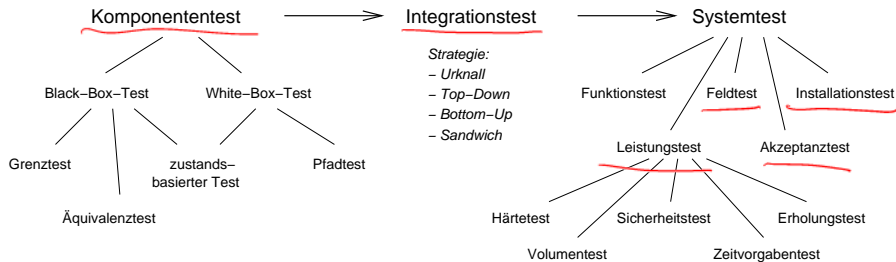


Zusammenfassung der Testarten



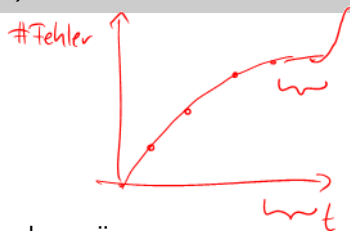
Testdokumentation: Aufzeichnung der Testaktivitäten

- Testplan: Projektplan fürs Testen
- Testfallspezifikation: Dokumentation eines jeden Testfalls
- Testvorfallbericht (Testprotokoll): Ergebnisse des Tests und Unterschiede zu erwarteten Ergebnissen
- Testübersicht: Auflistung aller Fehler (entdeckt und noch zu untersuchen)
 - Analyse und Priorisierung aller Fehler und deren Korrekturen

Testplan und Testfallspezifikation können sehr früh schon erstellt werden.

Testplan (IEEE Std. 829-1998 1998)

1. Einführung
2. Beziehung zu anderen Dokumenten
3. Systemüberblick
4. Merkmale, die getestet/nicht getestet werden müssen
5. Abnahmekriterien
6. Vorgehensweise
7. Aufhebung und Wiederaufnahme
8. Zu prüfendes Material (Hardware-/Softwareanforderungen)
9. Testfälle
10. Testzeitplan: Verantwortlichkeiten, Personalausstattung und Weiterbildungsbelange, Risiken und Schadensmöglichkeiten, Zeitplan



- Testfallbezeichner: eindeutiger Name des Testfalls; am Besten Namenskonventionen benutzen
- Testobjekte: Komponenten (und deren Merkmale), die getestet werden sollen
- Eingabespezifikationen: erwartete Eingaben
- Ausgabespezifikationen: erwartete Ausgaben
- Umgebungserfordernisse: notwendige Software- und Hardware-Plattform sowie Testrumpfe und -stümpfe
- Besondere prozedurale Anforderungen: Einschränkungen wie Zeitvorgaben, Belastung oder Eingreifen durch den Operator
- Abhängigkeiten zwischen Testfällen

- welche Merkmale wurden getestet?
- wurden sie erfüllt?
- bei Störfällen: wie können sie reproduziert werden?

→ Sammlung und Priorisierung in der Testübersicht

→ Test \neq Fehlersuche bzw. -korrektur!

Quizz

Woher kommt der englische Begriff *Bug* für Fehler?

Auflösung: *Bug* wird auf Grace Hopper zurückgeführt; bezeichnet einen Fehler, der auftrat als ein Nachtfalter mit einem Rechnerrelais in Konflikt geriet – mit dem Effekt, dass ein Programm anhielt.

2 Implementierung

- Motivation
- Architekturkonformität
- Bauhaus-Werkzeug zur Reflektionsmethode
- Programmierrichtlinien
- Anhang

- Feinentwurf eines Systems durchführen können
- Programmierrichtlinien entwerfen, kennen und einhalten
- Verständnis für Codequalität entwickeln

Der Feinentwurf ist die Brücke zwischen abstrakter Architektur und detailliertem Code.

Er legt fest:

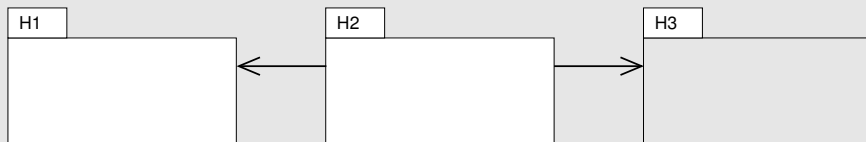
- Datenstrukturen
 - Klassendiagramme mit zusätzlichen Implementierungsdetails; (z.B. Navigierbarkeit, Implementierung von Assoziationen, Behandlung von Mehrfachvererbung oder weiteren Implementierungsklassen etc.)
- Abläufe
 - Zustandsautomaten
 - Aktivitätsdiagramme
 - Sequenzdiagramme

- Architekturbeschreibung und Quellcode sind zwei voneinander abhängige, aber separate Dokumente
 - in der Weiterentwicklung werden oft Änderungen im Quellcode gemacht, die in der Architekturbeschreibung nicht nachgezogen werden
 - Folge: Architekturbeschreibung wird obsolet
 - Planung erfolgt aber meist auf Basis der Architekturbeschreibung
 - Folge: böses Erwachen in der Umsetzung
- Reflektionsmethode von Murphy u. a. (1995, 2001) zur Synchronisation von Modulsicht und Quellcode

Reflektionsmethode (Murphy u. a. 2001)

- ① Stelle Architekturmodell auf
- ② Extrahiere Implementierungsmodell
- ③ Bilde Modelle aufeinander ab
- ④ Berechne Reflektionsmodell
- ⑤ Verfeinere/korrigiere

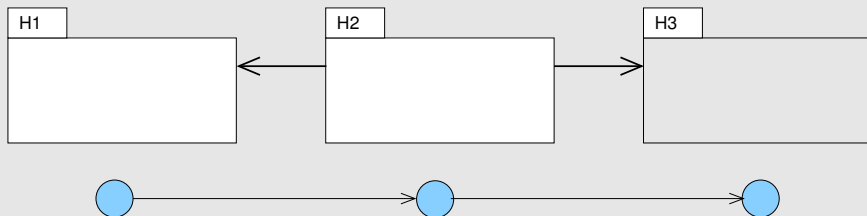
Beispiel



Reflektionsmethode (Murphy u. a. 2001)

- 1 Stelle Architekturmodell auf
- 2 **Extrahiere Implementierungsmodell**
- 3 Bilde Modelle aufeinander ab
- 4 Berechne Reflektionsmodell
- 5 Verfeinere/korrigiere

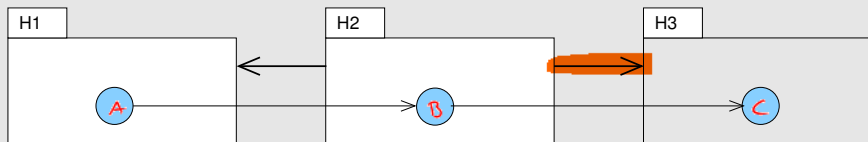
Beispiel



Reflektionsmethode (Murphy u. a. 2001)

- 1 Stelle Architekturmodell auf
- 2 Extrahiere Implementierungsmodell
- 3 **Bilde Modelle aufeinander ab**
- 4 Berechne Reflektionsmodell
- 5 Verfeinere/korrigiere

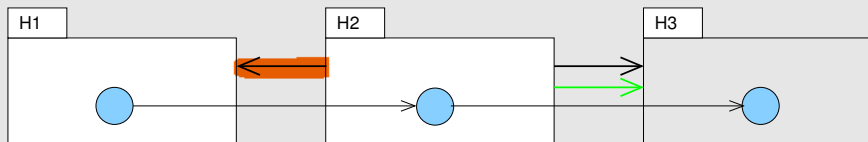
Beispiel



Reflektionsmethode (Murphy u. a. 2001)

- 1 Stelle Architekturmodell auf
- 2 Extrahiere Implementierungsmodell
- 3 Bilde Modelle aufeinander ab
- 4 **Berechne Reflektionsmodell**
- 5 Verfeinere/korrigiere

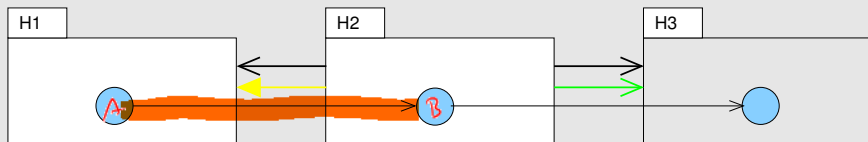
Beispiel



Reflektionsmethode (Murphy u. a. 2001)

- 1 Stelle Architekturmodell auf
- 2 Extrahiere Implementierungsmodell
- 3 Bilde Modelle aufeinander ab
- 4 **Berechne Reflektionsmodell**
- 5 Verfeinere/korrigiere

Beispiel



Reflektionsmethode (Murphy u. a. 2001)

- ① Stelle Architekturmodell auf
- ② Extrahiere Implementierungsmodell
- ③ Bilde Modelle aufeinander ab
- ④ Berechne Reflektionsmodell
- ⑤ Verfeinere/korrigiere

Beispiel

