

VU Softwarequalitätssicherung WS 2006

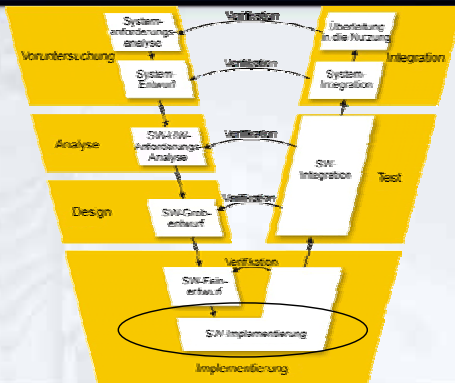
Testen als Entwickler

Mag. Dipl.-Ing. Denis Frast
denis.frast@qse.ifs.tuwien.ac.at

Motivation

- Erschaffung qualitativ hochwertiger Software erfordert
 - Einhaltung von definierten und passenden Prozessen
 - Gewissenhaftigkeit und Einsatz vom Einzelnen
- Fehlervermeidung ist einfacher und billiger als Fehlerkorrektur
- → Anforderungen an den Entwickler
 - Einhaltung von Standards und Konventionen
 - Einhaltung von Prozessen (z.B. CM)
 - Durchführung gewissenhafter Tests (sog. „private“ Tests oder Entwicklertests)

Entwicklertests



Private Tests

- Tests durch den Entwickler vor der „Freigabe“ an andere (Tester)
- Meist undokumentiert
- Endkriterium oft „Ich bin mir sicher, dass das Programm korrekt funktioniert“
- Häufige Fehleinschätzung: „Hier kann eh nichts schief gehen“
- Besser: objektive Kriterien bzw. Methoden, z.B. Abdeckung aller Codezeilen (CO-Überdeckung), aller Methoden etc.

JUnit, .NETUnit etc



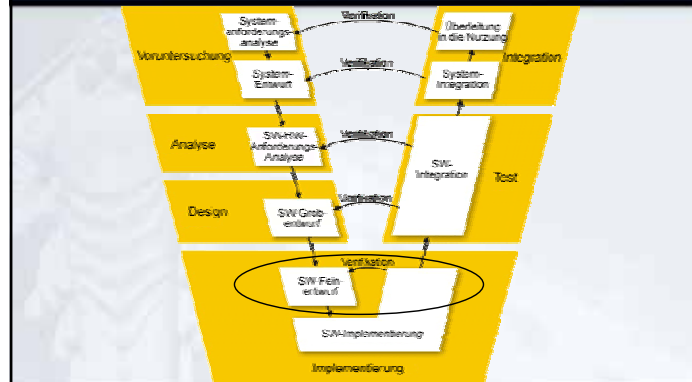
- Entwickler erstellt Test-Methoden (Testfälle) während oder sogar vor der Programmierung
- Testfall: Aufruf von Methoden mit bestimmten Parametern, Überprüfung von Ergebniswerten bzw. -zuständen
- Herstellung eines Ausgangszustands innerhalb des Testfalls oder durch andere Testfälle
- Framework erlaubt einfache Erstellung, Ausführung und Auswertung der Testfälle
- somit einfache Wiederholung von Tests z.B. nach Änderungen und Korrekturen möglich (Regressionstests)

30.11.2006

QS VU - Testorganisation

5

Modultests im V-Modell



30.11.2006

QS VU - Testorganisation

6

Module



- Ein Modul hat folgende Eigenschaften
 - Es ist das Werk eines Entwicklers
 - Es hat eine dokumentierte Spezifikation
 - Es ist ein sichtbares, identifizierbares Produkt mit expliziter Integration ins Gesamtsystem
 - Es kann kompiliert/interpretiert und getrennt von anderen Modulen getestet werden
 - Es ist notwendig

30.11.2006

QS VU - Testorganisation

7

Modultests



- Ziel: Aufspürung von Fehlern in der Implementierung
- Module werden jeweils einzeln gegen ihre Spezifikation getestet: Übergabe von Daten, Prüfung des Outputs
- Modultests erlauben eine genaue Lokalisierung und frühe Erkennung von (Implementierungs-)Fehlern
- Modultests können v.a. bei größeren Systemen sehr aufwendig werden, weshalb sie dann unstrukturiert und undokumentiert gestaltet werden – das ist falsch!

30.11.2006

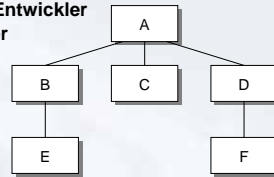
QS VU - Testorganisation

8

Modultests: Dilemma



- Grundsatz: Entwickler sollten ihre Software nicht selbst testen
- bei Modultests häufig verletzt, wenn
 - fundiertes Wissen notwendig ist, das nur der Entwickler selbst hat
 - eigenständiger Test der Module nicht möglich ist, sondern nur mit Stubs und Treibern, die der Entwickler eher erstellen kann als der Tester
- Kompromisslösung: Modultest durch andere Entwickler

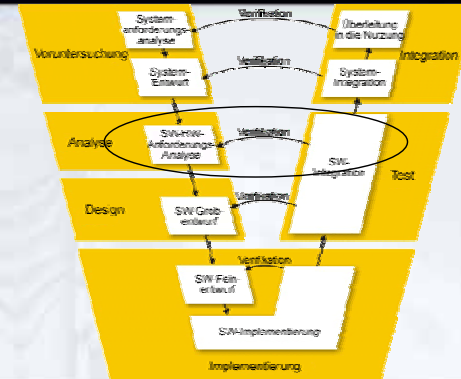


30.11.2006

QS VU - Testorganisation

9

Integrationstests



30.11.2006

QS VU - Testorganisation

10

Integrationstests



- Test der Interaktion zwischen Modulen
- Zusammenführung von Modulen zu größeren Strukturen
 - „Big Bang“
 - Inkrementell: Top-Down oder Bottom-Up
- Häufig Zusammenführung durch Entwickler, Test durch Tester
- Inkrementelles Testen ist vorzuziehen
 - weniger Aufwand (weniger Stubs/Treiber nötig)
 - frühere Erkennung von Schnittstellenfehlern
 - Fehler-Lokation besser eingrenzbar (leichteres Debugging)
 - Weniger Gefahr der Überdeckung von Fehlern in einem Modul durch Fehler in einem anderen

30.11.2006

QS VU - Testorganisation

11

Top-Down oder Bottom-Up?



Bottom-Up

- + frühe Erkennung von Fehlern in der untersten Hierarchie-Ebene
- + Testbedingungen sind leichter zu schaffen
- + Beobachtung der Testausgaben ist einfacher
- Treiber-Module müssen erzeugt werden
- Programm als Gesamtheit existiert erst am Ende

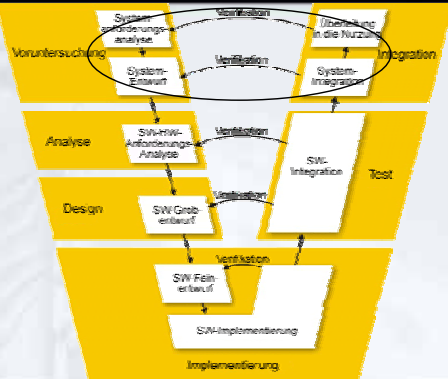
Top-Down

- + frühe Erkennung von Fehlern in der obersten Hierarchie-Ebene
- + Frühes Programm skelett erlaubt Demonstration und erhöht die Moral
- Manche Testbedingungen sind nur schwer zu schaffen
- Beobachtung der Testausgaben ist schwieriger

30.11.2006

QS VU - Testorganisation

12



- **Systemtest**
 - Test des Gesamtsystems gegen die Anforderungsanalyse bzw. den System-Entwurf
 - *“Bridges don't fall down because of bad steel, but because of bad architecture” [Beizer, 1984]*
 - durchgeführt durch Tester bzw. Test-Team

- **Abnahmetest bzw. Akzeptanztest**
 - vom Kunden durchgeführter Systemtest

VU Softwarequalitätssicherung WS 2006

Testen als Tester

Mag. Dipl.-Ing. Denis Frast
denis.frast@qse.ifs.tuwien.ac.at

- **Als Testmanager:**
 - Planung, Vorbereitung, Überwachung und Organisation von Tests und Test-Teams
 - Erstellung von Testberichten (Schnittstelle zur Projektleitung)

- **Als Testingenieur:**
 - Erstellung von Teststrategien
 - Auswahl von Testmethoden und Werkzeugen

- **Als Testentwickler:**
 - methodische Erstellung von Testfallspezifikationen
 - Umsetzung von Testfällen mit Testwerkzeugen, z.B. Capture/Replay

- **Als Tester:** Durchführung und Protokollierung von Tests

Herausforderungen für den Tester



- Fundiertes Wissen über Testmethoden und Testwerkzeuge
- Überblick bewahren auch bei komplexen Systemen
- Umgang mit
 - verschiedenen Abteilungen im Unternehmen
 - gegensätzlichen Interessen
 - Unvorhersehbarkeit
 - Komplexen administrativen Aufgaben
 - Mangel an Erfahrung(-szahlen)
 - Zeitdruck

30.11.2006

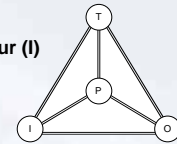
QS VU - Testorganisation

17

Testen als Prozess



- Testen ist der Prozess der Planung, Vorbereitung und Messung mit dem Ziel, die Eigenschaften eines Produkts festzustellen und den Unterschied zwischen dem tatsächlichen und dem erforderlichen Zustand aufzuzeigen.
- Der Testprozess besteht aus vier Eckpfeilern:
 - Ein Phasenmodell für Testaktivitäten (P)
 - Brauchbare Testmethoden (T)
 - Die richtigen Ressourcen und Infrastruktur (I)
 - Solide organisatorische Einbettung (O)



30.11.2006

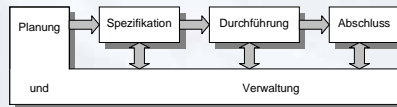
QS VU - Testorganisation

18

Testen als Prozess



- Also was tun Sie wenn Sie die Aufgabe bekommen, ein Programm zu testen?
 - „Unter Testen versteht man den Prozess des Planens, der Vorbereitung und der Messung, mit dem Ziel, die Eigenschaften eines IT-Systems festzustellen und den Unterschied zwischen dem tatsächlichen und dem erforderlichen Zustand aufzuzeigen.“ [Pol et al., 2000]
- Phasen im Testprozess
 - Planung und Verwaltung
 - Spezifikation
 - Durchführung
 - Abschluss



30.11.2006

QS VU - Testorganisation

19

Organisatorische Aspekte



- Betrieblich:
 - Wie viele Test-Personen?
 - Wann sollen sie was tun?
 - Welches Wissen benötigen sie?
 - ...
- Strukturell:
 - Welche Organisationsstruktur?
 - Ein Test-Team für alle Projekte?
 - ...
- Management: Verwaltung und Kontrolle
- Personal und Ausbildung: Kurse und Motivation

30.11.2006

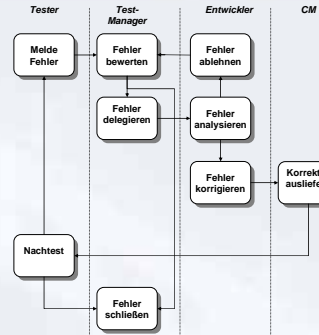
QS VU - Testorganisation

20

Organisatorische Best Practices



- Allgemein zugängliche Dokumente
 - Testplan
 - Testfall-Spezifikationen
 - Testprotokolle
- Verfolge die Fehler
 - Fehlerverfolgungs-Tool
 - Definierter Workflow
- Sichere die Testware
 - Erzeuge Backups der Testdatenbanken
 - Protokolliere jeden Testfall, auch ad hoc erfundene



30.11.2006

QS VU - Testorganisation

21

Test-Stufen und Test-Typen



- Test-Stufen
 - Modul-Test
 - Integrations-Test
 - System-Test
 - Regressions-Test
 - Akzeptanz-Test
- Test-Typen
 - Review
 - Funktionaler Test
 - Massen-Test
 - Stress-Test
 - Performance-Test
 - ...
- Jede Teststufe besteht aus einem oder mehreren Testtypen
- Jeder Testtyp testet ein oder mehrere Qualitätskriterien und verlangt eigene Test-Methoden

30.11.2006

QS VU - Testorganisation

22

Test-Methoden



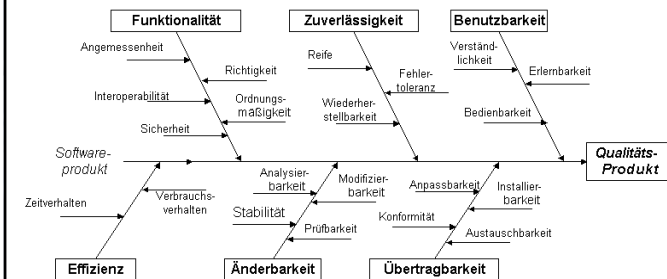
- Finden einer Teststrategie
 - Ziel: die **wichtigsten** Fehler so **früh** wie möglich mit den **geringsten Kosten** zu finden → effizienter Personaleinsatz!
 - Variation in der Testintensität je Produktkomponente und Qualitätsmerkmal
 - Unterschiedliche Strategie je Test-Stufe
- Testfallspezifikation
 - Definition der Testfälle der Teststrategie entsprechend und im voraus
- Checklisten
 - Test von Qualitätskriterien ohne Programmausführung
- Review-Techniken
 - Dokumente und Source-Code im Team prüfen

30.11.2006

QS VU - Testorganisation

23

Qualitätskriterien nach ISO 9126 / DIN 66272



30.11.2006

QS VU - Testorganisation

24

Test-Strategie – Generisches Beispiel



Kriterium	Kompon. (Größe)	Teilsyst. 1 (50)	Teilsyst. 2 (20)	Teilsyst. 3 (15)	Migration (10)	Gesamt-system (5)	Relative Bedeutung
Sicherheit		+	+				5
Einsetzbarkeit							-
Kontinuität							-
Kontrollierbarkeit							-
Funktionalität		++	+	+	++	+	60
Benutzungs-freundlichkeit		++	+				10
Leistung		+		+			5
Integrierbarkeit		+	+	+		++	20
Sparsamkeit							-
Rel. Bedeutung		30	15	20	15	20	100%

30.11.2006

QS VU - Testorganisation

25

Teststrategie – Funktionaler Test



Methode	Kompon. G/B	#TF / kLOC	Teilsyst. 1 (50/30)	Teilsyst. 2 (20/15)	Teilsyst. 3 (15/20)	Migration (10/15)	Gesamt (5/20)
Strukturtest Maß 2		80					
Strukturtest Maß 1		60					
Entsch.tabellentest		40					
Datenzyklus-Test		30					
Error Guessing		10					

- Schätze Anzahl Testfälle je Methode und Größeneinheit sowie Aufwand je Testfall (wenn möglich, aus empirischen Daten!)
- Ermittle Aufwand für mehrere verschiedene Szenarien (hier H, M, L) für Angebote, Reaktion auf eingetretene Risiken etc.
- Beispiel: Definiere 3 Szenarien H, M, L.

30.11.2006

QS VU - Testorganisation

26

Zusammenfassung



- Testen ist ein Prozess, der während des gesamten SE-Zyklus stattfindet
- Der Testprozess basiert auf vier Eckpfeilern
 - Phasenmodell
 - Techniken
 - Organisatorische Einbettung
 - Ressourcen und Infrastruktur
- Testplanung ermöglicht es, durch effizienten Personaleinsatz wichtige Probleme früh und kostengünstig zu erkennen
- Die wichtigste Komponente der Testplanung umfasst die Ermittlung der wichtigsten Qualitätsmerkmale und Systemkomponenten, deren Gewichtung sowie die darauf basierende Planung von Tests unterschiedlicher Intensität

30.11.2006

QS VU - Testorganisation

27

Fragen?



30.11.2006

QS VU - Testorganisation

28