

Software- Qualitätsmanagement

**Vorlesung im Modul 10-202-2319
Software-Management**

Sommersemester 2010

Prof. Dr. Hans-Gert Gräbe

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

Integrationstests

Die Integrationstests dienen der Überprüfung der **Schnittstellen** zwischen den System-Komponenten.

- Aufruf von Operationen, Funktionen und Prozeduren mit und ohne Parameterübergabe
- Verwendung von globalen Variablen oder Dateien
- Benutzung von global vereinbarten Konstanten und Typen
- Analyse der Bindung zwischen den Komponenten

Integrationstests werden unter verschiedenen Aspekten ausgeführt

Dynamischer Integrationstest

- **Ziel:** Plausibilität der funktionalen Korrektheit durch Testfälle, die ausschließlich in der Integrationsphase nachweisbare Fehler abdecken.
- **Vorgehen:** Stichprobentest, wie beim funktionalen Test von Komponenten

Strukturorientierter Integrationstest

- wie Strukturtests für Komponenten
- *Kontrollflussorientierter Integrationstest* betrachtet die unterschiedlichen Aufrufbeziehungen (Exporte und Importe) zwischen Komponenten.
Mögliche Überdeckungskriterien:
 - jeder Aufruf jeder exportierten Operation muss in jeder importierenden Komponente wenigstens einmal überdeckt sein.
 - alle Aufrufstellen sind in allen möglichen Reihenfolgen (mit Schleifenbeschränkung) zu überdecken.
- *Datenflussorientierter Integrationstest* betrachtet die Programmstellen genauer, an denen importierte Operationen aufgerufen werden.
 - analog datenflussorientierten Tests von Komponenten

Funktionaler Integrationstest

- Prüft die spezifizierte Funktionalität der einzelnen System-Komponenten und deren Zusammenwirken.
- Abweichungen liegen vor, wenn die Operation:
 - zu wenig Funktionalität liefert (z. B fehlende Teilfunktion),
 - zu viel Funktionalität liefert (z. B. Aufruf einer unerwarteten Teilfunktion) oder
 - falsche Funktionalität liefert.
- Diese Fehler resultieren meist aus ungenauen Spezifikationen.
 - werden beim Komponententest nicht erkannt, da dort die Spezifikation als „gesetzt“ gilt

Wertbezogener Integrationstest

- Schnittstellen werden mit möglichst extremen Werten getestet
- Entspricht der Grenzwertanalyse

Statischer Integrationstest

- Analysierendes Verfahren des Quellcodes der beteiligten System-Komponenten
- untersucht die Kopplung zwischen den Komponenten und erfasst Parameter systemweiter Metriken
 - **Ziel:** zusätzliche und unnötige Kopplungen identifizieren und eliminieren
- syntaktische Kompatibilität der Schnittstellen über die Komponentengrenze hinaus wird meist auf konstruktivem Weg erreicht
 - automatische Überprüfung der Schnittstellendeklarationen durch den Compiler (Header-Dateien, Import-Deklarationen)
- Analyse von Datenflussanomalien wie innerhalb von Komponenten
- **Verifizierende Methoden** können auch komponentenübergreifend eingesetzt werden.

Integration von OO-Systemen

Klassentest (Methodik bereits beschrieben)

- Integration von einzelnen Methoden einer Klasse
- Integrationstest der Vererbungen und Polymorphie

Integration von Unterklassen

- Unterscheide Integration von reinen dienst anbietenden, reinen dienstnutzenden und gemischten Klassen
- Situation: Oberklasse bereits integriert, Unterklasse für sich bereits getestet, Testmaterial für beide Tests liegt vor
- neben Integrationstests für einzelne Methoden sind auch Integrationstests für Operationen und Folgen von Operationen erforderlich (geht wie bereits beschrieben)

Integration einzelner Methoden

vollständig geerbte Methoden: neue Tests nicht erforderlich

- beim Oberklassen-Integrationstest mit den Dienstnutzern bereits geprüft

redefinierte Methoden mit gleicher Semantik (Spezifikation): neue Testfälle nicht erforderlich

- Integrationstests beziehen sich nur auf die Schnittstelle, funktionaler und struktureller Test erfolgte bereits beim Komponententest

redefinierte Methoden mit anderer Spezifikation: andere Testfälle erforderlich

- Vor- und/oder Nachbedingungen verändert (Einsatzbereich spezieller)

Bei restriktiveren Vorbedingungen Modifikation der Testfälle der Oberklasse erforderlich.

- Die alten Testfälle müssen auf die neue Zusicherung adäquat reagieren
- Beispiel: allg. Konto → Sparkonto
- Typisch für dienst anbietende Klassen

Bei erweiterten Nachbedingungen zusätzliche Testfälle erforderlich.

- zur Überdeckung neuer Nachbedingungen
- Beispiel: Speichern → Speichern mit Rechteverwaltung
- Typisch für dienstnutzende Klassen.

Testen von Sequenzen

- Oft Integration von ganzen Operationssequenzen erforderlich
- Ereignisbaum-Methode
 - ausgelöst durch ein Ereignis, Fortpflanzung durch Botschaften
- Zerlegung in atomare Systemfunktionen
 - Operationssequenzen werden durch ein Start-Ereignis angestoßen und durch ein Ausgabe-Ereignis abgeschlossen.

Der Systemtest

Der **Systemtest** ist der abschließende Test der Software-Entwickler und Qualitätssicherer in der realen Umgebung ohne den Auftraggeber.

- Umfasst Systemsoftware, Hardware, Bedienungsumfeld, technische Anlage
- System muss ggf. vor Beginn des Systemtests von der Entwicklungsauf die Einsatz- oder Zielplattform portiert werden.
- **Basis:** Produktdefinition (Pflichtenheft, Produktmodell, Konzept der Benutzerschnittstelle, Benutzerhandbuch)
 - Pflichtenheft sollte sowohl die Qualitätsziele als auch die Testszenarien und Testfälle fixieren.
- Auf der Grundlage werden **Testfälle** aus den bisherigen Testzyklen übernommen und ergänzt.
- Zerlegung des Systemtests in verschiedene **Teiltests** an Hand zu bestimmender **Prüfziele**.
 - Prüfung aller geforderten Qualitätsziele in ihrer jeweiligen Ausprägung

Prüfziele

- Vollständigkeit
 - Sind alle funktionalen und nicht funktionalen Anforderungen aus dem Pflichtenheft erfüllt? (**Funktionstest**)
- Volumen
 - Systemtest mit umfangreichen Datenmengen (**Massentest**)
- Zeit
 - Systemtest auf Antwortzeiten unter starker Belastung (**Zeittest**)
- Zuverlässigkeit
 - Systemtest unter längerer Spitzenlast im geforderten „grünen“ Bereich (**Lasttest**)
 - auch unter Ausfall einzelner externer Hardware- oder Software-Komponenten
 - Mehrbenutzerbetrieb im Grenzbereich
 - Reaktion auf ungewöhnliche oder widersprüchliche Daten
- Robustheit und Fehlertoleranz
 - Systemtest unter Überlast, im „roten“ Bereich (**Stresstest**)

- Benutzbarkeit
 - Test der Verständlichkeit, Erlernbarkeit, Bedienbarkeit aus der Sicht des Endnutzers (**Benutzbarkeitstest**)
 - Zielgruppenbezogen (Fachtermini, Metaphern etc.)
- Sicherheit
 - Datenschutzmechanismen, Zusammenspiel mit dem umgebenden System (**Sicherheitstest**)
- Interoperabilität
 - Relevant, wenn das System in einen größeren Verbund eingebettet ist (**Kompatibilitätstest**)
 - Kompatibilität der Schnittstellen und der Daten
- Konfiguration
 - wenn vorgesehen, Test der Systemausprägungen für verschiedene Hard- und Softwareplattformen (**Konfigurationstest**)
- Dokumentation
 - Vorhandensein, Angemessenheit und Güte der Benutzer- und Wartungsdokumentation (**Dokumentationstest**)

Teiltests

Funktionstest

- Test, ob alle in der Produktdefinition geforderten Funktionen vorhanden und wie vorgesehen realisiert sind.
- Testsequenzen sind aus dem Pflichtenheft zu übernehmen und/oder mit funktionalen Testverfahren systematisch und vollständig herzuleiten.

Leistungstest

- dient der Überprüfung des in der Produktdefinition festgelegten Leistungsverhaltens
 - Massentest, Zeittest, Lasttest, Stresstest
 - Einsatz eines Testdatengenerators oder realer Daten vom Auftraggeber oder von Pilotkunden
 - Frage der Systemstabilisierung nach Überlastphasen, etwa durch den Entzug von Ressourcen

Benutzbarkeitstest

- Oft entscheidend für die Akzeptanz eines Softwareprodukts
- Kann sehr aufwändig sein, wenn darauf in der Phase der Produktdefinition zu wenig Wert gelegt wurde

Interoperabilitätstest

- heutige Systeme sind in der Regel keine alleinstehenden Systeme, sondern in eine Standardumgebung integriert
- umfasst meist eine komplexe GUI-Schnittstelle zum Betriebssystem
- Frage der Interaktion mit diesen Oberflächen (etwa mit der Zwischenablage in Windows)

Installations- und Wiederinbetriebnahmetest

- **Installationstest:** Prüft, ob das System mit den erstellten Installationsbeschreibungen installiert und in Betrieb genommen werden kann.
- **Wiederinbetriebnahmetest:** Prüft, ob das System nach einer Unterbrechung oder einem Zusammenbruch des Basissystems mit den vorliegenden Beschreibungen wieder in Betrieb genommen werden kann und ob noch alle Daten aktuell und verfügbar sind.

Besonderheiten für OO-Systeme gibt es nicht, da der Systemtest ein Black-Box-Test ist, der gar nicht bemerken kann, ob das System ein OO-System ist.

Systemtest als Regressionstest: Aufzeichnen der Testfälle erlaubt es, diese bei späteren Fehlerkorrekturen oder inkrementeller Software-Entwicklung relativ problemlos zu wiederholen.

Abnahmetest

Der **Abnahmetest** ist eine besondere Ausprägung des Systemtests, bei dem das System getestet wird

- unter Mitwirkung und Federführung des Auftraggebers
- in der realen Einsatzumgebung beim Auftraggeber
- (unter Umständen) mit echten Daten des Auftraggebers

Auftraggeber kann die Testfälle aus dem Systemtest übernehmen, modifizieren und eigene Testszenarien durchführen.

- Konzentration in der Regel auf den Test unter normalen Betriebsbedingungen
- Sollte bereits im Auftrag vereinbart sein, wird aber in der Regel ein „freies Testen“ sein.
- Verfahren des Abnahmetests sollte bereits beim Systemtest zum Einsatz kommen

Methodik aus Auftraggebersicht

- Erzeugen des zu testenden Systems aus den Quellen
 - hilfsweise Löschen aller Objektdaten
 - Bilden und Speichern einer Prüfsumme über das gesamte System, um dessen Unversehrtheit am Schluss zu prüfen
- Durchführung der Abnahme nach der vereinbarten Testvorschrift
 - Einbeziehung des Benutzerhandbuchs (mindestens alle dort angegebene Beispiele müssen funktionieren)
- regelmäßige einvernehmliche schriftliche Fixierung der Testergebnisse
- regelmäßiges freies Testen und Dokumentation dieser Testfälle
- Abnahme endet mit einer Schluss-Sitzung
 - Wichtung der protokollierten Fehler
 - Entscheidung über Annahmen, Auftrag zur Nachbesserung, Ablehnung

Abnahme stellt immer einen Kompromiss zwischen optimalem (also fehlerfreiem) und akzeptablem Ergebnis dar.

Abnahme größerer Systeme

Mehrstufiges Abnahmeverfahren:

Werkabnahme

- Abnahme in einer speziellen werksseitig erstellten Testumgebung
- sinnvoll nur, wenn Installation weit entfernt erfolgen soll oder wenn die Installation den Betriebsablauf nachhaltig stört

Abnahme in der realen Umgebung

- unverzichtbar, evtl. sind Maßnahmen zur Sicherung des Betriebsablaufs zu treffen
- Durchführung auch der Tests, auf die in der Werksabnahme verzichtet werden musste, weil deren Implementierung in der Testumgebung zu aufwändig gewesen wäre

Betriebsabnahme

- Versuchsbetrieb in der Garantiephase mit aufwändigerer Protokollierung des Betriebs
 - Aufzeichnung aller Fehler, Ergänzung der Testreihe
 - Wiederholung der modifizierten Tests mit dem verbesserten System vor der endgültigen Inbetriebnahme

Abnahme von Produkten für den anonymen Markt

Auftraggeber und Nutzer sind verschieden.

- Interner Auftraggeber (Marketingabteilung, Produktmanager) nimmt das Produkt ab
- Systeme werden in der Regel einem Alpha- und Beta-Test unterzogen
 - Prüfziele Fehlertoleranz, Benutzbarkeit, Konfiguration und Interoperabilität lassen sich nur schwer durch den internen Auftraggeber testen
 - aufgetretene Fehler werden protokolliert und beseitigt
- **Alpha-Test:** System wird in der Zielumgebung des Herstellers durch Anwender erprobt.
- **Beta-Test:** System wird ausgewählten Pilot-Kunden in deren eigener Umgebung zur Probenutzung zur Verfügung gestellt.
 - nach umfangreichen Fehlerkorrekturen auch Beta2-Phase möglich
 - Pilotkunden erhalten beim späteren Kauf meist einen Rabatt

Beispiel: Roadmap für Windows Longhorn (<http://www.winsupersite.com>)

- Developer Preview (April 2005)
- Beta 1 (Juni 2005) – parallel zur Integration
- Beta 2 (Ende 2005)
- RC0 (Febr. 2006) – Client Release Candidate, „status feature complete“
- RC1 (April 2006)
- RTM (Juni 2006) – Client release to manufacturing
- Launch (Oktober 2006) – allgemeine Verfügbarkeit
- Server RTM (Client RTM + 6 Monate)

Produktzertifikate

Die Produktqualität eines Software-Systems ist zwar das Ergebnis der Prozessqualität, für den Endkunden aber von eigenständigem Interesse. Hersteller sind damit an Produktzertifikaten interessiert.

- Richtlinie der Gütegemeinschaft Software von 1985 zur einheitlichen Prüfung von Software-Produkten
- Überarbeitung als DIN 66285 sowie ISO 12119 (1994)
- reine Produktnorm, also keine Aussage über den Entwicklungsprozess
- Qualitätsanforderungen beziehen sich auf
 - Produktbeschreibung zu Information des Kunden vor dem Kauf
 - Dokumentation
 - Programme und Daten
- nicht berücksichtigt werden unterstützende Dienstleistungen

ISO 12119 - Qualitätsanforderungen

Produktbeschreibung

- Jedes SW-Erzeugnis muss eine P.-B. besitzen, die festlegt, was das Erzeugnis ist. Die P.-B. soll dem Benutzer oder potenziellen Käufer helfen, die Eignung des Erzeugnisses für ihn zu beurteilen und als eine Prüfgrundlage dienen.
- Unterpunkte spezifizieren und normieren
 - Allgemeine Anforderungen an den Inhalt
 - Bezeichnungen und Angaben
 - Angaben zu Zuverlässigkeit, Benutzbarkeit, Effizienz

Programme und Daten

- Funktionalität: Normen für
 - Benutzerinstallierung
 - Funktionalität entspricht Beschreibung und Dokumentation
 - Widerspruchsfreiheit, gleiche Benennungen mit gleicher Bedeutung

- Zuverlässigkeit: Das System aus Hardware, vorausgesetzter Software und den zum Erzeugnis gehörenden Programmen darf in keinen unbeherrschten Zustand geraten. Daten dürfen nicht verfälscht werden und nicht verloren gehen.
- Diese Anforderung muss auch erfüllt sein
 - bei Belastung bis zu den angegebenen Grenzwerten
 - bei Versuchen, angegebene Grenzwerte zu übersteigen
 - bei fehlerhafter Benutzereingabe oder Fehlfunktionen anderer in der Beschreibung genannter Programme
 - wenn ausdrückliche Anweisungen in der Benutzerdokumentation verletzt werden
- Benutzbarkeit: Das Produkt muss verständlich, übersichtlich und steuerbar sein (etwa DIN 66234 – ergonomische Dialoggestaltung)

Benutzerdokumentation

- muss vollständig, richtig, widerspruchsfrei, verständlich und übersichtlich sein

Auf der Basis sieht die Norm ausführliche Prüfbestimmungen vor.

Die Zertifizierung erfolgt durch unabhängige, akkreditierte Zertifizierungsstellen.

Qualitätssicherung in der Phase Betrieb und Wartung

Bug Tracking

- Einsatz automatisierter webgestützter Systeme wie Bugzilla

Arbeit mit Power Usern

- Rolle von Alpha- und Beta-Test-Phasen

Rolle der Qualität der Daten

ISO 9000 sieht Pflicht zur Nachweisführung vor.

- Wartungsplan, Wartungsaufzeichnungen und -berichte
- Konfigurationsmanagementplan