



Qualitätssicherung von Software

Prof. Dr. Holger Schlingloff

Humboldt-Universität zu Berlin
und
Fraunhofer FIRST

Hinweise

- Übung heute 13:15 Uhr **findet statt!**
- Frage: Test harness
 - harness: Harnisch, Ausrüstung, Geschirr
 - test harness: Gesamtmenge der Testwerkzeuge
- Folien kommen heute nachmittag ins Web

versprochener Nachtrag über Entscheidungstabellen

- Algorithmus nach Myers

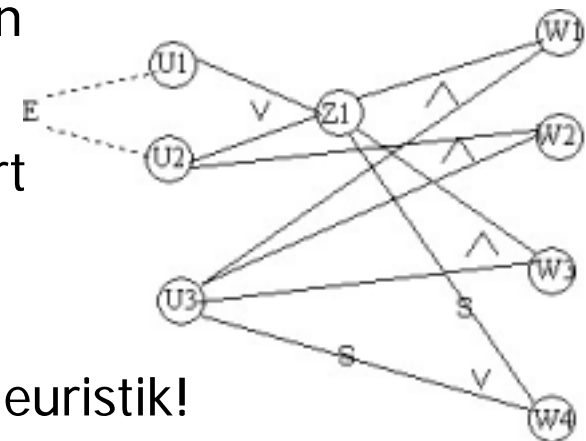
- Man wählt eine Wirkung aus, die im Zustand „vorhanden“ ist
- Durch Zurückverfolgung durch den Ursache-Wirkungs-Graphen bestimmt man alle Kombinationen von Ursachen, die diese Wirkung auf wahr setzen
- Für jede Kombination von Ursachen sieht man eine Spalte in der Entscheidungstabelle vor
- Für jede Kombination bestimmt man die Zustände aller anderen Wirkungen und trägt sie in jede Spalte ein

- Beispiel von der Webseite

- war nicht nach dieser Heuristik konstruiert
- hatte nur exemplarischen Charakter

- Hausaufgabe

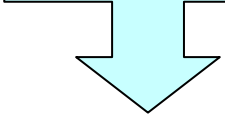
- Konstruieren Sie die Testfälle nach o.a. Heuristik!



Kapitel 2. Testverfahren

2.1 Testen im SW-Lebenszyklus	
2.2 funktionsorientierter Test	
➤ Modul- oder Komponententest	
➤ Integrations- und Systemtests	19.11.
2.3 strukturelle Tests, Überdeckungsmaße	24.11.
2.4 Test spezieller Systemklassen	26.11.
➤ Test objektorientierter Software	1.12.
➤ Test graphischer Oberflächen	3.11.
➤ Test eingebetteter Realzeitsysteme	8.12.
2.5 automatische Testfallgenerierung	10.12.
2.6 Testmanagement und –administration	15.12.

ungefähre
Planung



Strategien für den Integrationstest

- **testzielorientierte Strategie**

- Je nach vorher definierten Testzielen werden einzelne Komponenten zur Überprüfung dieser Ziele integriert und getestet.
 - z.B. können bei dem Testziel "performante Datenübertragung über das Intranet" zunächst die dafür vorgesehenen Komponenten integriert werden. Das restliche System gruppiert sich anschließend darum herum.

- **geschäftsprozessorientierte Strategie**

- Komponenten werden gemäß ihrer Zugehörigkeit zu einzelnen Geschäftsprozessen (Abrechnung, Lager, usw.) integriert.

- **funktionsorientierte Strategie**

- Testfälle werden so spezifiziert, dass sie auf funktionalen Merkmalen beruhen.
- Komponenten werden dementsprechend integriert und getestet.

dynamischer Integrationstest

- **funktionaler Integrationstest**

- spezifizierte Funktionalität wird auf korrektes Zusammenspiel geprüft. Eine Abweichung liegt vor, wenn:
 - zu wenig Funktionalität geliefert wird, d.h. der Aufrufer eine Teilfunktion erwartet, die nicht vorhanden oder erreichbar ist
 - zu viel Funktionalität geliefert wird, d.h. eine unerwartete Teilfunktion ausgeführt wird, oder
 - eine falsche Funktionalität geliefert wird

- **wertbezogener Integrationstest**

- Schnittstellen werden mit Extremwerten beaufschlagt, die ähnlich der Testfallermittlung in der Grenzwertanalyse gewonnen werden.

strukturorientierter Integrationstest

- **kontrollflussorientierter Integrationstest**
 - Aufrufbeziehungen der einzelnen Komponenten untereinander. D.h. alle Funktionen, die eine Komponente dem System zur Verfügung stellt (exportiert), müssen auch aufgerufen werden und jeder Aufruf muss mindestens einmal durchlaufen werden
- **datenflussorientierter Integrationstest**
 - Übergabeparameter und globale Variablen werden betrachtet
 - Überprüfung des Datenflusses
 - über globale Variablen und Parameter vor Aufruf einer importierten Operation und nach Eintritt in diese Operation
 - vor Austritt aus der importierten Operation und nach Rückkehr zum Aufrufer
 - In der aufgerufenen Komponente muss jeder Parameter mindestens einmal lesend verwendet werden

System- und Abnahmetest

- abschließender Test bei der Produkt(neu)-entwicklung
- Kundensicht statt Entwicklersicht
- mit oder ohne Auftraggeberbeteiligung
- möglichst in kontrollierter Testumgebung
- Basis: Produktdefinition (Pflichtenheft, Spezifikation etc.) und Produkt (Software, Handbücher etc.)
- Ziele
 - Vollständigkeit
 - Leistung bzw. Effizienz
 - Zuverlässigkeit und Robustheit
 - Sicherheit (Security)
 - Benutzbarkeit, Dokumentation

Systemtest: Vollständigkeit

- Überprüfung aller in der Spezifikation geforderten Systemeigenschaften
- Black-Box-Sicht auf das SUT
- Testdokumentation nach standardisierter Beschreibung
- Installation, Interoperabilität, Inbetriebnahme?

Systemtest: Leistung

- Massen- bzw. Zeittest: Wieviele Daten können wie schnell verarbeitet werden
 - Massentestdatengenerierung
 - durch automatische Skripte, Testdatengenerator
 - Importschnittstelle
 - praxisgerechtes Datenprofil?
 - reale Daten vom Auftraggeber?
 - Realzeitverhalten
 - Spezifikation mit festen Daten?
 - Tester muss schneller als Testling sein
 - Kommunikations-Verzögerungszeiten?

Systemtest: Zuverlässigkeit

- Lasttest: Test des Systems an den (geforderten) Grenzen der Leistungsfähigkeit
 - Beispiel: Mehrbenutzerbetrieb mit maximaler Benutzeranzahl, alle Benutzer verlangen gleichzeitig hohe Systemleistung
- Stresstest: temporäres Überschreiten der Grenzen
 - Wie verhält sich das System bei Überlast?
 - Normalisiert sich das System anschließend wieder?
- Robustheitstest: Test des Systemverhaltens bei Ausfall einzelner Teile oder unter anormalen Umgebungsbedingungen (Fehlertoleranz!)
 - z.B. Nichtverfügbarkeit von Systemressourcen, Ausfall von externen Softwareteilen, fehlerhafte Schnittstellenbenutzung
 - nichtzerstörende Methoden erwünscht!

Systemtest: Sicherheit

- Überprüfung der spezifizierten (!) Sicherheitsmaßnahmen
 - z.B. Zugriffsrechtsverletzungen auf Systemdaten
 - z.B. Abgleich von Passwörtern mit Lexika
- „systematische“ Einbruchversuche schwierig
- mindestens: Überprüfung bekannter Sicherheitslöcher

Systemtest: Benutzbarkeit

- Benutzerakzeptanz entscheidend!
- Evaluation von Oberflächen schwierig!
- meist nicht durch systematische Test, sondern Ausprobieren
 - Benutzung durch ausgewählte nichtvorgebildete Benutzer
 - Inspektion der Handbücher und Hilfe bzgl. Verständlichkeit
- Systematische Möglichkeiten
 - Aufzeichnung von Benutzerinteraktionen (wann wurde welcher Knopf geklickt)
 - Aufzeichnung von Mausspuren (Wegeminimierung)
 - Videoüberwachung in der Initialphase (wann musste das Handbuch konsultiert werden)
- Beispiel: Otto-Versand

Abnahmetest

- immer mit Auftraggeberbeteiligung
 - normalerweise in realer Einsatzumgebung
 - ggf. mit echten Daten, normale Betriebsbedingungen
- Unterteilung
 - Abnahmekriterien aus der Produktdefinition, Beispiele aus dem Benutzerhandbuch
 - Testfälle aus dem Systemtest
 - Test des typischen Verhaltens über einen gewissen Zeitraum
 - unsystematisches Ausprobieren (protokolliert)
- juristische Relevanz! (Unterschriften)

Vorgehensweise Abnahmetest

- Neuerzeugen bzw. Installieren des SUT
- danach Ausführen der Tests
- Gewichtung der aufgetretenen Probleme
 - Abnahmebescheinigung
 - Nachbesserungsforderungen
 - Rückgabe bzw. Minderung
- Feldtests (durch die künftigen Anwender)
 - alpha-Test: beim Hersteller
 - beta-Test: beim Kunden bzw. Anwender
 - Protokollierung der Fehler!
 - Problematik Public-domain-Feldtests

gescheiterter Abnahmetest



Does This Mean I Fail The Test