LE 18
1

Software-Qualitätssicherung

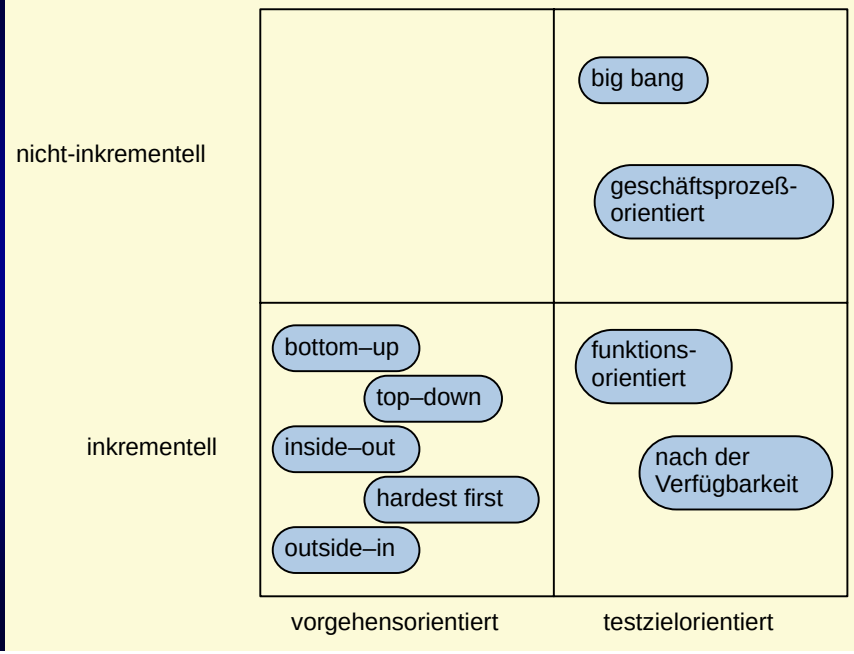
6 Produktqualität – Systeme: Integrationstest

[sehr stark gekürzt]

Prof. Dr. Helmut Balzert
Lehrstuhl für Software-Technik
Ruhr-Universität Bochum

© Helmut Balzert 1998

LE 18
2



LE 18
3

6.2.1 Integrationsstrategien

- Wahl der Integrationsstrategie
  - Hat großen Einfluß auf die Zahl der erforderlichen Testtreiber und Platzhalter
- Testtreiber
  - Testtreiber *(driver)* wird benötigt, um eine Systemkomponente zu testen, deren Dienst nicht direkt von der Benutzungsoberfläche aufgerufen und mit Parametern versorgt werden kann.

LE 18
4

6.2.1 Integrationsstrategien

- Platzhalter *(dummies, stubs)*
  - Werden benötigt, wenn eine zu testende Systemkomponente andere Systemkomponenten aufruft, die zwar spezifiziert, aber noch nicht implementiert sind oder aus anderen Gründen nicht für den Test benutzt werden sollen
  - Simulieren die nicht verwendeten Komponenten
  - Bieten eine interaktive Schnittstelle, simulieren die verlangten Funktionen bzw. Zugriffsoperationen und bilden die Semantik nach
    - Haben jedoch eingeschränkten Leistungs- und Typumfang
- Testtreiber sind i. allg. leichter zu realisieren als Platzhalter.

LE 18 5

6.2.1 Integrationsstrategien

- **Nicht-inkrementelle Strategie**
  - Es werden alle oder eine größere Anzahl von Systemkomponenten gleichzeitig integriert
  - \+ Testtreiber und Platzhalter werden nicht benötigt
  - – Alle Systemkomponenten müssen fertiggestellt und überprüft sein, bevor die Integration erfolgen kann
  - – Fehler sind schwer zu lokalisieren, da viele Komponenten zusammenwirken
  - – Eine Testüberdeckung ist schwierig sicherzustellen, da geeignete Testfälle schwer zu konstruieren sind
  - Wegen der Nachteile ist diese Strategie für umfangreiche Systeme *nicht* geeignet.

LE 18 6

6.2.1 Integrationsstrategien

- **Inkrementelle Strategie**
  - Die Systemkomponenten werden einzeln oder in sehr kleinen Gruppen integriert
  - \+ Systemkomponenten können integriert werden, sobald sie fertiggestellt und überprüft sind
  - \+ Testfälle sind leicht konstruierbar und die Testüberdeckung ist überprüfbar, da jeweils nur eine oder wenige Systemkomponenten integriert werden
  - – U.U. werden viele Testtreiber und/oder Platzhalter benötigt.

LE 18 7

6.2.1 Integrationsstrategien

- **Testzielorientierte Strategie**
  - Ausgehend von den Testzielen werden Testfälle erstellt
    - Für die Überprüfung der Testfälle werden dann die jeweils benötigten Systemkomponenten »zusammenmontiert«
    - Beispiel für ein Testziel
      - Fertiggestellte Systemkomponenten so schnell wie möglich zu integrieren
- **Vorgehensorientierte Strategie**
  - Leitet die Integrationsreihenfolge aus der Systemarchitektur ab.

LE 18 8

6.2.1 Integrationsstrategien

- ***big bang*-Integration**
  - Alle Systemkomponenten eines Systems oder Subsystems werden gleichzeitig integriert
  - Der Integrationstest läuft dabei »unstrukturiert« und unsystematisch ab
  - Die Fehlersuche und die Konstruktion von Testfällen sind schwierig
- **Geschäftsprozeßorientiert**
  - Es werden jeweils diejenigen Systemkomponenten integriert, die von einem Geschäftsprozeß betroffen sind
    - Beispiel: Abwicklung eines Kundenauftrags von der Akquisition über den Auftrag, die Auslieferung bis hin zur Bezahlung.

LE 18
9

6.2.1 Integrationsstrategien

- **Funktionsorientiert**
  - Funktionale Testfälle werden spezifiziert und überprüft
  - Die betroffenen Systemkomponenten werden **schrittweise integriert** und getestet
- **Nach Verfügbarkeit**
  - Systemkomponenten und Subsysteme werden sofort nach Abschluß ihrer Überprüfung integriert
  - Der Integrationsprozeß wird durch die **Implementierungsreihenfolge** determiniert.

LE 18
10

6.2.1 Integrationsstrategien

- **Vorgehensorientiert**
  - Hängen von der Vorgehensweise im Entwurf und von der **Software-Architektur** ab
    - Baumstruktur
      - strukturierter Entwurf
    - Schichtenstruktur (u.U.mit Enthaltenseins-Baum)
      - modularer Entwurf
    - Schichtenstruktur (u.U. baumartige Schichtenstruktur)
      - objektorientierter Entwurf.

LE 18
11

6.2.1 Integrationsstrategien

- ***top-down*-Integration**
  - Zuerst müssen die in der Baumhierarchie oder der Schichtenstruktur **am weitesten oben** stehenden Systemkomponenten für sich überprüft werden
  - Danach werden sie schrittweise integriert
  - Da die Systemkomponenten auf Dienstleistungen anderer Systemkomponenten zugreifen, müssen die Leistungen dieser Komponenten durch **Platzhalter** simuliert werden
  - Bei einer objektorientierten Schichtenarchitektur beginnt die Integration bei der **Benutzungsoberfläche** und den Komponenten der Anwendung, die direkt mit dieser kommunizieren.

LE 18
12

6.2.1 Integrationsstrategien

+ Es entsteht frühzeitig ein Simulationsmodell, das aus der Sicht des Benutzers bereits einen Teil der Funktionen des endgültigen Systems ausführt
  - Das Simulationsmodell läßt sich noch ändern und verbessern
  - Alternativen können ausprobiert werden
+ Gezielte Prüfung der Fehlerbehandlung bei fehlerhaften Rückgabewerten unterlagerter Dienste ist möglich, da die Rückgabewerte durch Platzhalter eingegeben werden.

LE 18
13

6.2.1 Integrationsstrategien

- Eine Verzahnung von Entwurf und Implementierung ist möglich
  - Zunächst wird die oberste Schicht entworfen
  - Während die Komponenten dieser Schicht implementiert und mit Platzhaltern getestet werden, wird die nächste Schicht entworfen usw.
- Es werden Platzhalter benötigt, die aufwendig und schwierig zu erstellen sind
- Mit zunehmender Integrationstiefe wird die Erzeugung bestimmter Testsituationen in tiefer angeordneten Komponenten schwieriger
- Das Zusammenwirken von zu prüfender Software, Systemsoftware und Hardware wird spät geprüft.

LE 18
14

6.2.1 Integrationsstrategien

- *bottom-up*-Integration
  - Es werden zunächst die Systemkomponenten implementiert und getestet, die keine Dienste anderer Komponenten benötigen
    - Baumstruktur
      - Blätter des Baumes
    - Schichtenstruktur
      - Komponenten der untersten Schicht
  - Anschl. werden die nächsthöheren Komponenten, d.h. die Komponenten, die die Basiskomponenten in Anspruch nehmen, implementiert, jeweils mit den aufgerufenen Basiskomponenten integriert und zusammen getestet.

LE 18
15

6.2.1 Integrationsstrategien

- Dies wird solange fortgesetzt, bis der *»top«* erreicht ist
  - Die Reihenfolge sollte stets so gewählt werden, daß die nächst höhere Systemkomponente erst dann integriert wird, wenn alle von ihr benutzten Komponenten bereits existieren.

LE 18
16

6.2.1 Integrationsstrategien

- Es sind keine Platzhalter erforderlich
- Testbedingungen sind leichter herstellbar
- Testergebnisse können leichter interpretiert werden
- Da die Testdateneingaben über Treiber erfolgen, ist keine komplexe Zurückrechnung der Eingaben erforderlich
- Bewußte Fehleingaben zur Prüfung von Fehlerbehandlungen sind leicht möglich
- Das Zusammenwirken zwischen zu prüfender Software, Systemsoftware und Hardware wird früh geprüft.

6.2.1 Integrationsstrategien

- Es sind **Testtreiber** erforderlich
- Ein lauffähiges Gesamtsystem ist erst am Ende der Integration verfügbar
- Fehler in der Produktdefinition werden daher erst spät gefunden und können zu umfangreichen Änderungen führen
- Eine gezielte Überprüfung der Fehlerbehandlung bei fehlerhaften Rückgabewerten unterlagerter Komponenten ist kaum möglich, da die realen Komponenten benutzt werden.

- Danke!
- Aufgaben

- Diese Präsentation bzw. Teile dieser Präsentation enthalten Inhalte und Grafiken des **Lehrbuchs der Software-Technik** (Band 2) von Helmut Balzert,
Spektrum Akademischer Verlag, Heidelberg 1998