

Vorwort

Am 23. und 24. Juni 2016 hat sich die Fachgruppe »Test, Analyse und Verifikation von Software« (TAV) im Fachbereich »Softwaretechnik« der »Gesellschaft für Informatik e.V.« (GI) in Bremen zu ihrem 39. Treffen zusammengefunden. Das erste Treffen der Fachgruppe (damals als neuer Arbeitskreis der Fachgruppe »Softwaretechnik«) fand vom 17. bis 19. Juni 1991 in Benthe bei Hannover statt. Damit war das 39. Treffen im Juni 2016 ein ganz besonderes:

25 Jahre TAV!

Auf dem Treffen wurden zu unterschiedlichen Themen aus dem Bereich der Fachgruppe Vorträge gehalten. Zu jedem Vortrag wurde vor Ort eine Sketchnote von Diana Meier-Soriat¹ angefertigt. Alle Sketchnotes des Treffens sind auf den Seiten der Softwaretechnik-Trends zu finden (<http://pi.informatik.uni-siegen.de/gi/sttl> im Band 36 (2016), Heft 3 (August) unter FG TAV Bericht, 2. PDF). Zu einigen ausgewählten Vorträgen des 39. Treffens finden sich im Buch Beiträge, die daran zu erkennen sind, dass zu den Vorträgen die jeweilige Sketchnote mit abgedruckt ist.

Wir – die Herausgeber – hatten als jeweils langjährige Sprecher der Fachgruppe die Idee, zum 25-jährigen Jubiläum der Fachgruppe TAV nicht nur einen Blick zurück in die Vergangenheit zu werfen, sondern auch nach vorn zu schauen und die aktuelle und zukünftige Ausrichtung des Testens in Theorie und Praxis in den weiteren Buchbeiträgen zu beleuchten.

Nach einem Grußwort zum Jubiläumstreffen des Präsidenten der Gesellschaft für Informatik und damaligen Mitbegründers der Fachgruppe, Prof. Dr. Peter Liggesmeyer, und dem Vorwort der Herausgeber startet das Buch mit dem Beitrag von Jochen Ludewig »Softwarequalität zwischen hohem Ziel und hohler Floskel«. Dieser bietet einen sehr guten Einstieg in die Thematik und regt an der ein oder anderen Stelle doch zum kritischen Nachdenken über Qualität im Allgemeinen und Softwarequalität im Besonderen an.

1. <https://www.sketchnotes-by-diana.com>

Neues aus Theorie und Praxis

Oft ist ja zu hören, dass sich im Testen nichts tut, keine Neuerungen vorhanden sind. Dass dem nicht so ist, zeigen die nächsten drei Beiträge.

Joachim Wegener und Peter Kruse stellen leistungsfähige neue Testverfahren vor. Sie gehen auf die Klassifikationsbaum-Methode und ihre aktuellen Erweiterungen ebenso ein wie auf das modellbasierte Testen mit dem Schwerpunkt Time Partition Testing. Evolutionäre und suchbasierte Testverfahren runden den Beitrag ab.

Matthias Höschele, Alexander Kampmann und Andreas Zeller beschreiben das »Testen mit automatisch bestimmten Eingabegrammatiken«. Unter Verwendung von Fuzzing-Vorgehensweisen werden automatisch Testfälle aus Grammatikbeschreibungen der Eingaben erzeugt, die fehlerhafte Konstellationen der Eingaben aufdecken. Ein vielversprechender Ansatz, der auch mit ganz geringen Ausgangsinformationen über das Testobjekt unter Verwendung von aktiven Lernalgorithmen sein Ziel erreicht.

Ergebnisse der Trendstudie »The Future of Testing« stellen Tilo Linz und Bernd Flessner vor und wagen damit einen Blick in die Zukunft des Testens. Tilo Linz kommt aus der Praxis der Qualitätssicherung. Bernd Flessner ist ein renommierter Zukunftsforscher und Wissenschaftsautor und arbeitet am Zentralinstitut für Wissenschaftsreflexion und Schlüsselqualifikationen (ZiWiS) der Friedrich-Alexander-Universität Erlangen-Nürnberg. Sie debattieren unterschiedliche (Test-)Szenarien der Zukunft.

Modellbasiertes Testen

Modellbasiertes Testen, also die Nutzung von Modellen für die Automatisierung von Testaktivitäten sowie die Modellierung von Artefakten im Testprozess, wird als ein aktueller Trend der Qualitätssicherung in den folgenden drei Beiträgen ausführlich behandelt.

Jürgen Großmann liefert mit seinem Übersichtsvortrag »Modellbasiertes Security-Testen« einen sehr guten Einstieg in das modellbasierte Testen und zeigt, wie das proaktive Erkennen von Fehlern und Sicherheitslücken vor der Inbetriebnahme durch modellbasiertes Testen möglich ist. Seit Jahren gibt es substanzielle Forschungsbeiträge im Bereich Security-Testen, die darauf abzielen, als Maßnahmen bereits im Entwicklungs- und Integrationsprozess die Qualität der Softwareprodukte nachhaltig zu verbessern und die der Sicherheitseigenschaften systematisch zu erhöhen. Der Beitrag geht auf die Ergebnisse der Forschungsprojekte DIAMONDS, RASEN, MIDAS und PREVENT näher ein und stellt unterstützende Werkzeuge vor.

Jan Peleska, Wen-ling Huang und Felix Hübner stellen einen vollständigen modellbasierten Testansatz vor. Sie beschreiben die aktuellen Ergebnisse beim

Einsatz einer neuartigen vollständigen Teststrategie auf der Grundlage der Äquivalenzklassenpartitionierung in Kombination mit adaptiven Stichproben. Unter Verwendung von komplexeren Fehlerdomänen führen die neuen Strategien zu einer deutlichen Reduzierung der Testfälle.

Unter dem Titel »Lazy On-the-fly Model-Based Testing« berichtet David Faragó über die Generierung von Testfällen aus Modellen. Dies ist ein wichtiges Forschungsthema, da es schwer ist, sinnvolle Testfälle effizient zu generieren. State of the Art ist »on-the-fly Model-Based Testing« und »offline Model-Based Testing«. Erstes führt die Testgenerierung und -ausführung synchron durch, erzeugt einen Testschritt und führt diesen Testschritt auf dem zu prüfenden System aus. Im Gegensatz dazu erzeugt offline Model-Based Testing zuerst die gesamte Testsuite, d.h. alle Testschritte, und führt diese danach erst aus. David Faragó stellt einen neuartigen Ansatz vor, das »Lazy On-the-fly Model-Based Testing«. Der Ansatz integriert die beiden vorhandenen Ansätze und erreicht bzw. erhöht deren jeweilige Vorteile.

Spezialthemen

Testen hat viele Aspekte und wird ständig durch neue Technologien herausgefordert! Dies reflektieren die folgenden Beiträge aus den Bereichen Testmanagement sowie IoT und autonome Systeme.

Baris Güldali und Masud Fazal-Baqiae durchleuchten die Herausforderungen des Test- und Qualitätsmanagements in der agilen Softwareentwicklung und ziehen aus eigenen Projekterfahrungen einen Vergleich zwischen klassischem und neuartigem Test- und Qualitätsmanagement. Betitelt ist ihr Beitrag mit »Test- und Qualitätsmanagement in agilen Projekten«.

»Quality Engineering für das Internet der Dinge« behandeln Ina Schieferdecker, Axel Rennoch und Michael Wagner. Vom Internet der Dinge (Internet of Things, kurz IoT) sind umfassende Weiterentwicklungen und Umbrüche zu erwarten, wie sie entlang der Verbreitung des *traditionellen* Internets in Kombination mit der Mobilkommunikation geschehen sind. Die Qualitätsanforderungen an IoT-Komponenten und -Lösungen sind sehr hoch und müssen durch moderne Methoden zur Konstruktion und Absicherung der Qualitäten adressiert werden. Die Autoren geben einen Überblick zum Verständnis von IoT und skizzieren Qualitätsanforderungen und grundlegende Testansätze.

Im Beitrag »Test Case Selection Strategy for Self-Organization Mechanisms« von Benedikt Eberhardinger, Hella Seebach, Dominik Klumpp und Wolfgang Reif geht es um das Testen von Selbstorganisationsmechanismen. Die Herausforderung besteht darin, dass bei autonomen, sich selbst organisierenden Systemen der Zustandsraum sehr groß wird und die Umgebung, in der das System betrieben werden soll, teilweise unvorhersehbar ist. Die Testfallgenerierung und -auswahl gestaltet sich daher als äußerst schwierig. Die Angemessenheit einer Test-

suite hängt von Laufzeitaspekten ab, z.B. von der aktuellen Struktur des Systems in seiner Umgebung. Die Autoren stellen einen Ansatz für die Testfallgenerierung und -auswahl vor, der auf Selbstorganisationsmechanismen und deren Charakteristik abgestimmt ist. Somit ist es möglich, die Anzahl der Testfälle im Vergleich zu einer zufälligen Auswahlstrategie effizient zu reduzieren, ohne dabei die Angemessenheit zu verlieren.

Historisches

Die drei Beiträge des letzten Buchteils blicken zurück auf die Historie des Testens in Deutschland sowie auf die Berücksichtigung der heute zum Stand der Technik zählenden objektorientierten Entwicklung im Test.

Rudolf van Megen vergleicht das Testen von damals und heute und fragt: Was hat sich in 25 Jahren (nicht) geändert? Softwaresysteme haben heute einen vielfältigen Einsatzbereich und die Software eine viel höhere Komplexität als vor 25 Jahren. Die Bedeutung von Qualitätsmerkmalen wie Usability, Security und Portability rückte immer mehr in den Vordergrund, wofür passende Testverfahren gefunden werden mussten. Das Testvorgehen an sich hat sich ebenso weiterentwickelt, und zwar in Bezug auf die Teststufen, Testobjekte, die Personen und Organisationsformen sowie die Aktivitäten und die dafür benötigten Methoden und Techniken. Stillstand gab und gibt es weder in der Entwicklung noch im Testen von Software!

Von den Anfängen der Testautomation berichtet Harry Sneed. Er schildert die Entwicklung der ersten Testwerkzeuge im deutschsprachigen Raum. Es begann 1977 mit dem Software-Prüfstand. Das Werkzeug wurde für den Modultest in dem integrierten Transportsteuerungssystem der Bundesbahn eingesetzt. Es folgten weitere ähnliche Modultestwerkzeuge u.a. für IBM-Assembler, IBM-8100 COBOL, PL1 und Kienzle's strukturierte Programmiersprache für Mikroprozessoren. Im Jahre 1982 entwickelte er auf dem IBM-DOS-VSE das erste Integrationstestsystem für die Tornado-Lagerhaltungssoftware. Dieses System enthielt schon alle Komponenten eines modernen Testsystems einschließlich Testdatengenerator, Testmonitor, Testergebnisvalidator, Dateikomparator und Testskriptsprache, nur alles sehr rudimentär implementiert. Die Frühphase der Testautomation endete mit der ACM-Testkonferenz im März 1982 an der Bundeswehrhochschule in Neubiberg.

Im Beitrag »Testen objektorientierter Programme – eine Retrospektive« erinnert Dehla Sokenou daran, dass mit der zunehmenden Verbreitung objektorientierter Programmiersprachen wie C++ und Java Mitte der 1990er-Jahre neben Fragestellungen zu Analyse und Design auch solche zum Thema Testen objektorientierter Software aufkamen. Inzwischen ist es ruhig um die Forschung im Bereich Testen objektorientierter Programme geworden. Ist also bereits alles erforscht und finden die Techniken, die damals entwickelt wurden, heute Anwendung? Wie haben neuere Entwicklungen im Bereich objektorientierter Softwareentwicklung, wie aspektorientierte Programmierung und agile Vorgehensweisen, das Testen beeinflusst? Der Beitrag gibt einen Abriss über die historische Entwicklung des objektorientierten Testens und den aktuellen Stand der Forschung.

Wir wünschen unseren Lesern viel Spaß bei der Lektüre und hoffen, dass Sie einige Ideen und Anregungen (wieder-)finden und vielleicht sogar in Ihrem Testalltag umsetzen können.²

Wenn Sie darüber hinaus das Buch ermuntert hat, sich intensiver mit den Fragestellungen von Test und Qualitätssicherung zu beschäftigen, würde es uns sehr freuen, Sie bei einem der nächsten Treffen der Fachgruppe TAV begrüßen zu dürfen. Alle aktuellen Informationen dazu finden Sie unter:

<http://fg-tav.gi.de/>

Andreas Spillner, Mario Winter, Andrej Pietschker
Bremen, Wuppertal, München, im Herbst 2017

2. Alle im Buch verwendeten URLs wurden am 3.10.2017 überprüft.