Software-Qualitätsmanagement

Vorlesung im Modul 10-202-2319 Software-Management

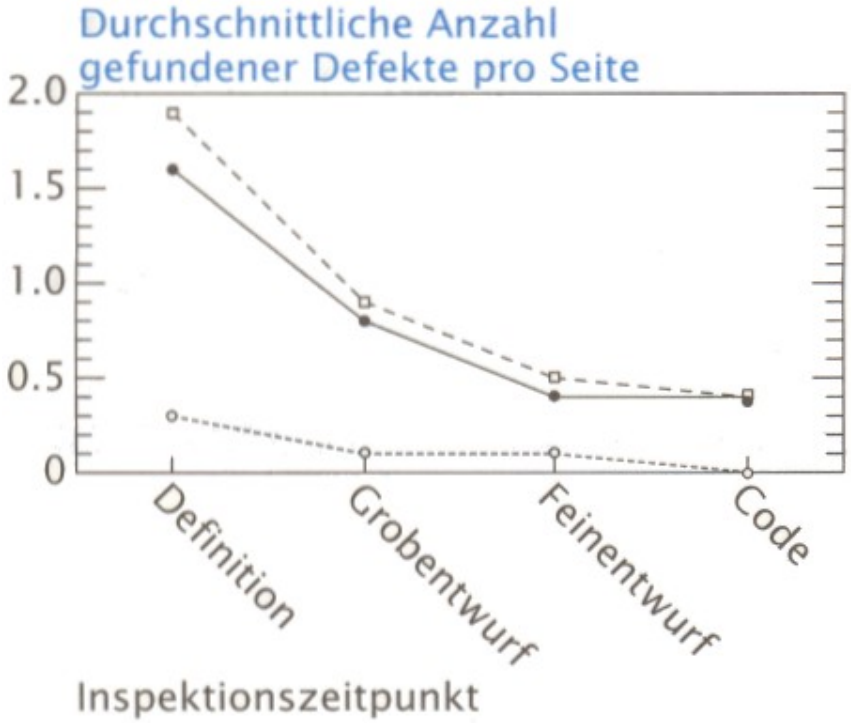
Sommersemester 2012

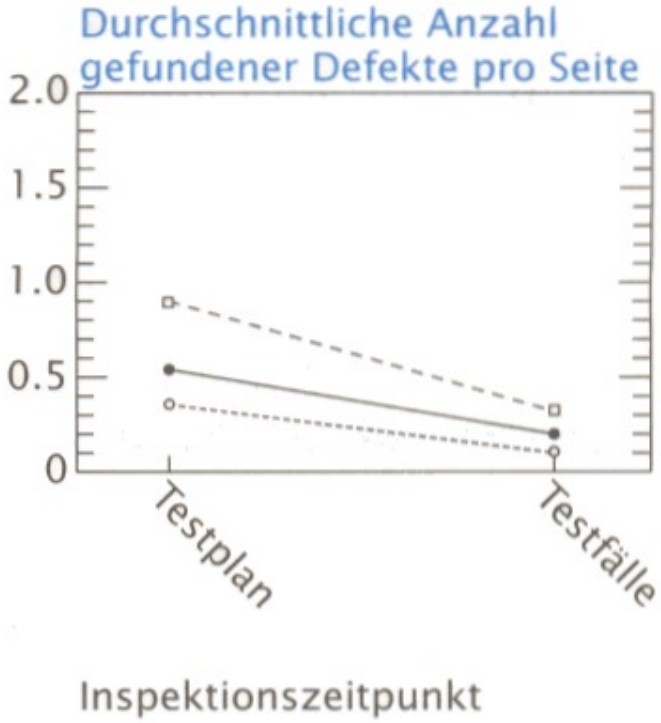
Prof. Dr. Hans-Gert Gräbe

http://bis.informatik.uni-leipzig.de/HansGertGraebe

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Empirische Ergebnisse





Legende: schwere Defekte, leichte Defekte, alle Defekte

Testplan = Testplandokument
Testfälle = dokumentierte Testfälle

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Faustregeln:

- 50 bis 75% aller Entwurfsfehler können durch Inspektionen gefunden werden.
- Code-Inspektionen sind ein sehr kosteneffektiver Weg, um Defekte aufzudecken.

- Die Investitionseffizienz (Verhältnis der ersparten Ingenieurstunden zu den Kosten) ist viel besser als für viele andere Investitionen.
- Weitere Vorteile:
  - Kürzere Entwicklungszeit
  - Geringeres Risiko
  - Die Trainings- und Einführungskosten fallen pro Team nur einmal an.
- Frühe Inspektionen sind effizienter als spätere

Review

Manuelle Prüfmethode, mit welcher Stärken und Schwächen eines schriftlichen Dokuments in Bezug auf Referenzunterlagen nach individueller Vorbereitung der Gutachter in einer Teamsitzung identifiziert werden, um diese durch den Autor beheben zu lassen.

- Weniger stark formalisiert als Inspektion.
- **Ziel**: Feststellung von Mängeln, Fehlern, Inkonsistenzen, Unvollständigkeiten sowie Verstößen gegen Vorgaben und Richtlinien
- **Dokumente**: Prüfobjekt, Referenzdokumente
  - Dokumente: max. 50 Seiten, 5 Gutachter, 10 Seiten/Std.
  - Code: max. 20 Seiten, 3 Gutachter, 5 Seiten/Std.
- **Rollen**: Moderator, Autor, Protokollführer, 2-5 Gutachter

- **Vorgehen**: Beantragung, Eingangsprüfung, optionale Einführungs-Sitzung, individuelle Vorbereitung, Review-Sitzung, Überarbeitung, Bestätigung
  - Ablauf ist im Wesentlichen wie bei einer Inspektion
  - Eingangsprüfung wird vom Manager u. U. zusammen mit dem Moderator durchgeführt
- **Aufwand**: 15% (Code) – 20% (Dokumente) des Aufwands für die Erstellung des Prüfobjekts
- **Nutzen**: 60 – 70% der Fehler werden gefunden.
  - Reduktion der Fehlerkosten in der SE um 75% und mehr
  - Nettoeinsparungen in der Entwicklung um ca. 20%, in der Wartung um ca. 30%.

Durchsprache (Walkthrough)

Manuelle informale Prüfmethode, um in einer Teamsitzung Fehler, Defekte, Unklarheiten und Probleme in schriftlichen Dokumenten zu identifizieren und durch den Autor beheben zu lassen.

- Noch weniger stark formalisiert als Review.
- **Ziele**:
  - Identifikation von Defekten
  - Vermittlung von Inhalten (Ausbildung der Mitarbeiter)
- **Dokumente**: Prüfobjekt oder Teilprodukte
- **Rollen**: Autor, Gutachter
- **Vorgehen**: Gruppensitzung unter Leitung des Autors nach (optionaler) individueller Vorbereitung
- **Aufwand**: relativ gering
- **Nutzen**: Auf Grund des informellen Charakters schwer messbar

Vorteile

- geringer Aufwand
- auch für kleine Entwicklungsteams geeignet
- sinnvoll für „unkritische“ Dokumente bzw. Dokumente in frühen Entwicklungsphasen
- Durch Einbeziehung von Kunden/Nutzern als Gutachter können Unvollständigkeiten und Missverständnisse aufgedeckt werden
- gut geeignet, um das Wissen über ein Dokument auf eine breite Basis zu stellen.

Nachteile

- Es werden wenig konkrete Defekte identifiziert.
- Autor kann die Durchsprache dominieren
- Überarbeitung des Prüfobjekts liegt im Ermessen des Autors und wird nicht nachgeprüft.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Andere manuelle Prüfmethoden

- **Stellungnahme**: Der Autor bittet ein oder mehrere Kollegen um Kommentare zu einem Prüfobjekt. Der Autor gibt den Kollegen Kopien des Prüfobjekts und erhält diese mit Kommentaren zurück.
- **Round-Robin-*Review***: Ziel ist es, Argumente für die Güte des Prüflings zu sammeln. Jeder Gutachter versucht in der *Review*-Sitzung die anderen davon zu überzeugen, dass die Qualität des Prüfobjekts akzeptabel ist.
- **Peer-*Review***: Gutachter werden in einem Raum „eingeschlossen“, untersuchen ein oder mehrere Prüfobjekte, und liefern Gutachten dazu. Das *Review*-Team bestimmt selbst die Aufgabenverteilung und die Vorgehensweise.

Zusammenfassung

Manuelle Prüfmethoden dienen dazu, Produkt- oder Prozesseigenschaften zu überprüfen, welche durch automatische Werkzeuge nicht oder nur unzureichend festgestellt werden können.

Die Effektivität hängt vor allem von folgenden Punkten ab

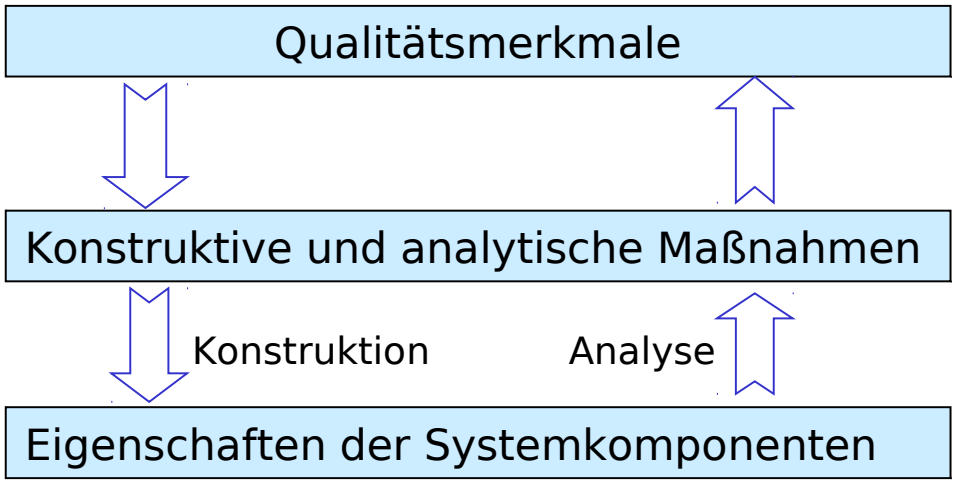
- Gutachter konzentrieren sich auf einzelne Aspekte
- Gutachter bereiten sich individuell und schriftlich vor
- In einer Team-Sitzung werden moderiert die Ergebnisse zusammengetragen und weiter analysiert. Lösungen werden dabei nicht diskutiert.
- Der Prüfaufwand ist in der Projektplanung berücksichtigt.

Der Aufwand (Review oder Inspektion) liegt bei 15-20% des Erstellungsaufwands für das entsprechende Produkt oder Dokument.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

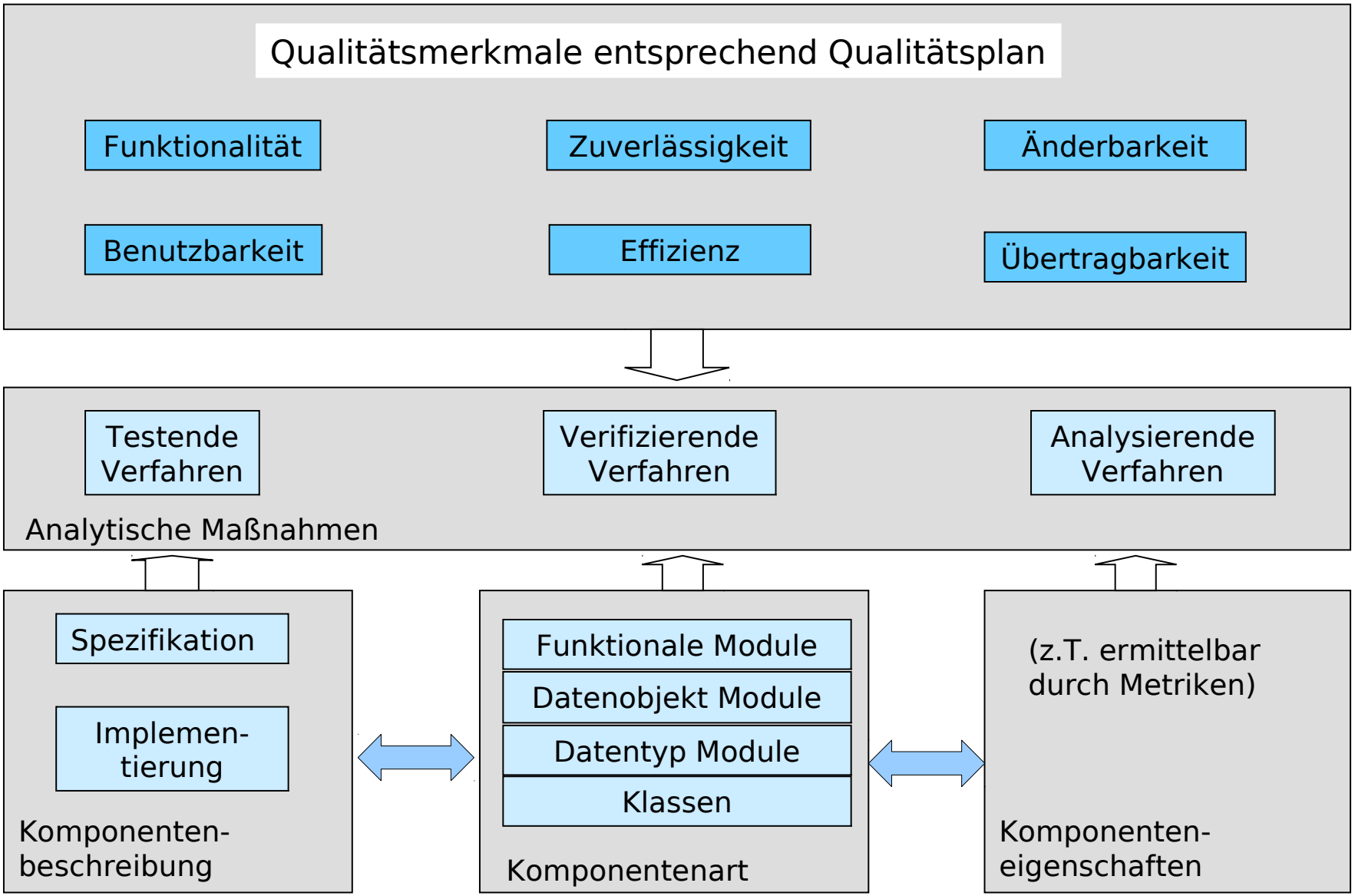
Software-Produktqualität ist abhängig von:

- Qualität der Systemkomponenten
- Qualität der Beziehungen zwischen den Komponenten



- Konstruktive Maßnahmen siehe VL „Software-Technik“
- analytische Maßnahmen beziehen sich im Wesentlichen auf Funktionalität, Zuverlässigkeit und evtl. Änderbarkeit

1. Einführung



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Fehler als zentraler Begriff

Konstruktives Ziel: Entwicklung fehlerfreier Software-Komponenten

Analytisches Ziel: Nachweis der Fehlerfreiheit von Software-Komponenten

Als **Fehler** wird

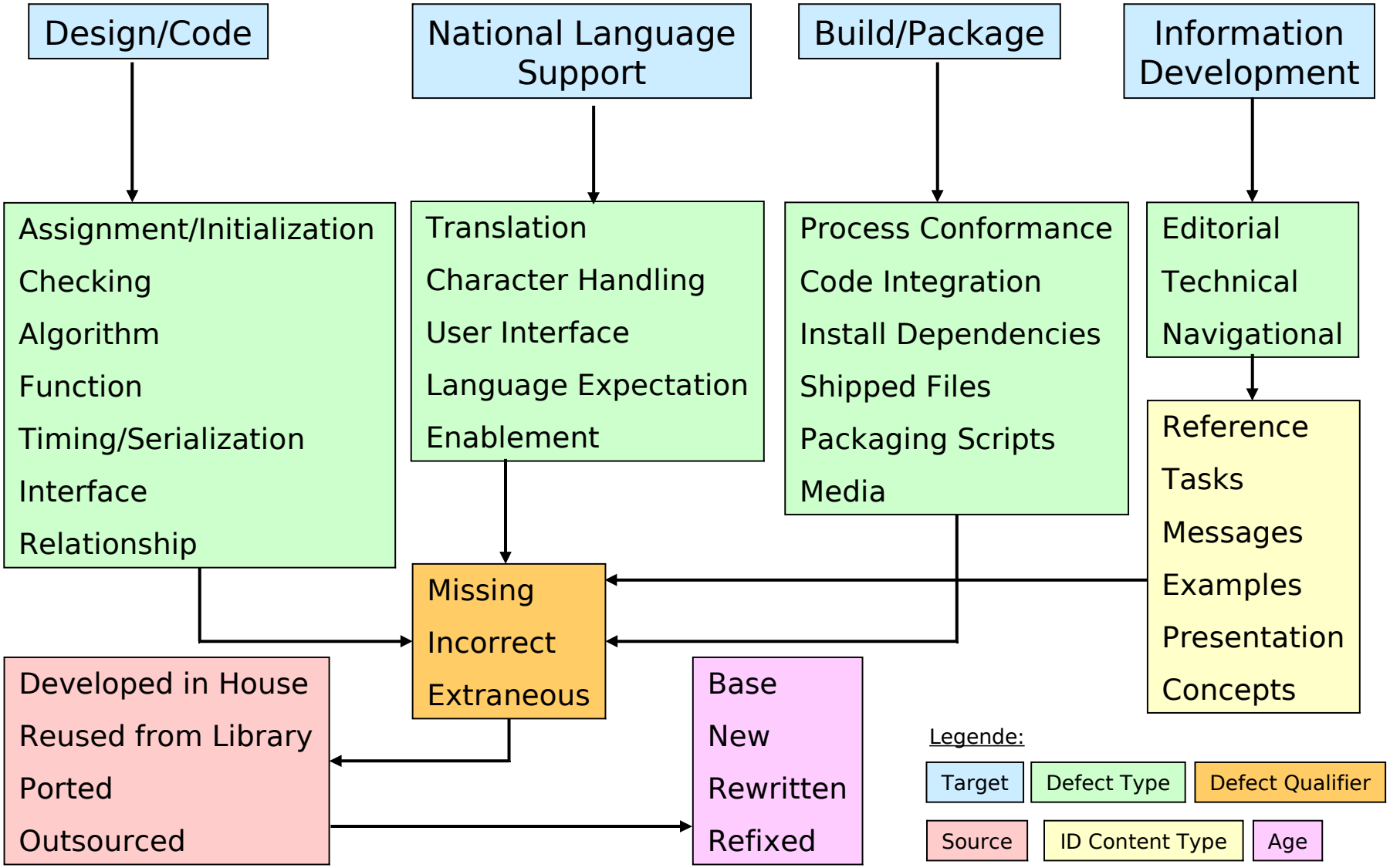
- jede Abweichung der tatsächlichen Ausprägung eines Qualitätsmerkmals von der vorgesehenen Soll-Ausprägung,
- jede Inkonsistenz zwischen der Spezifikation und der Implementierung und
- jedes strukturelle Merkmal des Programmtexts, das ein fehlerhaftes Verhalten des Programms verursacht,

bezeichnet.

UNIVERSITÄT LEIPZIG

Institut für Informatik
Betriebliche Informationssysteme

2. Was sind Fehler?



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Analyseverfahren

Die Art und die Eigenschaften der Systemkomponenten bestimmen die Auswahl geeigneter analytischer Maßnahmen:

- Funktionale Module
- Datenobjekt-Module
- Datentyp-Module
- Klassen

Unterschied der eingesetzten analytischen Maßnahmen insbesondere zwischen Komponenten mit komplexen Kontrollstrukturen und Komponenten mit komplexen Datenstrukturen.

3. Analyseverfahren

Testende Verfahren haben das Ziel, Fehler zu erkennen

- Dynamische Testverfahren
- Statische Verfahren

Verifizierende Verfahren sollen die Korrektheit einer Komponente beweisen

- Verifikation
- Symbolische Ausführung

Analysierende Verfahren sollen Eigenschaften von Komponenten darstellen oder vermessen

- Analyse der Bindungsart
- Metriken
- Grafiken und Tabellen
- Anomalienanalyse

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Testende Verfahren - Prinzipieller Zugang

- Das tatsächliche Verhalten wird **stichprobenartig** an Hand einer Menge von **Testfällen** untersucht und die Ergebnisse mit den erwarteten Ergebnissen (Spezifikation, Normen) verglichen und dokumentiert.
- Testfälle werden entsprechend den Testzielen speziell ausgewählt.
- Einsatz um
  - Programmfehler aufzufinden (Bugs)
  - Wiederauftreten von Fehlern zu vermeiden (Regressionstests)
- Fehlerfreiheit ist plausibel, aber nicht garantiert.
- Abzugrenzen von
  - **Verifikation** als strengem Korrektheitsbeweis
  - **Ausprobieren** als einer Entwicklungsmethode (trial and error)

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Begriff des Programms

- Programm = schrittweise Transformation einer Menge von Eingabedaten in eine Menge von Ausgabedaten nach einem vorgegebenen Algorithmus
- Black-Box-Betrachtung: f: X $\rightarrow$ Y
  - Spezifikation, funktionale Korrektheit
- Transformation = Abarbeiten einzelner Programmschritte, in denen die Daten entsprechend den angegebenen Instruktionen verändert werden.
  - zustandsorientierte Betrachtung: Datenfluss
  - übergangsorientierte Betrachtung: Kontrollfluss
- Programmstatus = Zustand der Gesamtheit der durch das Programm manipulierten Daten
  - Anweisungen und Deklarationen
  - Variablenbegriff als Wertcontainer
    - Sichtbarkeit und Lebensdauer
    - Compilezeit und Laufzeit

Statische und dynamische Testverfahren

Dynamische Testverfahren

- übersetztes und ausführbares Programm wird mit konkreten Eingabewerten ausgeführt
- evtl. Instrumentierung des Programms
- Test in realer Laufzeitumgebung
- Stichprobenverfahren (Testfälle)
- Ziele: Finden von Fehlern (debugging), Finden und Optimieren laufzeitkritischer Bereiche (profiling)
- Korrektheit kann so nicht bewiesen werden
- Klassifikation nach Herkunft der Testfälle

Statische Testverfahren

- Analyse des Quellcodes, evtl. testfallorientiertes Durchgehen
- typische Verfahren: manuelle Prüfmethoden

Glossar

Prüfling, **Testling** oder **Testobjekt**

- das zu testende Programm oder die Komponente (Prüfling als allgemeiner Begriff)

Testverfahren

- grundlegendes Verfahren, mit dem einzelne Eigenschaften eines Testlings durch eine geeignete Anzahl von Testfällen untersucht werden

Testfall

- Satz von Testdaten, der die vollständige Ausführung eines zu testenden Programms bewirkt

Testdatum

- Eingabewert, der einen Eingabeparameter des Testobjekts instanziiert

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Testtreiber

- Testrahmen, mit dem eine nicht extern zugängliche Funktion interaktiv aufgerufen werden kann.
- Können durch Testwerkzeuge automatisch generiert werden.

Instrumentierung

- Der Quellcode des Testlings wird für die Analyse der Testfälle mit zusätzlichem Protokollcode versehen oder dieser aktiviert.
- Instrumentierter Testling wird übersetzt. Protokoll enthält Informationen über das Laufzeitverhalten des Testlings während des Abarbeitens der einzelnen Testfälle.

Überdeckungsgrad

- Maß für den Grad der Vollständigkeit eines Tests bezogen auf ein bestimmtes Testverfahren

Regressionstest

- automatische werkzeuggestützte Neuausführung bereits durchlaufener Tests nach Änderungen am Testling.