



□ Prof. Dr. Ina Schieferdecker

(E-Mail: ina.schieferdecker@fokus.fraunhofer.de)

beschäftigt sich mit Fragen der modellbasierten Softwareentwicklung, der Analyse, des Testens und Bewertung softwareintensiver Systeme und der Automatisierung und Optimierung von Prozessen zur Software-[Weiter-]-Entwicklung und Qualitätssicherung. An der Freien Universität Berlin leitet sie das Fachgebiet „Modellbasierte Entwicklung und Qualitätssicherung Software-basierter Systeme“ und ist am Fraunhofer Institut FOKUS, Berlin, Leiterin des Kompetenzzentrums MOTION, das sich mit Modellierungs- und Testansätzen und der Automatisierung und Optimierung von Software-Entwicklungsprozessen beschäftigt.

objektspektrum themenspecial: testing

Beim Testen Neues wagen

Zum 15. Jubiläum des Arbeitskreises Software-Qualität und -Fortbildung (ASQF) wurde am 24. September 2011 erstmalig der deutsche Preis für Software-Qualität an Harry M. Sneed für sein Lebenswerk verliehen. Harry M. Sneed bearbeitet seit 35 Jahren das Thema der Software-Qualität, nicht nur, aber immer wieder auch als Software-Tester. Dabei hat er bereits Ende der 70er und mittlerweile gut 20 Testwerkzeuge konzipiert, umgesetzt und angewendet, eine Leidenschaft, die ihm noch heute immer wieder neue Einsichten in Testmethoden gibt, mit allen nötigen als auch möglichen Erweiterungen.

Testen und Testautomatisierung gehen augenscheinlich Hand in Hand, zu mindestens für eine führende Persönlichkeit des Software-Testens wie ihn. Doch lassen Sie uns in die aktuelle Umfrage zum Thema „Softwaretest in der Praxis“ des German Testing Boards (GTB) und anderen unter der Leitung von Andreas Spillner, Karin Vosseberg, Mario Winter und Peter Haberl schauen: zwar hat sich der Berufsstand des Software-Testers etabliert – so wurde in der Zwischenzeit das 25.000 Software-Tester-Zertifikat in Deutschland vergeben – und dennoch ist Testautomatisierung noch nicht prominent vertreten. Es wird zwar auf verschiedenen Stufen funktionsorientiert und anforderungsbasiert getestet, jedoch hat sich weder eine werkzeuggestützte Erfassung von Testfällen noch deren werkzeuggestützte Durchführung etabliert. Ein neben den positiven Indikatoren der Umfrage erschreckendes Ergebnis – hat uns doch beispielsweise Harry M. Sneed bereits Ende der 70er gezeigt, wie Testautomatisierung angegangen werden kann.

Komplimentiert wird die Umfrage des GTB durch den Artikel von Murat Aksu und Stefan Gerstner zum World Quality Report 2011–2012. Dieser bescheinigt u. a., dass Budgets für Qualitätssicherung weiterhin unter Druck bleiben und dass das Potential

von Automatisierungswerkzeugen beim Testen noch nicht ausgereizt ist. Zwar liegen deutsche Unternehmen beim Einsatz automatisierter Testverfahren 13 % über dem internationalen Durchschnitt, jedoch wird Automatisierung noch nicht voll ausgeschöpft. Aufgrund kürzerer Produkteinführungszeit und geringerer Kosten wird der Automatisierungsgrad beim Software-Testen weiter zunehmen.

So nennt der World Quality Report aktuelle Trends und neue Anforderungen beim Software-Testen, die in dieser Ausgabe des Online Issue Software-Testen vertieft werden. Beispielsweise geht es um das Testen in der Cloud, um das Testen der Cloud selber als auch um das Testen mit der Cloud. Erste Ergebnisse finden Sie dazu im Beitrag von Wolfram Kusterer zu Möglichkeiten der Virtualisierung des Anwendungsverhaltens, um eine höhere Flexibilität und bessere Verfügbarkeit von Testumgebungen mit Hilfe von Virtualisierungstechnologien zu erzielen. Die Virtualisierung von Anwendungsszenarien verspricht eine Vereinfachung von Testumgebungen.

Eine andere virtualisierungsbasierte Perspektive auf die Bereitstellung von Testumgebungen nimmt Nico Orschel ein. Für agile Entwicklungsprozesse sind die permanente Verfügbarkeit, Weiterent-

wicklung als auch Separierung von Testumgebungen wesentlich, was durch die Einbettung virtualisierter Testumgebungen und automatisierter Tests in kontinuierliche, agile Entwicklungsprozesse erreicht werden kann. Ein baukastenartiges Vorgehen für Testumgebungen, die die für die Tests nötigen virtualisierten Software-Komponenten in verschiedenen Zielumgebungen bereithalten, wird anhand des Microsoft Team Foundation Server und Testmanagers erläutert.

Nicht nur für reine Software, sondern auch für Software-basierte Systeme wie Steuergeräte oder andere eingebettete Systeme sind kontinuierliche Entwicklungsprozesse wesentlich und sind bestenfalls mit Testumgebungen für automatisierte Tests zu unterstützen. Martin Kropp und Anja Kellner beschreiben eine solche Testlösung basierend auf den Ranorex und Open Source Jenkins Werkzeugen zum Testen von Mess- und Sensorkomponenten, die für die Steuerung und Überwachung von Industrieanlagen genutzt werden. Für das Testen der Software in verschiedenen Sprachen und Betriebssystemen werden vorkonfigurierte virtuelle Maschinen genutzt, die die jeweiligen Zielsysteme emulieren.

So wird, wie in den drei Artikeln beschrieben, bereits heute das Testen mit

der Cloud unter Nutzung von Virtualisierungskonzepten für Anwendungen, Softwarekomponenten und/oder Zielsysteme erfolgreich eingesetzt und steht für eine breitere Nutzung zur Verfügung – wagen Sie es. Tests der Cloud und Tests in der Cloud verfolgen andere Ziele und werden hoffentlich in einer der kommenden Ausgaben des Online Issues Gegenstand von Artikeln werden. Mit der zunehmenden Nutzung von (öffentlichen, privaten oder hybriden) Clouds müssen auch diese Technologien selber in das Zentrum des Testens rücken – lassen Sie uns auch dieses angehen.

Zurück zu Agilität in Entwicklungsprozessen – einem Paradigma, das derzeit in bald jeder Diskussion zu Entwicklungsprozessen angeführt wird und durchaus mit überzeugenden Ergebnissen für kleine bis mittelgroße Softwareentwicklungsprojekte aufwarten kann. Dabei ist einerseits die Rolle des Testens beispielsweise entlang von Test-First- oder Test-Driven-Ansätzen betont, gleichzeitig aber die Differenzierung in Spezialisierungen wie den Tester oder Anforderungsentwickler unterbelichtet. Insofern gilt es die jeweiligen notwendigen Rollen in agilen Entwicklungsprozessen zu stärken.

Demgemäß beschreibt Andreas Becker ein strukturiertes Vorgehen zur Entwicklung von Akzeptanzkriterien im klassischen als auch agilen Testumfeld. Erst mit einer präzisen Festlegung von Akzeptanzkriterien wird eine zielgerichtete Softwareentwicklung als auch deren Prüfung durch statische oder dynamische Tests ermöglicht. Die Zeiten von Anforderungen an eine zu entwickelnde Software, die widersprüchlich, unvollständig, nicht nachweisbar oder nicht messbar sind, oder aber von Tests, die nicht stimmig bezüglich der Anforderungen sind oder aber diese nicht oder nur teilweise abdecken, sollten damit vorbei sein. Zwar sind schlüsselwortbasierte Ansätze für die Schärfung von Anforderungen bereits bekannt, doch zeigt der hier aufgezeigte Weg zu Akzeptanzkriterien und Testfällen den Brückenschlag zum Testen auf.

Wie Tester in ihren Rollen als Test Manager, Test Analyst oder Technical Tester den wachsenden Anforderungen, den zunehmend komplexeren Systemen als auch Testtechnologien und Werkzeugen, den umfangreicher werdenden Testprozessen mit ihren Verzahnungen in die Entwicklungsprozesse, in das Projektmanagement und die Qualitätssicherung hinein

gerecht werden können, beschreiben die Lehrpläne der Software-Tester-Ausbildung des International Software Testing Qualifications Boards (ISTQB), das in Deutschland durch das German Testing Board (GTB) repräsentiert wird. Einen Überblick dazu gibt der Artikel von Tilo Linz, Imbus. Sollten Sie sich noch nicht der Prüfung auf Foundation oder Advanced Level gestellt haben, wagen Sie es – in zunehmendem Maß wird das Software-Tester-Zertifikat als Voraussetzung bei Job-Angeboten oder Ausschreibungen genannt.

Und ja, es gibt sie: Rahmenbedingungen, bei denen manuelle Tests doch die bessere Wahl im Vergleich zu automatisierten Tests sind. Dennoch sollte man sich auch in solchen Fällen eine Werkzeugunterstützung zu Hilfe nehmen. Wie im Artikel von Claudia Simsek-Graf und Harald Winter beschrieben, sind auch manuelle Tests zu entwerfen, zu planen, zu dokumentieren, etc. Ein Werkzeug wie TEQtest kann helfen, Prozesse rund um manuelle Tests nachhaltiger zu gestalten und Testressourcen zu optimieren.

Der World Quality Report von Capgemini geht ebenso auf die wachsende Bedeutung von Sicherheitstests auch für Applikationen ein. In seinem Artikel berichtet Tobias Kutzer von der Absicherung von Web-Auftritten mittels des AppScan Werkzeuges. Dieses prüft mittels statischer und dynamischer Analysen u. a. die Absicherung von Kundeninformationen, die vom Payment Card Industry Data Security Standard (OCIDSS) festgelegt wurde. So können mögliche Angriffe auf Web-Anwendungen gezielt geprüft und verhindert werden.

Auch wenn spezifikationsbasiertes Testen einen hohen Stellenwert hat, so bietet strukturbasiertes Testen komplementäre Verfahren, die das Testen einer Software abrunden und von einigen Standards in sicherheitskritischen Bereichen auch gefordert werden. Einen Ansatz zur Automatisierung von Strukturtests für in der Programmiersprache C entwickelte Software beschreibt Kiran Lakhota. Dieser Ansatz nutzt suchbasierte Verfahren zur Ableitung von Testdaten für eine möglichst optimierte Abdeckung des Codes. Es wird insbesondere das frei verfügbare, prototypische Werkzeug AUSTIN vorgestellt, das u.a. in Industrieprojekten von Daimler genutzt wird.

Auch wenn die GTB-Umfrage zum „Software-Test in der Praxis“ aufgezeigt hat, dass modell-basiertes Testen (MBT) noch nicht umfänglich in der industriellen Testerwelt angekommen ist, so nehmen

doch die Berichte über erfolgreiche Anwendungen von MBT zu. So auch in dieser Ausgabe: Florian Prester und Gerhard Baier beschreiben wesentliche Verbesserungen beim Testen mittels eines modellbasierten Testansatzes. Es ist gelungen, bei Siemens Healthcare reibungslos MBT einzuführen, wobei der angepasste, durch eine Werkzeugkette unterstützte Testprozess durch Beteiligungs-, Kommunikations- und Pilotierungskonzepte flankiert wurde. Im Kern der Werkzeugkette steht das Werkzeug MBTsuite, das ebenso wie das vorher beschriebene Werkzeug AUSTIN suchbasierte Verfahren zur Ableitung von Testfällen nutzt. Bereits in der Einführung von MBT konnte eine analoge Testgüte und Testabdeckung bei gleichen Kosten wie beim traditionellen Vorgehen erreicht werden, so dass mit der zunehmenden Verstärkung des Ansatzes Kostenersparungen erwartet werden.

Vergleichbar positive Ergebnisse zum modell-basierten Testen werden im Artikel von Marc-Florian Wendland, Markus Schacher und mir für das Testen von Steuerungssoftware in der Bahn Technik präsentiert. Dabei wird das UML Testing Profile (UTP) der Object Management Group (OMG) genutzt, das von den Autoren maßgeblich entwickelt und weiterentwickelt wurde. Die Überarbeitung des UTP wurde im Sommer 2011 bei der OMG als Standard verabschiedet und wird durch eine steigende Zahl von Werkzeugen unterstützt.

Und wenn Sie die präsentierten Ergebnisse neugierig machen und zu neuen Ufern beim Testen aufbrechen lassen, so wagen Sie gegebenenfalls ebenso, andere an Ihren Ergebnissen teilhaben zu lassen. Die Weiterentwicklung der Testmethoden, -prozesse und -werkzeuge ist auch auf Ihre Ergebnisse und Ihr Feedback angewiesen: was hat sich bewährt, was hat sich nicht bewährt, wo gibt es Bedarf, wo weniger, was wird empfohlen, wovon wird abgeraten? Lassen Sie es uns und andere wissen: diese Themenreihe zum Software-Testen lebt insbesondere von den Erfahrungsberichten der Tester und den vielfältigen Sichtweisen auf die Materie. An dieser Stelle ein herzliches Dankeschön den Autoren, den Editoren und dem Verlag, ohne die die Themenreihe nicht möglich wäre! Verbunden mit einer Einladung an alle Leserinnen und Leser, selber als Autorin oder Autor aktiv zu werden. Bis dahin verbleibe ich mit besten Tester-Grüßen,
Ihre Ina Schieferdecker.