

Lesbar auf den ersten Blick!

Wertvolle Unittests durch DSL orientiertes API Design
Ulf Müller, Hewlett-Packard GmbH
(Dauer: 90 min)

Durch den Erfolg agiler Vorgehensweisen in den vergangenen Jahren haben viele Softwareentwickler Unittests in ihre tägliche Praxis übernommen. Es zeigt sich jedoch, daß auch eine auf den ersten Blick einfache Disziplin wie das Unittesting einiges an Erahrung braucht, damit die resultierenden Tests ihren Wert über die Zeit bewahren. In der Praxis entstehen häufig Tests, deren ursprünglicher Zweck nur noch schwer zu erkennen ist. Dies führt dazu, daß Teammitglieder, die mit den Tests ihrer Kollegen nicht vertraut sind, Anpassungen und Wartungsarbeiten halbherzig und fehlerhaft durchführen und somit die Situation weiter verschärfen. Man wird beobachten können, daß der Wartungsaufwand für Unittests steigt, die Testcoverage sinkt und sich im Team ein latenter Widerstand gegen Unittesting entwickelt.

Erstrebenswert sind Tests, denen ihr Zweck auf den ersten Blick zu entnehmen ist. Nur solche Tests werden auch in Streßsituationen verstanden und können im Sinne ihres ursprünglichen Zwecks erweitert und gepflegt werden. Je schwerer Tests zu verstehen sind, desto größer die Wahrscheinlichkeit, daß sie im Fehlerfall und unter Streß mit den oben genannten Folgen einfach „stumm geschaltet“ werden.

Mein Vortrag wird ein paar einfache Testprinzipien wiederholen (u.a. „One Concern per Test“, „One Condition per Test“, „Minimized Test Overlap“, etc), und zeigen wie deren Einhaltung durch die Verwendung von APIs, die sich an DSLs (Domain-Specific Languages) orientieren unterstützt wird. Dazu werde ich einen kurzen Überblick über existierende Werkzeuge geben (z.B. jMock, FEST –Assert, Hamcrest) . Hauptsächlich aber werde ich zeigen, wie man durch die Verwendung bestehender Sprachmittel (static Imports, Varargs) und einfacher Programmier Techniken (Method Chaining) eigene embedded DSLs in Java erstellen kann. In der Praxis haben sich solch einfachen DSLs als ausgezeichnetes Mittel zum Erzeugen und zur Konfiguration von Eingangsdaten für Tests herausgestellt. Abgesehen davon sind sie aber auch auf andere Bereiche anwendbar. Insbesondere Tests, für die signifikante Mengen an Daten erzeugt werden müssen, profitieren erheblich durch die Verwendung einer embedded DSL, da diese vieles des „Grundrauschens“ der Datenerzeugung beseitigt, und damit die Sicht auf die wesentlichen Aspekte des Tests öffnet. Weiterhin werden durch eine DSL Ausdrucksmittel geschaffen, die den Zweck der erzeugten Daten klarer vermitteln, als dies mit herkömmlichen Mittel geschieht. Zu guter Letzt stellt eine DSL zur Datenerzeugung ein Framework dar, welches dem Team viele wiederkehrende Aufgaben abnehmen kann.

Übung

Die Workshop-Übung wird aus einem oder mehreren vorbereiteten „schlechten“ Unittests bestehen. Diese Tests werden als Demonstration dienen, wie schwer es

mitunter ist, Tests ihren wirklichen Zweck anzusehen. Die Aufgabe der Teilnehmer wird darin bestehen, durch den Einsatz der vorgestellten Techniken und Werkzeuge die Tests so zu Refaktorisieren, daß ihr ursprünglicher Zweck Stück für Stück wieder sichtbar wird. Sie werden weiterhin erkennen können, daß die Tests gegen die zu Beginn des Vortrags wiederholten Prinzipien verstoßen. Mit dieser Erkenntnis werden sie die Tests weiter vereinfachen können, so daß am Ende Tests vorliegen, die auch von anderen auf den ersten Blick verstanden und im Sinne ihres eigentlichen Zwecks erweitert und gepflegt werden können.

Zu meiner Person

Seit dem Ende meines Studiums an der Fachhochschule Osnabrück vor knapp 11 Jahren arbeite ich bei der Hewlett-Packard GmbH als Software-Designer. Seit rund 7-8 Jahren arbeite ich fast ausschliesslich in „Java lastigen“ Projekten. Eines meiner speziellen Interessen ist seitdem Unittesting. Die so erworbenen Kenntnisse gebe ich intern an die Kollegen anderer Teams sowie an neue Kollegen und unsere Studenten weiter. Eine Vorlesung, die ich Ende vergangenen Jahres an der BA in Stuttgart gehalten habe motivierte mich den oben beschriebenen Vortrag für die Test-Tage einzureichen.