtest** (Beta-Versionen an Auswahl von Kunden)

### 3. Qualitäts- sicherung

3.1. Einführung

3.2. Testen

3.3. Dokumenten-  
qualität

3.4. Inspektion

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