



©Fraunhofer FOKUS/ Matthias Heyde

□ Prof. Dr. Ina Schieferdecker

(ina.schieferdecker@fokus.fraunhofer.de)

beschäftigt sich mit Fragen der modellbasierten Softwareentwicklung, der Analyse, dem Testen und der Bewertung softwareintensiver Systeme sowie mit der Automatisierung und Optimierung von Prozessen zur Software-(Weiter-)Entwicklung und Qualitätssicherung. Als Professorin im Fachbereich Informatik an der Freien Universität Berlin leitet sie die Fachgruppe „Modellbasierte Entwicklung und Qualitätssicherung softwarebasierter Systeme“ und ist am Fraunhofer-Institut FOKUS, Berlin, Leiterin des Kompetenzzentrums MOTION, das sich mit der (modellbasierten) Softwareentwicklung, dem (modellbasierten) Testen, der Zertifizierung und den dazu benötigten Prozessen und Werkzeugen beschäftigt. Sie engagiert sich beim GTB u. a. für Vorlesungen zum Thema Software Testing an deutschen Universitäten und Hochschulen und weltweit.

objektspektrum themenspecial:

Testing – neue Konzepte, Strategien, Zukunftsaussichten

Testautomatisierung? Ja, bitte!

Von der Automatisierung von Testaufgaben durch softwarebasierte Werkzeuge wird bereits seit der Softwarekrise in den 1960er Jahren gesprochen. So formulierte z. B. Herbert D. Benington bei seinem Vortrag zum Symposium „Advanced programming methods for digital computers“ des Navy Mathematical Computing Advisory Panels und des Office of Naval Research im Juni 1956 unter anderem: „After coding, each component subprogram is parameter tested on the machine by itself. This testing phase uses an environment that simulates pertinent portions of the system program. Each test performed during this phase is documented in a set of test specifications that detail the environment used and the outputs obtained (vgl. [Ben83]).“

Es folgten Gehversuche in der Industrie mit unterschiedlichem Erfolg sowie Forschungsarbeiten, die schlussendlich in der Konsolidierung von Testautomatisierungsansätzen bei IEEE (vgl. [Iee10]) und ETSI/ITU (vgl. [Ets13-b]) für die Testausführungsautomatisierung bis hin zur automatisierten Generierung von Tests bei OMG (vgl. [Omg13]) und ETSI (vgl. [Ets13-a], [Ets11]) resultierten.

Nun scheint jedoch der Durchbruch bevorzustehen: Wie auch IDC im „Worldwide Automated Software Quality 2012–2016 Forecast“ (vgl. [Idc12], [Idc13]) sagen viele Experten voraus, dass die Qualitätssicherung mobiler Applikationen, die Nutzung agiler Entwicklungsprozesse und Testdienstleistungen aus der Cloud einen entscheidenden Unterschied bei der Einführung und Verbreitung von Testautomatisierungslösungen machen werden. Im Forecast von IDC für den Zeitraum 2012

bis 2016 wird von einer jährlichen Wachstumsrate des Marktsegments Automated Software Quality (ASQ) von 6,6 % auf jährlich 2,2 Milliarden US-Dollar ausgegangen (im Vergleich dazu lag die jährliche Wachstumsrate des ASQ-Segments im Jahr 2010 bei 4,6 %). „In a global economy that remained volatile in 2011, but with dramatically increased complexity for multimodal software development and deployment with constrained resources, we saw ongoing investment and demand for ASQ solutions“, sagt dazu Melinda Ballou, Programmdirektorin im Bereich Application Life-Cycle Management bei IDC.

Sicher, Softwaretests sind nur eine, wenn auch die wichtigste Methode (automatisierter) Softwarequalitätssicherung, jedoch konzentriert sich das vorliegende Themenspecial „Testing“ der OBJEKTSpektrum auf diesen Bereich. Wie man sieht, gibt es auch in Deutschland zu dem Thema viel zu

sagen: fünf der elf Artikel der Ausgabe beschäftigen sich mit Testausführungsautomatisierung im Allgemeinen, drei Artikel diskutieren Testdienste vornehmlich über die Cloud und zwei weitere Artikel diskutieren Ansätze zur Automatisierung der Testfallgenerierung.

Aber lassen Sie uns zuerst den ersten Artikel hervorheben: Er wurde bei der obigen Aufzählung nicht mitgerechnet, da er sich mit Fragen der Ausbildung zum professionellen Softwaretester und dem Nachweis der benötigten Fähigkeiten und Expertisen beschäftigt. Von Frank Simon und Graham Bath vom German Testing Board (GTB) wird in „Was Autofahrer von Testern lernen können: Der neue Advanced Level“ das Ausbildungsschema Software Testing des ISTQB und insbesondere der neue Advanced Level-Lehrplan in Deutschland (vgl. [Gtb13]) vorgestellt. Der in der Online-Themenspecial-Reihe Testing bereits dritte Artikel zum

Thema Software Testing als Profession zeigt die Relevanz und die Fortschritte bei der Etablierung und Verbreitung des Berufsbildes des Softwaretesters mit Hilfe des Certified Software Tester-Schemas. Der nun dritte Artikel zu diesem Thema zeigt die Relevanz, aber auch die Fortschritte bei der Etablierung und Verbreitung des Berufsbildes des Softwaretesters.

Zurück zur Testautomatisierung. Nun könnte ich meine Ausführungen im Anschluss an den Advanced Level mit dem Software Test Expert Level zur Testautomatisierung fortsetzen, der auch unter Mitarbeit des GTB in Vorbereitung ist, aber lassen Sie uns direkt auf die Inhalte und Ergebnisse der vorliegenden Artikel des Themenspecials eingehen.

Den Reigen zur Testausführungsautomatisierung eröffnet Johannes Hochrainer mit „Die Kunst der Testautomatisierung über das GUI“. Nach einer Diskussion grundlegender Schwierigkeiten der GUI-Testautomatisierung, wie z. B. der Pflege der GUI-Tests oder aber deren Robustheit, werden verschiedene Ansätze zur Realisierung von GUI-Tests vom Capture & Replay bis hin zu empfohlenen keywordgetriebenen und workflowbasierten Methoden vorgestellt. Neben Vor- und Nachteilen der jeweiligen Ansätze werden Empfehlungen für die GUI-Testautomatisierung formuliert. So sollte der Schwerpunkt automatisierter Tests auf technischen Schnittstellen, nicht aber auf der GUI liegen, um so von vornherein die Komplexität der GUI-Tests handhabbar zu machen.

Johann Gietl präsentiert in „Testautomatisierung heute: Erfolgsfaktoren bei der automatischen Ausführung von GUI-Tests“ eine Checkliste für die Umsetzung von GUI-Testautomatisierungsprojekten. Er stellt dazu die technische Architektur und das Testframework einer GUI-Testautomatisierung heraus und diskutiert deren Notwendigkeiten und Abgrenzungen. Ein derart strukturiertes Vorgehen bei der GUI-Testautomatisierung verbessert die Zusammenarbeit der Fachexperten und Projektmanagern mit den Testautomatisierungsexperten. Auch ist die Etablierung und Verfolgung einer langfristigen Strategie wesentlich für den Erfolg einer GUI-Testautomatisierung, da auf diese Weise Wiederverwendungspotenziale über mehrere Projekte hinweg genutzt werden können.

In „Ein Dialog unter Fremden: Testautomatisierung in der Praxis“ entführt uns Nico Orschel in ein imaginäres, aber durchaus repräsentatives Zwiegespräch via

E-Mail zwischen einem Tester und einem Testautomatisierungsexperten. Dabei werden Chancen, Grenzen und Fallstricke einer Testautomatisierung in agilen Projekten diskutiert. So werden Testarten, Teamorganisation und die Verteilung der Teststufen und Testaufwände für Legacy-Software und neuentwickelte Software diskutiert. Ausgangspunkt wird darauf verwiesen, dass Werkzeuge zwar die Grundlage einer jeden Testautomatisierung sind, letztendlich jedoch Prozesse und Teams die Testautomatisierung tragen und weiterentwickeln müssen.

Ralph Guderlei argumentiert in „Gegurke' für Fortgeschrittene“, dass die Wartung und Weiterentwicklung automatisierter Systemtests umso besser gelingt, je mehr von konkreten Testskripten abstrahiert und je besser eine Ebene für die logische Testfallbeschreibung eingebaut werden kann. So können Tests verständlicher beschrieben und direkt für die Testausführung verwendet werden. Ein besonderer Vorteil ist dabei, dass die automatisierten Systemtests nicht erst am Ende, sondern bereits während der Entwicklungsphase zur Verfügung stehen und genutzt werden können. Das Vorgehen wurde mit Hilfe des Ruby on Rails-Werkzeugs Cucumber realisiert und in an Scrum angelegten Projekten eingesetzt.

In „Effiziente Testtechniken zur Prüfung der IT-Sicherheit“ beschreiben Jürgen Großmann und Martin Schneider Methoden zur modellbasierten Risikoanalyse sicherheitskritischer Systeme, zur Entwicklung automatisierter Sicherheitstests und zur Variation dieser Tests mittels Fuzzing, so dass auch unerwartete Sicherheitslücken aufgedeckt werden können. Um den Überblick beim Sicherheitstesten, die Nachverfolgbarkeit und die Ableitung von Metriken zu gewährleisten, wird ein werkzeugübergreifendes Abhängigkeitsmanagement beschrieben. Die Bewertung der Ergebnisse industrieller Fallstudien in Bezug auf die Verbesserungen des Sicherheitstestens erfolgt mittels sogenannter Verbesserungsprofile, die in Anlehnung an TMMI und TPI für Sicherheitstests erarbeitet wurden.

Dirk Hovemann erläutert in „Mobile Testing: Beherrschung von Komplexität und Kostentreibern – Testing as a Service“ die Anforderungen an das Testen mobiler Applikationen, wie z. B. die Verfügbarkeit verschiedenster Endgeräte und Testkonfigurationen (als Emulationen und reale Geräte) sowie die Testautomatisierung von

funktionalen Tests, Last- und Leistungstests sowie Sicherheitstests. Mit seinem neuen Testing as a Service-Ansatz ermöglicht das vorgestellte Konzept das flexible Aufsetzen von Testumgebungen und eine effiziente Ausführung der Tests auch in agilen Entwicklungsprozessen.

In „Delta-orientiertes Testen von variantenreichen Systemen“ diskutieren Sascha Lity, Remo Lachmann, Malte Lochau, Michael Dukaczewski und Ina Schaefer die Herausforderungen des Testens von variantenreichen Systemen, wie sie in der Automobilindustrie, der Industrieautomatisierung, der Medizintechnik etc. zu finden sind. Die vorgestellte Methode des Delta-Testens kann für das effiziente Testen von Varianten auf Komponenten-, Integrations- und Systemteststufe genutzt werden. Die so erzielbaren Einsparpotenziale werden anhand der Body Comfort System-Fallstudie erläutert.

Baris Güldali, Dr. Michael Mlynarski, Marvin Grieger, Dr. Stefan Sauer und Torsten Grünen erläutern in „Testen bei Migrationsprojekten: Migrationsszenarien und relevante Testtypen“ vier Arten der Softwaremigration und deren Auswirkungen auf die Teststufen, die Testspezifikationen und den Testprozess für das Zielsystem. Anhand eines Migrationsprojekts mit dem Ziel einer Betriebssystemumstellung wird gezeigt, wie bei der Wiederverwendung von Alt-Testfällen oder aber bei deren Restrukturierung und Anpassung für das Zielsystem vorgegangen werden kann. Zudem erfordert das Zielsystem oftmals neue Tests für die zusätzlichen Schnittstellen sowie zur Absicherung der nichtfunktionalen Eigenschaften.

Eine neue Art des Testens unter Nutzung einer Crowd von Testern stellt Jan Wolter in „Crowdtesting: Anwendungsfälle und Grenzen einer neuen Methode“ vor. Es wird erläutert, wie ein Crowdtesting-Prozess, bei dem sowohl automatisierte als auch manuelle Tests zur Anwendung kommen können, prinzipiell funktioniert. Am Beispiel der testhub-Plattform wird dargestellt, welche Vorteile mit Crowdtesting verbunden sein können und welche Aspekte beachtet werden müssen, um diese Vorteile zu nutzen.

In „Das Testumgebungsmanagement der Zukunft: Virtualisierung des Anwendungsverhaltens“ diskutiert Wolfram Kusterer die Nutzung von Virtualisierungskonzepten für die Effektivierung von Systemtests. Während viele Organisationen bereits die Virtualisierung von Hardware- und

Betriebssystemen für reale Testsysteme und -umgebungen nutzen, liegt ein weiterer möglicher Gewinn in der sogenannten Virtualisierung von Anwendungsverhalten mittels Virtual Assets. Die vorgestellte Methode zur Erfassung, Modellierung, Konfiguration und Bereitstellung der Virtual Assets vermeidet Engpässe bei der Bereitstellung von Testumgebungen und damit verbundene Kapazitätseinschränkungen bei den Testteams und/oder den Testumgebungen.

Ich hoffe, dass Sie diese Zusammenstellung von aktuellen Arbeiten und Ergebnissen rund um die Automatisierung der Testausführung und Testgenerierung auch und gerade in agilen Entwicklungsprozessen anregend für Ihre tägliche Test-, Testautomatisierungs- und Qualitätssicherungsarbeit finden werden. Das Rad der Testautomatisierung lässt sich nicht zurückdrehen. Nutzen auch Sie die Vorteile und bauen Sie auf den Erfahrungen anderer auf, um Testautomatisierung angemessen und erfolgreich umzusetzen.

Referenzen

Neben den Referenzen im Online-Themenspecial „Testing“ vom Oktober 2013 verweist das Editorial auf:

[Ben83] H. D. Benington, Production of large computer programs, *Annals of the History of Computing* 5.4, S. 350-361, 1983.

[Ets11] ETSI ES 202 951, Methods for Testing and Specification (MTS), Model-Based Testing (MBT), Requirements for Modelling Notations, V1.1.1, Juli 2011.

[Ets13-a] ETSI EG 203 130, Methods for Testing and Specification (MTS), Model-Based Testing (MBT), Methodology for standardized test specification development, V.1.1.1, Februar 2013.

[Ets13-b] ETSI TTCN-3, ETSI Testing and Test Control Notation, v4.5.1, April 2013, ebenso ITU-T Z.140 ff.

[Iee10] IEEE 1671, Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Information via XML, 2010.

[Gtb13] GTB, Advanced Level, Stand September 2013, siehe:

<http://www.german-testing-board.info/pruefungsinteressierte/istqbR-certified-tester-schema/advanced-level.html>.

[Idc12] IDC, Worldwide Automated Software Quality 2012-2016 Forecast: Multimodal Development with Mobile, Social, and Cloud Drives ASQ Growth, Juni 2012.

[Idc13] IDC MarketScape, Worldwide Cloud Testing and ASQ SaaS 2012-2013 Vendor Analysis – Enabling Business Agility and Quality in the Cloud, Februar 2013.

[Omg13] OMG UTP, UML Testing Profile, v.1.2, April 2013.