



□ Prof. Dr. Ina Schieferdecker

(E-Mail: [ina.schieferdecker@fokus.fraunhofer.de](mailto:ina.schieferdecker@fokus.fraunhofer.de)) hält seit 2003 einen Fraunhofer-Stiftungslehrstuhl an der Technischen Universität Berlin zum Entwurf und Testen von kommunikationsbasierten Systemen. Frau Schieferdecker beschäftigt sich seit 1994 mit Fragen der Analyse, des Testens und der Bewertung von kommunikationsbasierten Softwaresystemen unter Nutzung von spezifikationsbasierten Techniken wie UML, MSC und TTCN-3.

## objektspektrum themenspecial: testing

### Software-Qualität – ja, und?

Deutschland ist bekannt für seine qualitativ hochwertigen Produkte. Über Jahrhunderte haben sich rigide Qualitätsvorschriften in den Ingenieursdisziplinen entwickelt, welche über Standards gefordert und durch verschiedene Abnahmeverfahren (z. B. den TÜV) geprüft werden. Bei der Software sind wir mitnichten soweit. Es wird zwar verstanden, dass dieser immaterielle Teil der Produkte maßgeblich zur Funktionalität der Produkte beiträgt (so ist beispielsweise Software in der Automobilelektronik mittlerweile der Innovationstreiber), doch ist die Qualitätssicherung der Software noch nicht im gleichen Maße durch Standards unterlegt wie in den anderen technischen Bereichen. Das liegt einerseits daran, dass sich noch das Verständnis der Notwendigkeiten für die Qualitätssicherung entwickeln muss, und andererseits die Aufwände für das Testen als auch die Bedingungen für ein Testende präziser zu fassen sind. So verstehen wir beispielsweise relativ gut, welche codebasierten Testverfahren (auch strukturorientiertes Testen bzw. White-Box-Testing genannt) welche Qualitätsstufen ermöglichen, sodass sich diese Verfahren in den jeweiligen standardisierten Methoden zur Qualitätssicherung sicherheitskritischer Systeme wiederfinden (beispielsweise fordert D0178b für die Luftfahrt auf Sicherheitsebene 3 die MC/DC-Abdeckung der codebasierten Tests). Andererseits verstehen wir ebenso gut, dass codebasierte Testverfahren nur in Kombination mit spezifikationsbasierten Testverfahren (auch Black-Box-Testing genannt) die nötige Qualität absichern. Das Arbeitsgebiet zum spezifikationsbasierten Testen bezüglich der Methoden und Werkzeuge ist noch relativ offen, sodass zwar Empfehlungen, aber keine genauen Vorgaben möglich sind.

Diese Situation spiegelt sich direkt im Ihnen vorliegenden Online-Themenspecial des OBJEKTSpektrums zum Thema Software-Testen wider. Die Autoren sind frei in der Wahl der präsentierten Themen, wählen aber in der Mehrzahl Themen aus dem Bereich des spezifikationsbasierten Testens. So können Sie beispielsweise Neuigkeiten zum anforderungsbasierten Testen, zum modellbasierten/modellgetriebenen Testen und zum risikoorientierten Testen erfahren. Artikel zu verwandten Themen wie Testmanagement oder Testerausbildung an den Hochschulen runden die Softwaretest-Ausgabe ab.

Lassen Sie mich Ihnen nun die Artikel einzeln näher bringen. Ich möchte Sie aber bezüglich der Artikel zum spezifikationsbasierten Testen noch etwas um Geduld bitten und mit den verwandten Themen beginnen.

**Simon Gaßner** beschäftigt sich in seinem Artikel „Managed Test Services“ mit der Auslagerung von Testaufgaben an IT-Dienstleister. Es werden Anforderungen an das Outsourcing von Test Services aufbauend auf einer aktuellen Umfrage zu diesem Thema reflektiert und das Vorgehen bei Dienstleistersauswahl und Assessment und entlang des Modells der Test Factory und der Service Level Agreements beschrieben. Der Artikel nennt eine Reihe von Faktoren, die ein erfolgreiches Outsourcing befördern.

Einen anderen Blick auf das Thema Testmanagement nimmt **Kai-Uwe Wahl** ein. In seinem Artikel „TechniData spart 20 Prozent Zeit beim Testmanagement“ beschreibt er, wie bei TechniData ein vereinheitlichter Testprozess und eine vereinheitlichte Testumgebung ein erhebliches Einsparpotenzial aufzeigen.

Die GUI (Graphical User Interface) bildet oftmals den Zugang für Black-Box-orientierte Tests. **Christoph Preschern** präsentiert

in seinem Artikel „Ein neuer Weg der GUI-Testautomatisierung“ einen GUI-Testwerkzeugkasten, der eine Vielzahl von GUI-Technologien unterstützt und für die Tests zugänglich macht. Im Kern steht ein sogenanntes Repository, das die GUI-Elemente logisch in ihren Attributen erfasst und so die Wiederverwendung und Weiterentwicklung von GUI-Tests flexibilisiert.

**Joachim Herschmann** diskutiert in seinem Artikel „Agile Testprozesse – mit großem „A“ wie Automatisierung“ die zentrale Rolle der Testautomatisierung bei einem auf Entwicklungsiteration, kontinuierliches Testen und Effizienz getrimmten Prozess. Testautomatisierung wird hierbei als Vorbedingung für die nach agilen Prinzipien notwendigen wiederholten Tests verstanden, bei der man insbesondere auf Wiederverwendung und Reproduzierbarkeit der Tests und eine damit einhergehende Flexibilität der verwendeten Testwerkzeuge achten muss.

Einem bislang weniger beachteten Gebiet des Testens widmen sich **Professor Jens Grabowski, Thomas Rings, Philip Makedonski** und **Benjamin Zeiss** von der Universität Göttingen in ihrem Artikel „Systematische Qualitätssicherung für Testartefakte“. Sie stellen die Wichtigkeit der Testartefakte und die Notwendigkeit, diese selbst bezüglich ihrer Qualität zu sichern, heraus. Es werden die verschiedenen Methoden der Qualitätssicherung von Testartefakten diskutiert und in ihrer Anwendung auf TTCN-3 Tests beispielhaft präsentiert. Es ist zu erwarten, dass statische Analysewerkzeuge demnächst ebenso Testartefakte zum Ziel ihrer Analyse machen werden.

Darüber hinaus ist das angemessene Testen bei Softwareänderungen noch nicht genügend betrachtet. **Rüdiger Schilling** stellt in seinem Artikel „Radikale Änderung der Teststrategien durch

# objektspektrum themenspecial: testing

100%-ige Automation von Massenänderungen“ ein regelbares, vollautomatisiertes Vorgehen für Massenänderungen an Software vor, das mit einem auf die Regel fokussierten Testansatz verknüpft wird, sodass sich die Testaufwände reduzieren und auf das Wesentliche konzentrieren lassen. Interessant hierbei ist die Verlagerung des Testobjekts von der geänderten Software auf die ändernde Software, die durch Änderungsregeln formalisiert ist.

Das Testprozessverbesserungsmodell TPI ist vielen bekannt. Die Weiterentwicklung desselben zum Business Driven TPI, welches in diesem Jahr veröffentlicht werden wird, wird von **Hubert Beckmann** und **Marc Barnscheidt** in ihrem Artikel „Business Driven TPI® – Testprozessverbesserung im Spannungsfeld verschiedener Ziele“ vorgestellt. Hierbei dienen Geschäftsziele der Priorisierung von Testoptimierungsmöglichkeiten. Die Verknüpfung des Testprozesses mit dem Entwicklungsprozess ist expliziter gestaltet. Die Zukunft wird zeigen, ob Business Driven TPI ähnlich erfolgreich angenommen wird wie TPI.

Nicht zuletzt müssen wir uns um das Berufsbild des Software-Testers kümmern. Während die Certified Tester Ausbildung bereits gut in der Industrie angenommen wird, nimmt sie genauso an den Hochschulen Fahrt auf. Das ist, wie im Artikel „Berufsbild „Software-Tester“ immer beliebter - Hochschulen steigen in die Ausbildung ein“ von **Professor Andreas Spillner** und **mir** beschrieben, ein wesentlicher Schritt, um Studenten auf eine spätere Arbeit in der Qualitätssicherung und dem systematischen Testen von Software vorzubereiten.

Das Ihnen vorliegende Online Themenspecial Testing enthält zudem eine Vielzahl von Artikeln zu verschiedenen spezifikationsbasierten Testansätzen, die nachfolgend kurz beschrieben werden.

Das geschäftsprozessbasierte Ende-zu-Ende Testen von integrierten Geschäftssystemen wird von **Wolfram Kusterer** in seinem Artikel „End-to-End Test und Validierung“ diskutiert. Dazu wird ausgehend von den Nutzungsszenarien der zu testenden Applikation ein mehrschichtiger Testansatz gewählt, der auf Browser-, Middleware- und Code-Ebene die für die Validierung notwendigen Teststimuli anlegt und Beobachtungen der Applikation und seiner Komponenten bereitstellen kann. Dieser auf dienstorientierte Architekturen ausgerichtete Testansatz kann auch auf in Entwicklung befindenen Applikationen angewendet werden, für die Platzhalterkomponenten für nichtverfügbare oder instabile Systemkomponenten genutzt werden können.

**Kerstin Nagel** stellt in ihrem Artikel „Bessere Softwarequalität durch anforderungsbasiertes Testen“ eine Methodik vor, bei der Anforderungen die Basis für das Test-Management sind. Zudem sind Software-Configuration-Management, Change-Management, Defect-Management, Release-Management und Portfolio-Management in die Methodik eingebunden. Im Zentrum steht

eine anforderungsbasierte Testplanung und der permanente Abgleich der Testergebnisse und Fehlerreports gegen die Anforderungen.

Der Artikel von **Dr. George Din**, **Klaus-D. Engel** und **Axel Rennoch** beschreibt einen neuen Ansatz zum architekturbasierten Testen von eingebetteten Systemen. Dabei werden verschiedene Perspektiven auf das zu testende System in Relation zueinander gestellt, sodass übergreifende Testszenarien abgeleitet werden können. So gelingt es beispielsweise zu prüfen, ob Abhängigkeiten auf Funktionsebene korrekt auf Komponentenebene abgebildet und umgesetzt sind. Der Ansatz des architekturbasierten Testens steht am Anfang, doch zeigen erste Ergebnisse, dass die bislang auf ein Modell konzentrierten Tests durch die integrativen Tests über Modelle hinweg sinnvoll ergänzt werden.

Das modellbasierte Testen wird in diesem Online Themenspecial von mehreren Autoren adressiert. So diskutiert **Dr. Horst Kargl** in seinem Artikel „Flexibilität beim modellbasierten Testen“ wie modellbasierte Tests bereits in der Entwurfsphase von Systemen genutzt werden können, um in frühen Phasen der Entwicklung Fehler bei Anforderungen, im Entwurf und bei der Modellierung identifizieren zu können. Simulationswerkzeuge kombiniert mit Mock-Objekten oder externen Applikationen bzw. Systemen werden hierbei genutzt, um die Modelle ausführbar zu machen und auf sie die Tests anzuwenden. Das vorgestellte Werkzeug unterstützt dabei das dynamische Testen von Modellen in Form von Aktivitätsdiagrammen und Zustandsautomaten.

**Dr. Marc Lettrari** und **Frank Braun** präsentieren in ihrem Artikel „Steigerung der Testeffizienz – Von modellgetriebener Entwicklung zum modellgetriebenen Testen“ die Möglichkeiten und Vorteile eines modellgetriebenen Testansatzes. Auch in diesem Artikel wird die zentrale Rolle einer Testautomatisierung als Vorbedingung für die Nutzung der Vorteile modellgetriebener Tests verdeutlicht. Die Erhöhung der Abstraktion und damit der Flexibilität der Tests durch Testmodellierung bei gleichzeitiger Pflege der Relationen zu anderen Entwicklungsartefakten werden als weitere wichtige Elemente des modellgetriebenen Tests diskutiert.

Ebenso widmen sich **Dr. Andreas Rätz** und **Martin Koch** mit ihrem Artikel „Testfälle automatisch aus Modellen generieren – ein großer Effizienzgewinn“ der automatisierten Testgenerierung. Für Aktivitätsdiagramme werden verschiedene Überdeckungskriterien und risikoorientierte Bewertungs- und Selektionsstrategien genutzt, um Tests mit der nötigen Testtiefe zu erzeugen. Anforderungen, Tests und identifizierte Fehler werden gemeinsam in einem Werkzeug verwaltet, was die automatisierte Testgenerierung ergänzt.

**Holger Kühnl** beschäftigt sich in seinem Artikel „Testmanagement im hybriden Umfeld – Umgang mit dem Risiko“ im Detail

# objektspektrum themenspecial: testing

mit risikoorientierten Testansätzen im Spannungsfeld hybrider Systeme. Der Artikel leitet ausgehend von Begriffen wie Risiko und funktionale Sicherheit Testanforderungen ab und betrachtet deren Umsetzung für hybride Systeme mit ihren vielfältigen Varianten in Konfiguration und Parametrisierung. Die Risikoabschätzung ergänzt die Ausrichtung der Tests an den Anforderungen und wird neben Testautomatisierung und einem angemessenen Management des Testprozesses als essenziell für die Beherrschung der Komplexität hybrider Systeme gesehen.

Einen modellbasierten Testansatz für eingebettete Systeme stellen **Dr. Joachim Wegener, Peter M. Kruse und Hamilton Gross** in ihrem Artikel „Evolutionäre Funktionstests für MiL-, SiL- und HiL-Umgebungen“ vor. Dabei werden evolutionäre Algorithmen zur Generierung optimierter Funktionstests genutzt, wobei zusammen mit dem evolutionären Testframework eine Modellierungs- und Simulationsumgebung und optional ein HiL-Teststand genutzt werden. Ich würde erwarten, dass evolutionäre Testverfahren nicht nur helfen, die Komplexität eingebetteter Systeme und ihrer Tests, sondern ebenso von Software-Tests in anderen Gebieten wie dem Testen von Applikationen in SOA-Umgebungen beherrschen zu können.

Lassen Sie uns abschließend zu einem Kernpunkt zurückkommen, dem jeder Software-Tester über kurz oder lang gegenübersteht: Software-Qualität – ja, und? Argumentationen, die die Aufwände für die Qualitätssicherung, wobei wohl bemerkt dazu mehr gehört als „nur“ das Software-Testen, herleiten, darstellen bzw. rechtfertigen sind für jede systematische Projektplanung notwendig. Dabei müssen sich analytische und empirische Ergebnisse ergänzen. Während die Empirie uns sagt, dass sich QS-Aufwände und Entwicklungsaufwände ungefähr die Waage halten, sagen uns analytische Ergebnisse je nach genutzten Annahmen bezüglich der vorhandenen Prozess-, Entwicklungs- und Software-Qualität weniger bis mehr Aufwände im Vergleich zum Entwicklungsaufwand voraus. Jüngstens habe ich eine Firma, die in einem geschäftskritischen Umfeld agiert, kennengelernt, bei der die Testaufwände mehr als das Dreifache (!) der Entwicklungsaufwände betragen, was mich in dieser Größenordnung selber überrascht hat. Wir wissen alle, dass beispielsweise bei Microsoft ab Windows XP der Testaufwand den Entwicklungsaufwand überstiegen hat. Andererseits kennen wir sicher ebenso alle Firmen, bei denen mit wenigen Ressourcen und ohne die nötige Konzentration getestet wird. Das ist oftmals der Fall, wenn das Management die

Notwendigkeit der Qualitätssicherung nicht erkannt bzw. nicht verinnerlicht hat, die Qualitätssicherung auf ihre Kunden abwälzt oder aber es noch nicht zum „Super-Gau“ mit dieser Software gekommen ist.

Jüngste Ereignisse haben uns jedoch wieder vor Augen geführt, wie zentral und geschäfts- beziehungsweise sicherheitskritisch die Software mittlerweile geworden ist. Denken Sie beispielsweise an den Totalausfall bei T-Mobile am 21. April 2009, 16:10, der durch einen Software-Fehler im Home Location Register verursacht worden ist. Schon am 10. März 2009 gab es Störungen beim selben Anbieter im Datennetz. Nun liegt es nicht am Anbieter, fielen doch beispielsweise am 15. Januar 2009 das Datennetz von E-Plus und am 19. Mai 2008 das von Vodafone aus. Vermutet wurden ebenso Softwarefehler, welche aber nicht abschließend öffentlich bestätigt wurden, noch wurden die Ausfallkosten beziffert. Beim Ausfall am 21. April 2009 ging Prof. Torsten Gerpott von der Universität Duisburg-Essen von Millionen-Verlusten pro Ausfallstunde aus. Verbleibt noch zu sagen, dass softwareverursachte Ausfälle ebenso wenig nur einem Wirtschaftszweig zuzuordnen sind, ist derartiges doch auch aus anderen Bereichen wie Banken, Automobil, Bahn, Luftfahrt, Medizintechnik, etc. bekannt. Der interessierte Leser sei auf die „Software Horror Stories“ (<http://www.cs.tau.ac.il/~nachumd/horror.html>) bzw. auf die „Collection of Software Bugs“ (<http://www5.in.tum.de/~huckle/bugse.html>) verwiesen.

Wie stellen wir uns also Fragen nach angemessenen Testmethoden, zweckmäßigen Testprozessen und sinnvollen Testaufwänden für ein gegebenes Produkt/einen Dienst? Erste Antworten liefern risikoorientierte Testansätze als auch anforderungsbasierte und modellbasierte Testmethoden – Artikel dazu finden Sie, wie beschrieben, in dieser Ausgabe. Die Arbeiten sind aber mitnichten abgeschlossen, sodass ich Sie schon heute gerne auf das Online-Themenspecial Testing im nächsten Jahr mit neuesten Ergebnissen einstimmen möchte. Bis dahin möchte ich Sie ermuntern, mir Ihre Anforderungen und Vorschläge für die kommende Ausgabe zukommen zu lassen. Bleiben Sie dem Gebiet des Software-Testens weiter gesonnen und überlegen Sie, Anregungen aus den Software-Test-Ausgaben in Ihr tägliches Geschäft zu übernehmen. Eine gute Grundlage kann das beschriebene Certified-Tester-Schema sein, das auch an Hochschulen greift, seinen Schwerpunkt aber in der berufsbegleitenden Ausbildung in der Industrie hat. ■