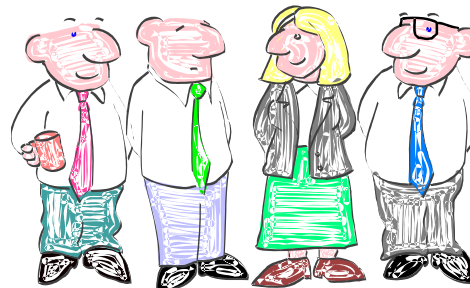


Quality Point München

Test im OO-Umfeld

Highlights aus skeptischer Sicht



Gemeinsamkeiten mit Test prozeduraler Software

- repräsentative Testdaten definieren und erzeugen
 - Zwischenergebnisse prüfen
 - fachliche Ablaufpfade verfolgen
 - Testergebnisse validieren (Ist = Soll)
-
- **d.h. fachlicher Test ist unabhängig von Soft- und Hardwareplattformen, Entwicklungsparadigmen usw.**

Unterschiede beim Modultest (Entwicklertest)

- **zu testen sind**
 - intermodulare Abhängigkeiten (Methoden, Klassen) durch stärkere Modularisierung
 - Methoden, die jeden potentiellen Zweck erfüllen können sollen -> Vielzahl möglicher Zustände
 - Methoden, die zwar prozedural getrennt aber evtl. über Objektattribute verquickt sind
 - Zustände der Objekte in Klassen und Zustandübergänge
- **zu testen sind wesentlich mehr Fälle als bisher**

Ergebnisse über Fehler in objektorientierten Systemen

- Die Anzahl der Fehler aus mangelnder Erfahrung mit der Technologie ist auffallend hoch.
- Fehler in der Analyse und im Entwurf haben eine viel größere Auswirkung als in den bisherigen Analyse- und Entwurfsmethoden.
- Es ist schwieriger Fehlerursachen aufzudecken.
- Der Code ist weniger umfangreich, d.h. die Fehlerdichte ist höher.
- Wiederverwendete Klassen weisen in der Regel weniger Fehler auf.

nach Capers Jones (600 OO-Projekte in 150 Firmen)

Hypothese vom kompetenten Entwickler

- OO-Systeme könnten im Prinzip zuverlässiger sein als herkömmliche, wenn
 - Analyse sauber war und
 - Realisierung korrekt erfolgte
- d.h. Entwickler die Technologie beherrschen
- ***provokative These: nur wenige Entwickler beherrschen die Technologie ausreichend um mit der gestiegenen intermodularen Komplexität und der Klassenvielfalt sauber umgehen zu können***

Testgegenstände

- **Methoden**
- **Klassen**
- **Klassenmenge**
- **Subsystem/Komponente**
- **Applikation/System**



Vererbung aus Sicht des Testers

- **Vererbt werden auch die Fehler**
- **Regressionstest bei Änderungen in der Basisklassen**
- **Probleme**
 - Überschreiben geerbter Funktionen (overloading) erhöht die Komplexität der Abläufe
 - Ändern von Parametertypen (overriding) verschleiert Schnittstellen
 - Vererbung bewirkt automatische Wiederverwendung evtl. auch an ungeplanten Stellen

Test objektorientierter Software – Herausforderung an Test

- **Vorteile der Verfahrungsweise bedeuten Nachteile für den Test (Aufwand, Transparenz)**
- **Bedarf an methodischem Vorgehen und Werkzeugunterstützung noch höher als bei konventionellem Vorgehen**
- **noch mehr Gewicht auf Fehlervermeidung legen z.B.**
 - **konsequente Benutzung von abgesicherten Vor- und Nachbedingungen in Methoden oder Klassen (evtl.nur für Entwicklungsphase oder Debugging einschaltbar)**

Quellenhinweise

- Testen objektorientierter Software (Sneed/Winter)