



Zusammenfassung

- Fehlerwirkung - Nichterfüllung einer Anforderung!
 - Beschreibung dieses Sachverhalts als Fehlerwirkung (*failure*)
 - Ursache: Fehlerzustand (*fault*) in der Software
 - die durch eine Fehlhandlung (*error*) einer Person verursacht wurde
- Tests sind wichtige Maßnahmen zur Sicherung der Qualitätsmerkmale
 - Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit und Übertragbarkeit [nach ISO 9126]
- Austesten ist nicht möglich, Tests sind immer Stichproben, sie sollen deshalb mit einer gewissen Wahrscheinlichkeit Fehlerwirkungen nachweisen
- Intensität und Umfang der Tests ist abhängig vom erwarteten Risiko bei fehlerhaftem Verhalten des Programms
- Priorisierung der Tests ist vorzunehmen



Zusammenfassung

- Der fundamentale Testprozess besteht aus den Hauptaktivitäten
 - Testplanung und -steuerung
 - Testanalyse und Testentwurf
 - Testrealisierung und -durchführung
 - Bewertung von Ausgangskriterien und Bericht
 - Abschluss der Testaktivitäten
- Ein Testfall besteht aus Eingabewert und Soll-Ergebnis sowie Vorbedingungen, unter denen der Testfall ablaufen muss, und Nachbedingungen, die nach Ablauf erfüllt sein müssen.
- Wird der Testfall ausgeführt, zeigt das Testobjekt ein Ist-Verhalten.
 - Sind Soll- und Ist-Verhalten verschieden, liegt ggfs. eine Fehlerwirkung vor.
 - Es ist zu prüfen, ob ein Fehler im Soll-Verhalten, im Testfall, in der Spezifikation oder im Testobjekt liegt.
- Ein Testorakel bestimmt die Soll-Werte für jeden Testfall vor Testdurchführung.
- Menschen machen Fehler – aber sie geben es nur ungern zu!



Zusammenfassung (1)



- Das allgemeine V-Modell definiert grundlegende Teststufen: Komponententest, Integrationstest, Systemtest und Abnahmetest und unterscheidet zwischen verifizierender und validierender Prüfung.
- Der Komponententest testet einzelne Softwarebausteine.
- Der Integrationstest prüft das Zusammenwirken bereits getesteter Bausteine.
- Funktionaler und nicht-funktionaler Systemtest betrachten das Gesamtsystem aus Sicht des späteren Anwenders.
- Im Abnahmetest prüft der Auftraggeber das erstellte Produkt auf vertragliche Akzeptanz und Benutzerakzeptanz.
- Wenn das System in sehr vielen Produktivumgebungen betrieben werden soll, bieten Alpha- und Beta-Tests eine zusätzliche Möglichkeit, Einsatzerfahrungen mit Vorabversionen des Produkts zu sammeln.

Zusammenfassung (2) – Vergleich der Teststufen

Kriterium	Komponententest	Integrationstest	Systemtest	Abnahmetest
Testziele	Fehlerzustände in Software (-bausteinen), die separat getestet werden können, finden	Fehlerzustände in Schnittstellen und im Zusammenspiel zwischen integrierten Komponenten finden.	Prüfung, ob die spezifizierten Anforderungen (funktional, nicht-funktional) vom Produkt erfüllt werden.	Vertrauen in das System oder in bestimmte nicht-funktionale Eigenschaften gewinnen
Testbasis	<ul style="list-style-type: none"> →Komponentenspezifikation →Detaillierter Entwurf →Datenmodell → Programmcode 	<ul style="list-style-type: none"> →Software- und Systementwurf → Architektur →Nutzungsabläufe, Workflows → Anwendungsfälle 	<ul style="list-style-type: none"> →System- und Anforderungsspez. → Anwendungsfälle → funktionale Spezifikation → Geschäftsprozesse → Risikoanalyseberichte 	<ul style="list-style-type: none"> →Benutzeranforderungen → Systemanforderungen → Anwendungsfälle → Geschäftsprozesse → Risikoanalyseberichte
Typische Testobjekte	Isolierte Softwarebausteine (Klasse, Unit, Modul) <ul style="list-style-type: none"> → Komponenten, Programme, → Datenumwandlungs-/ Migrationsprogramme, → Datenbankmodule 	Zu integrierende Einzelbausteine, Subsysteme und zugekaufte Standard-Komponenten <ul style="list-style-type: none"> →Datenbankimplementierungen → Infrastruktur → Schnittstellen → Systemkonfiguration und Konfigurationsdaten 	<ul style="list-style-type: none"> →System-, Anwender- und Betriebshandbücher → Systemkonfiguration und Konfigurationsdaten 	<ul style="list-style-type: none"> →Geschäftsprozesse des integrierten Systems → Betriebs- und Wartungsprozesse → Anwenderverfahren → Formulare → Berichte → Konfigurationsdaten
Testwerkzeuge	Entwicklungsumgebung, Debugging-Unterstützung Stat. Analysewerkzeuge, Komponententestumgebung	Testmonitore zur Überwachung des Datenaustauschs zwischen Komponenten	Testmanagement-Werkzeuge, GUI-Automatisierungswerkzeuge	
Testumgebung	Platzhalter, Treiber, Simulatoren	Wiederverwendung/ Erweiterung der Platzhalter, Treiber, Simulatoren aus dem Komponententest	Test- und Produktivumgebung sollten so weit wie möglich übereinstimmen.	Test- und Produktivumgebung sollten so weit wie möglich übereinstimmen.



Zusammenfassung (3)



- Durch Fehlerkorrekturen (Wartung) und geplante Weiterentwicklung (Pflege) wird ein Softwareprodukt im Laufe seines Lebenszyklus immer wieder geändert oder erweitert.
- Jede dieser geänderten Versionen muss erneut getestet werden.
- Welchen Umfang solche Regressionstests haben, muss eine individuelle Risikoabschätzung festlegen.
- Die grundlegenden Testarten sind funktionaler und nicht-funktionaler Test, sowie struktur- und änderungsorientierter Test



Zusammenfassung



- Mehrere Augenpaare sehen mehr als ein Augenpaar
- Der Reviewprozess umfasst typischerweise Planung, Kick-off, individuelle Vorbereitung, Reviewsitzung, Überarbeitung und Nachbereitung
- Rollen sind:
Manager, Moderator, Autor, Gutachter und Protokollant
- Arten von Reviews sind
informelles Review, Walkthrough, technisches Review und Inspektion
- Werkzeuggestützte statische Analysen ohne Ausführung des Prüfobjektes sind für Dokumente mit gewissem Formalisierungsgrad möglich
- Compiler decken Fehlerzustände in der Syntax des Programmcodes auf, bieten aber meist noch weitere Möglichkeiten



Zusammenfassung



- Analysewerkzeuge melden Verstöße gegen Standards und andere Konventionen
- Anomalien im Daten- und Kontrollfluss der Programmtexte weisen oft auf fehlerträchtige Stellen hin
- Metriken messen Eigenschaften von Produkten, Projekten und Prozessen. Um Aussagen über deren Qualität zu treffen, müssen sie aber interpretiert werden
- Die zyklomatische Zahl errechnet die Anzahl der linear unabhängigen Pfade im untersuchten Programmtext und gibt Hinweise zum Test- und Wartungsaufwand



Zusammenfassung



- Dynamische Tests führen das Testobjekt aus
- Spezieller Testrahmen oft notwendig
- Auswahl der Testfälle als (gute) Stichprobe
- Black-Box-Testentwurfsverfahren benötigen zur Auswahl der Testfälle keine Kenntnis der Programmlogik
- Funktionale Tests leiten die Testfälle anhand der Spezifikation des Testobjekts ab
- Äquivalenzklassenbildung in Kombination mit der Grenzwertanalyse zur Erstellung der Testfälle einsetzen
- Zustandsbasierte Tests mit Übergangsbaum und Zustands-Konformanztest sowie erweitertem Übergangsbaum und Zustands-Robustheitstest
- Entscheidungstabellentest bei regelbasierten Anforderungen
- Anwendungsfallbasierter Test für Szenarien der Systembenutzung



Zusammenfassung dynamischer Test (1)



- Allgemeines Ziel: mit wenig Aufwand ausreichend unterschiedliche Testfälle zu erzeugen, die mit einer gewissen Wahrscheinlichkeit vorhandene Fehlerzustände zur Wirkungen bringen
 - Testentwurfsverfahren zur Erstellung der Testfälle passend zum Testobjekt auswählen
 - Auf jeden Fall ausreichende Prüfung der Funktionalität des Testobjektes gewährleisten
 - Bei der Aufstellung aller Testfälle erwartete Ergebnisse und Reaktionen des Testobjektes mit vermerken, so dass eine Prüfung der Funktionalität bei jeder Auswertung der durchgeführten Testfälle erfolgt
 - Bei jedem Testobjekt Äquivalenzklassenbildung in Kombination mit der Grenzwertanalyse zur Erstellung der Testfälle einsetzen
 - Bei der Ausführung dieser Testfälle die codebasierte Überdeckung messen und die nach der Durchführung aller Testfälle der Äquivalenzklassenbildung und Grenzwertanalyse bereits erreichte Anweisungs- oder Entscheidungsüberdeckung ermitteln
 - Haben unterschiedliche Zustände einen Einfluss auf den Ablauf innerhalb des Testobjektes, muss ein zustandsbasierter Test durchgeführt werden



Zusammenfassung dynamischer Test (2)



- Bisher nicht ausgeführte Teile des Testobjektes werden dann gezielt einem der White-Box Testentwurfsverfahren unterzogen
 - Je nach Kritikalität und Beschaffenheit des Testobjektes entsprechend aufwändiges White-Box Testentwurfsverfahren wählen
 - Als minimales Kriterium die Entscheidungsüberdeckung verwenden
 - Bei den White-Box Testentwurfsverfahren soll die Struktur des Testobjektes die Grundlage bei der Auswahl der Testentwurfsverfahren sein - sind z.B. komplexe Bedingungen im Testobjekt vorhanden, so ist die minimale Mehrfachbedingungsüberdeckung das adäquate Testentwurfsverfahren, um fehlerhafte Bedingungen zu erkennen.
 - Bei den Überdeckungsmessungen darauf achten, das Schleifen auch mehr als einmal wiederholt werden
 - Bei kritischen Systemteilen muss die Prüfung der Schleifen anhand von entsprechenden Methoden erfolgen
 - Die Pfadüberdeckung ist als eher theoretisches Maß anzusehen und hat auf Grund des riesigen Aufwands für die Praxis keine Bedeutung
- Die White-Box Testentwurfsverfahren lassen sich sinnvollerweise auf den unteren Teststufen einsetzen; auf den oberen Teststufen sind die Black-Box Testentwurfsverfahren die adäquaten Methoden



Zusammenfassung dynamischer Test (3)



- Auf die erfahrungsbasierte Ermittlung der Testfälle als Ergänzung sollte nicht verzichtet werden – es ist immer sinnvoll, die Erfahrungen der Tester zu nutzen, um weitere Fehler aufzudecken
- Testen umfasst immer die Kombination von unterschiedlichen Testentwurfsverfahren, da es kein Testentwurfsverfahren gibt, dass alle Aspekte, die beim Testen zu berücksichtigen sind, gleich gut abdeckt.
 - Die Auswahl der Testentwurfsverfahren und die Intensität der Durchführung sind anhand der Kritikalität und dem zu erwartenden Risiko im Fehlerfall festzulegen



Zusammenfassung (1)



- Entwicklungs- und Testaktivitäten sollen organisatorisch getrennt sein. Je klarer diese Trennung erfolgt, umso wirksamer kann getestet werden!
- Je nach Aufgabenstellung innerhalb des Testprozesses werden Mitarbeiter mit rollenspezifischen Testkenntnissen benötigt. Neben fachlichen Fähigkeiten ist auch soziale Kompetenz gefragt!
- Zu den Aufgaben des Testmanagers gehören Planung, Überwachung und Steuerung der einzelnen Testzyklen
- In dem mit dem übergeordneten Qualitätssicherungsplan abgestimmten Testkonzept beschreibt und begründet der Testmanager, wie er die anstehenden Testaufgaben lösen will, welchen Testumfang er plant, welche Werkzeuge eingesetzt werden sollen usw.



Zusammenfassung (2)



- Fehler und Mängel, die im Test übersehen werden, können hohe Fehlerkosten nach sich ziehen. Bei der Wahl der Teststrategie wird ein optimales Verhältnis zwischen Testkosten, verfügbaren Ressourcen und drohenden Fehlerkosten angestrebt
- Risikomanagement und risikoorientierter Test helfen, die kritischen Fehler frühzeitig zu finden und den Testaufwand zu optimieren
- Fehlermanagement und Konfigurationsmanagement bilden die Basis für einen effizienten Testprozess
- Fehlermeldungen müssen nach einem projektweit einheitlichen Schema erfasst und über alle Stadien des Fehleranalyse- und Korrekturprozesses verfolgt werden
- Normen und Standards enthalten Vorgaben und Empfehlungen zur fachgerechten Durchführung von Softwaretests. Eine Orientierung an solchen Standards macht auch dort Sinn, wo deren Einhaltung nicht verpflichtend vorgeschrieben ist



Zusammenfassung (1)

- Für jede Phase im Testprozess sind Werkzeuge zur Qualitätsverbesserung und/oder zur Automatisierung der Testarbeiten verfügbar
- Testwerkzeuge können unterschiedlich klassifiziert werden
 - Je nach Aktivitäten des Testprozesses, die unterstützt bzw. automatisiert werden
 - Je nachdem, ob das Werkzeug das Verhalten des Testobjekts beeinflusst oder nicht
 - Je nachdem, wer das Werkzeug typischerweise benutzt
- Unterschiedliche Ansätze zur Automatisierung der Testdurchführung
 - Capture / Replay Ansatz: „Ad-hoc“ Automatisierung
 - Datengetriebener Ansatz: Trennung von Testdaten und Testvorgehensspezifikation
 - Schlüsselwortgetriebener Ansatz: Trennung von Testentwurf und Testrealisierung



Zusammenfassung (2)

- Der Einsatz von Testwerkzeugen verspricht zahlreiche Vorteile, z.B. die Einsparung von Ressourcen oder die Erhöhung der Transparenz der Testdokumentation
- Dem potenziellen Nutzen stehen häufig hohe Investitionskosten gegenüber
- Die Auswahl sollte daher sorgfältig vorbereitet und gegen klar spezifizierte Anforderungen erfolgen, um eine objektive Auswahl eines Testwerkzeugs zu gewährleisten
- Das Vorhandensein eines Werkzeugs garantiert aber nicht, dass dieses von den Anwendern auch benutzt wird. Rechtzeitige Information und Schulung der Betroffenen erhöhen die spätere Akzeptanz des ausgewählten Testwerkzeugs



Zusammenfassung (3)

- Bevor ein Testwerkzeug unternehmensweit eingeführt wird, sollten Erfahrung im Rahmen eines Pilotprojektes gesammelt werden
- Bei der Einführung der verschiedenen Arten von Testwerkzeugen ist es empfehlenswert, intellektuell anspruchsvolle Aktivitäten immer zuletzt zu automatisieren
- **Der Einsatz von Testwerkzeugen bedingt einen funktionierenden Testprozess!**