

GI-2.1.7. TAV AK-TOOP

Protokoll des Treffs am 11.6.99 in Darmstadt

Anwesende

Frank Berghäuser, PMS-Micado
Michael Brunner, TU Darmstadt
Dirk Ehmke, Software AG (bis 12:30 Uhr)
Falk Fraikin, TU Darmstadt
Wolfgang Henapl, TU Darmstadt (bis 12:30 Uhr)
Stefan Jungmayr, FernUniversität Hagen
Franz Maurer, Softlab GmbH
Sven Nordhoff, DTK GmbH
Harry Sneed, SES - Software-Engineering Service GmbH
Andreas Spillner, Hochschule Bremen
Mario Winter, FernUniversität Hagen

Ablauf

1030 - 1045 Uhr	Begrüßung
1045 - 1145 Uhr	Vortrag Michael Brunner Use Cases zur Generierung von Systemtests (JAVA)
1145 - 12:45 Uhr	Diskussion
1330 - 1430 Uhr	Werkzeug-Fragebogen — Wie Weiter?
1430 - 1500 Uhr	Diskussion: Die Top 10 Liste der OOT-Projektkiller

Themen

Vortrag Michael Brunner: Use Cases zur Generierung von Systemtests (JAVA)

Michael stellte seinen Use Case-basierten Ansatz zur Generierung von Systemtests für JAVA-Programme vor. Eckpfeiler seiner Methode sind die Generierung von Methodestubs auf der Basis exemplarischer Benutzeraktionen, der zustandsbasierte Test und die Benutzung wiederverwendbarer Komponenten (und Tests) auf der Basis von JavaBeans.

In der Diskussion wurde es einerseits begrüßt, dass die Einbeziehung der Benutzungsschnittstelle schon in der Anforderungsermittlung sowohl bei der Validierung der Anforderungsspezifikation als auch bei der frühen Ermittlung von Systemtests hilfreich ist. Andererseits wurde der hohe Detaillierungsgrad bemängelt (Harry Sneed: In der Industrie werden Modelle wie z.B. Objektkonstellationen und Sequenzdiagramme selten eingesetzt).

Übereinstimmend wurden Use Cases als adäquate Basis der Anforderungsermittlung gesehen — allerdings mit der Einschränkung, das technisch nicht versierte Benutzer bzw. Stakeholder besser mit konkreten Abläufen denn mit „statischen“ Use Case Diagrammen und „generischen“ Beschreibungen zurecht kommen.

Als Validierungsverfahren für die Anforderungsspezifikation und den Entwurf wurde noch einmal das Review diskutiert. Es kristallisierte sich die Erfahrung heraus, das Reviews eher im Rahmen der Konzeptualisierung „ernst“ genommen werden, Code Reviews nach Checklisten mit anschließenden Sitzungen jedoch schnell „langweilig“ werden und dementsprechend nicht mehr effizient sind.

Werkzeug-Fragebogen — Wie Weiter?

Wir wollen den Fragebogen für Werkzeughersteller ab Oktober noch einmal publizieren. Dazu sollen im wesentlichen die Anmerkungen zu den einzelnen Fragen erweitert und überarbeitet werden, um eine größere Verständlichkeit zu erzielen. Bereit erklärt zur Mitarbeit haben sich:

Michael Brunner	(Fragekomplex F)
Stefan Jungmayr	(Fragekomplex A7-A10)
Franz Maurer	(Fragekomplex A1-A6)
Sven Nordhoff	(Fragekomplex C)
Andreas Spillner	(Fragekomplex E)
Mario Winter	(Fragekomplex B, D)

Die Promotion des Fragebogens wollen wir wieder in die Hände von Stephanie Ulrich (Journal) und Stefan Jungmayr (Internet) legen.

An einem Artikel zum Fragebogen arbeiten zur Zeit Stefan Jungmayr, Andreas Spillner und Mario Winter — sobald eine gewisse Reife erreicht ist, soll wieder eine Review-Runde mit anschließender eMail-Diskussion geführt werden.

AK TOOP — Wie Weiter?

Wir haben beschlossen, zunächst die Homepage des AK TOOP attraktiver zu gestalten. Diese soll nach Themengebieten unterteilt werden und dem Besucher möglichst übersichtlich Informationen zu Werkzeugen, Seminaren, Büchern, WWW-Links etc. anbieten. Vorschläge bitte an Mario Winter. Im Anschluss daran haben wir begonnen, eine Liste mit dem Arbeitstitel „OO-Test: How and how not to start“ anzufertigen, die zunächst in einer internen eMail-Diskussion verfeinert und ausgeschmückt werden soll.

In der Abschlussdiskussion stellten wir die folgende Liste von „Attitüden“, Verhaltensweisen und Annahmen zusammen, die nach unserer Erfahrung ein OO-Testprojekt gefährden.

Die Liste der Top 10 OO-Testprojektkiller

1. Keine Komponenten bzw. „unabhängigen“ Teilsysteme bilden
2. Mitarbeiter nicht schulen
3. Ohne Architektur implementieren
4. OO-Test nicht besonders vorbereiten, da einfacher als „normaler“ Test
5. Sofort Werkzeuge kaufen
6. Tests nur auf Entwurfsebene ableiten
7. OO-Migration direkt und ad-hoc angehen, ist einfach
8. Kein „Reworking“ einplanen bzw. durchführen
9. Tiefe Vererbungshierarchien konstruieren
10. Die Anwendung eng gekoppelt auf eine Bibliothek aufbauen

Bezüglich der Reihenfolge darf noch gewählt werden! Auch wären Beispiele und erläuternde Texte für die einzelnen Punkte sicher hilfreich.

Nächster Treff

Das nächste Treffen des AK-TOOP findet im Rahmen des 14. Treffens der Fachgruppe am 28. und 29. Oktober 1999 bei der Robert Bosch GmbH in Stuttgart statt. Themenvorschläge (neben der TOP-10 Liste) werden erbeten an Mario Winter.