

Protokoll des Treffs am 14. Mai 2004 in Düsseldorf (PSI AG)

Teilnehmer

Michael Averstegge	
Matthias Hamburg	PSI AG (Protokollführer)
Stefan Jungmayr	Teradyne DS
Roger Müller	Universität Münster
Andreas Schönknecht	TUI InfoTec
Mario Winter	Fachhochschule Köln

Ablauf

- 1) Vortrag Roger Müller zum Thema “GlassTT – eine symbolische virtuelle Maschine zur Testfallerzeugung”
Artikel zum Vortrag siehe <http://danae.uni-muenster.de/lehre/kuchen/PUBLICATIONS/TR1103.ps.gz>
- 2) Vortrag Michael Averstegge: “Konzeption und Realisierung eines Werkzeugs für strukturelle und funktionale Bedingungs-Überdeckungstests von Java-Programmen mit OCL-Spezifikationen”
Zwischenbericht zu einer von Mario Winter betreuten Diplomarbeit
http://giserver.gi-ev.de/fachbereiche/softwaretechnik/tav/toop/Protokolle/DA_Averstegge.ppt
- 3) Themenfindung für die nächsten Treffen.

Themenfindung

Folgende Themen wurden vorgeschlagen:

1. Was sind die spezifischen Fehlerquellen von OO-Programmen?
 - Abgrenzung: nur Programmierfehler, keine Konfiguration; Eingrenzung auf Systemtest bzw. auf Teilsysteme, die vom Entwickler freigegeben sind.
 - Verschiedene Programmiersprachen: Java, C++, Ada95, C#, Smalltalk, Eiffel, ABAP Objects; nur die gemeinsamen OO-Fehlerquellen.
 - Literaturrecherche
 - Statistiken bei Capers Jones suchen
 - Klassifikation der Fehlerarten (erster Entwurf ggf. Boris Beizer, dann dynamisch weiterentwickeln)
 - Umfrage formulieren, in Foren aushängen, Firmen (QSQ-Dirk Meyerhoff, Rational, JTest, Sogeti-Stefan Gerster)
 - Auswertung mit orthogonaler Fehlerklassifikation
2. Eine empirische Validierung der Testfall-Spezifikationsmethoden
 - Was sind die spezifischen Fehlerquellen von OO-Programmen?
 - Wir haben eine Inflation an Testtechniken

- Ihre empirische Validierung fehlt, wäre aber ziemlich aufwändig
 - Experiment entwerfen im TAV-TOOP, Durchführen an Hochschulen.
 - Musterklassen in Java wählen (4 bis 5);
 - Mutationsprogramm zum Error Seeding auswählen;
 - Zu validierende Testfall-Spezifikationsmethoden auswählen;
 - Studenten: Testfälle nach den jeweiligen Methoden spezifizieren; QS durch uns.
 - Mutierte Klassen durchtesten, Trefferquoten und auch Aufwand messen
3. Testfallspezifikation und Überdeckungskriterien für die spezifischen Fehlerquellen von OO-Programmen
- Was ist bekannt an echten OO-Test-Techniken? Bücher (Binder, McGregor/Sykes, Noack: Techniken der OO-SE); Literatur ist unübersichtlich
 - Wir haben eine Inflation an Testtechniken
 - Was sind die spezifischen Fehlerquellen von OO-Programmen? Das müsste man zuerst untersuchen, subjektive Aussagen reichen nicht.
4. Integrationstest bei agiler Entwicklung:
- Wie komme ich vom Unit-Test zum vernünftigen Integrationstest?
 - Bekannte Methoden gehen das spezifische Fehlerrisiko der Integration nicht gezielt an, sondern wiederholen einfach die Testfälle der Klassen.
5. Erzeugung von Testdaten
- Mock Objekte werden in der Literatur empfohlen, aber die Praktikabilität ist fraglich;
 - Setup der Testdaten / Mock Objekte ist aufwändig;
 - Idee: Testdatenfabrik.
 - In der Praxis werden oft echte oder halbwegs entwickelte Klassen verwendet, mit dem Risiko, dass sie oft noch nicht zuverlässig sind.

Thema 1 wurde als nächstes Schwerpunktthema für den Arbeitskreis ausgewählt. Die typischen OO-Fehler sollen dann als Input für die Bearbeitung von Thema 2 dienen.

Aktionen bis zum nächsten Treffen

Literaturrecherche zu häufigen OO-Fehlern parallel starten; (alle)

Kommunikationsforum feslegen (Stefan Jungmayr)

- HTML-Seite auf unserer GI-Seite
- Eigenen Wiki-Bereich einrichten, mit Zugangsbeschränkung, evtl. FH Köln,
- Public Wiki-Bereich benutzen (Dt. SW-Entwicklerforum?)

Ergebnisse der Literaturrecherche in das gewählte Kommunikationsforum einstellen als Input für das TOOP-Treffen bei TAV 21. (alle)

Nächstes Treffen des Arbeitskreises

Das nächste Treffen findet am 17./18. Juni in Berlin im Rahmen des TAV-21-Treffens statt.

Weitere Informationen über den Arbeitskreis:

<http://giserver.gi-ev.de/fachbereiche/softwaretechnik/tav/toop/>