

GI-2.1.7. TAV AK-TOOP

Protokoll des Treffs am 16. Februar 2001 in Elmshorn

<http://www.informatik.fernuni-hagen.de/import/pi3/GI/akOOT.html>

Anwesende

Michael Brunner, Dresdner Bank,
Falk Fraikin, TU Darmstadt,
Bernhard Gehrke, Software Quality Engineering
Thomas Leonhardt, TU Darmstadt,
Ute Pelkmann, SCOPE GmbH,
Dehla Sokenou, TU Berlin,
Andreas Spillner, Hochschule Bremen,
Mario Winter, FernUniversität Hagen.

Ablauf

11⁰⁰ - 12³⁰ Uhr Kurzvorstellung der Teilnehmer, Themenfindung
13¹⁵ - 15¹⁵ Uhr Diskussion/Brainstorming: Test von Komponenten.

Test von Komponenten

Zunächst wurden grundsätzliche Merkmale und Unterschiede von Klassen und Komponenten erörtert.

Komponente vs. Klasse

Nach [Szyperski1998] hat eine Komponente folgende Merkmale:

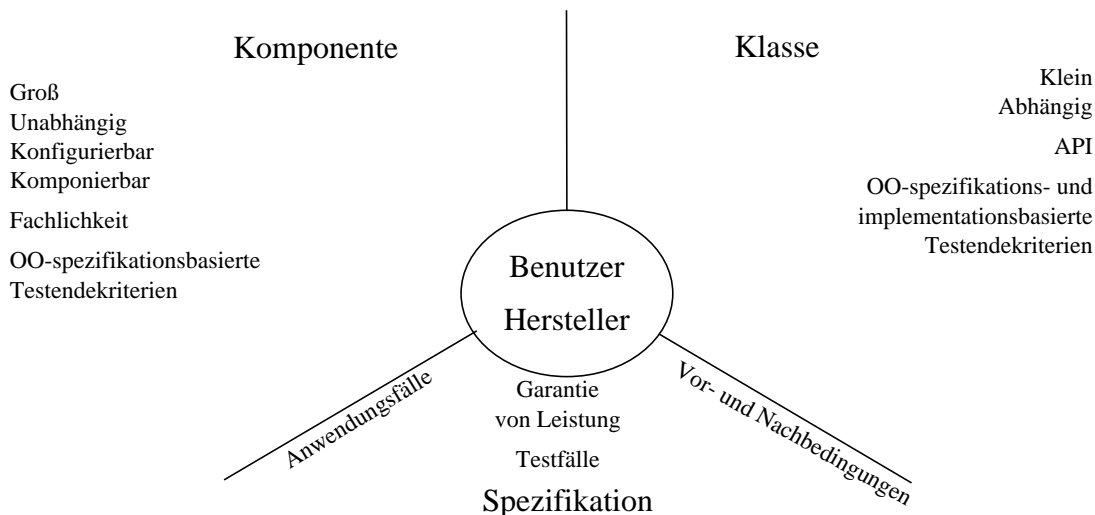
- Einzel auslieferbar
- In binärer, ausführbarer Form vorliegend
- Mit anderen Komponenten kombinierbar
- Eher "groß" (relativ zu Klassen)
- Konfigurierbar
- (Nur) in einer geeigneten Umgebung einzusetzen
- Dokumentation beinhaltet auch fachliches Wissen aus der Anwendungsdomäne
- Keine Vererbung
- Tests (durch Benutzer der Komponente) meist nur in Form von Black Box Tests

Eine Klasse dagegen hat als charakteristische Eigenschaften:

- Eher "klein" (relativ zu Komponenten)
- Dokumentation beschränkt sich oft eher auf eine "API-Beschreibung"
- Vererbung
- Auch White Box Tests (da der Source Code vorliegt)

Testfälle/Daten

Ebenfalls wurde der Nutzen der Auslieferung von Komponenten in Kombination mit zugehörigen Testfällen diskutiert. Die Testfälle sollen (und können) dabei nicht die Spezifikation der Komponente ersetzen. Vielmehr liegt ihr Nutzen einerseits darin, dass der Nutzer einer Kom-



ponente nachvollziehen kann, für welche Fälle der Hersteller eine Verwendung der Komponente in Betracht gezogen hat. Andererseits werden dem Benutzer durch die Testfälle Code-Beispiele an die Hand gegeben, welche sozusagen als "Mehrwert" die programmiertechnische Verwendung der Komponente demonstrieren.

Im Open Source Bereich gibt es einige Beispiele für Projekte, bei denen Testfälle in Form von JUnit-Klassen mit ausgeliefert werden. Es seien zwei Beispiele genannt:

Ant (eine Art "make" Ersatz), s.a.: <http://jakarta.apache.org/ant/>
 JDOM (API zum Manipulieren von XML-Strukturen), s.a.: <http://www.jdom.org>.

Testenkriterien

In Bezug auf Komponenten sind insbesondere auch mögliche Testenkriterien interessant. Hier ergaben sich folgende Ergebnisse:

- Kontrollflussorientierte Verfahren sind zwar der State of the Art in kommerziellen (Test-) Werkzeugen, aber für objektorientierte Programme und insbesondere Komponenten eher ungeeignet
- OO-Metriken haben sich noch nicht in der Praxis bewährt
- Datenflussorientierte Verfahren sind unhandlich (komplex) und auch keineswegs immer sachdienlich
- Aus der Praxis kommen diesbezüglich (noch?) keine Impulse / Anforderungen
- Für Komponenten braucht man spezifikationsbasierte Testenkriterien
 (ToDo: ausfindig machen bis zum nächsten Treffen ;-)

Literatur: [Szyperski1998] Clemens Szyperski: Component Software, acm Press, Addison Wesley, Reading, Mass. 1998

Nächstes Treffen des Arbeitskreises

Das nächste Treffen des AK-TOOP findet am 18. Mai 2001 bei der Dresdner Bank in Frankfurt/Main statt. Themenvorschläge:

- Spezifikationsbasierte Testenkriterien für Komponenten

Bitte per eMail bis zum 01. Mai 2001 bei Michael Brunner von der Dresdner Bank anmelden (Michael.Brunner@dresdner-bank.com, ggf. inkl. weiteren Themenvorschlägen).