

Manfred Baumgartner • Stefan Gwihs • Richard Seidl  
Thomas Steirer • Marc-Florian Wendland

# Basiswissen



# Testautomatisierung

Aus- und Weiterbildung zum  
**ISTQB® Advanced Level Specialist –**  
**Certified Test Automation Engineer**



dpunkt.verlag



**Manfred Baumgartner** verfügt über mehr als 30 Jahre Erfahrung in den Bereichen Softwaretest und Qualitätssicherung. Seit 2001 hat er das QS-Beratungs- und Schulungsangebot der Nagarro GmbH (ehemals ANECON), eines der führenden Dienstleistungsunternehmen im Bereich Softwaretest, auf- und ausgebaut. Er ist Präsidiumsmitglied des Vereins für Softwarequalität und Weiterbildung (ASQF) und des Vereins für Software-Qualitätsmanagement Österreich (STEV) sowie Mitglied des Austrian Testing Board (ATB). Seine umfangreichen Erfahrungen bringt er in viele Präsentationen auf Konferenzen im gesamten deutschsprachigen Raum und in Artikeln und Büchern zum Thema Softwaretest ein.



**Stefan Gwihs** ist als begeisterter Softwareentwickler, Softwaretester und Testautomatisierungsarchitekt für die Nagarro GmbH (ehemals ANECON) tätig, wo er sich aktuell vor allem mit Themen im Bereich Testautomatisierung agiler Softwareentwicklung und DevOps beschäftigt.



**Richard Seidl** hat in seiner beruflichen Laufbahn schon viel Software gesehen und getestet: gute und schlechte, große und kleine, alte und neue, Schokolade und Grütze. Sein Credo: Qualität ist eine Haltung. Wer heute exzellente Software kreieren möchte, denkt den Entwicklungsprozess ganzheitlich: Menschen, Methoden, Tools und Mindset. Als Berater und Coach unterstützt er Unternehmen dabei, Agilität und Qualität zu leben und in der Unternehmens-DNA zu verankern.



**Thomas Steirer** (ehem. Bucsics) leitet als Automatisierungsexperte, Testmanager und Trainer für die Nagarro GmbH (ehemals ANECON) die globale Einheit für Testautomatisierung. Seit 2010 ist er als ISTQB® Certified Tester - Full Advanced Level zertifiziert. Er ist Vortragender einer Vorlesung für Testautomatisierung im Masterstudiengang Software-Engineering am Technikum Wien und forscht an der Nutzung von künstlicher Intelligenz mit dem Ziel, Testautomatisierung noch effizienter zu gestalten.



**Marc-Florian Wendland** ist wissenschaftlicher Mitarbeiter am Fraunhofer-Institut FOKUS in Berlin. Seit über 10 Jahren beschäftigt er sich in nationalen und internationalen, branchenübergreifenden Forschungs- und Industrieprojekten mit Themen der Testautomatisierung in Entwurf und Ausführung. Er ist Mitglied des German Testing Board (GTB) und auch als Trainer für die verschiedenen ISTQB-Programme unterwegs.



Zu diesem Buch – sowie zu vielen weiteren dpunkt.büchern – können Sie auch das entsprechende E-Book im PDF-Format herunterladen. Werden Sie dazu einfach Mitglied bei dpunkt.plus+:

**[www.dpunkt.plus](http://www.dpunkt.plus)**

**Manfred Baumgartner · Stefan Gwihs · Richard Seidl ·  
Thomas Steirer · Marc-Florian Wendland**

# **Basiswissen Testautomatisierung**

**Aus- und Weiterbildung zum ISTQB® Advanced  
Level Specialist - Certified Test Automation  
Engineer**

3., aktualisierte und überarbeitete Auflage



Manfred Baumgartner · [manfred.baumgartner@nagarro.com](mailto:manfred.baumgartner@nagarro.com)  
Stefan Gwihs · [stefan.gwihs@nagarro.com](mailto:stefan.gwihs@nagarro.com)  
Richard Seidl · [office@richard-seidl.com](mailto:office@richard-seidl.com)  
Thomas Steirer · [thomas.steirer@nagarro.com](mailto:thomas.steirer@nagarro.com)  
Marc-Florian Wendland · [marc-florian.wendland@fokus.fraunhofer.de](mailto:marc-florian.wendland@fokus.fraunhofer.de)

Lektorat: Christa Preisendanz  
Copy-Editing: Ursula Zimpfer, Herrenberg  
Satz & Layout: Birgit Bäuerlein  
Herstellung: Stefanie Weidner  
Umschlaggestaltung: Helmut Kraus, [www.exclam.de](http://www.exclam.de)

Bibliografische Information der Deutschen Nationalbibliothek Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Fachliche Beratung und Herausgabe von dpunkt.büchern zum Thema »ISTQB® Certified Tester«: Prof. Dr. Andreas Spillner · [Andreas.Spillner@hs-bremen.de](mailto:Andreas.Spillner@hs-bremen.de)

ISBN:  
Print 978-3-86490-675-6  
PDF 978-3-96088-787-4  
ePub 978-3-96088-788-1  
mobi 978-3-96088-789-8

3., aktualisierte und überarbeitete Auflage  
Copyright © 2021 dpunkt.verlag GmbH  
Wieblinger Weg 17  
69123 Heidelberg

*Hinweis:*  
Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.



*Schreiben Sie uns:*

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: [hallo@dpunkt.de](mailto:hallo@dpunkt.de).

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

# **Vorwort zur 3. Auflage**

»*Mit Testautomatisierung automatisch besser!?*«

100% Testüberdeckung, 400% Effizienzsteigerung, deutlich reduziertes Risiko, schnellere Time to Market und stabile Qualität – das waren und sind die Versprechen der Testautomatisierung oder, besser gesagt, jener, die mit Werkzeugen und Beratungsleistungen rund um dieses Thema ihr Geld verdienen. Und seit dem Erscheinen der ersten Auflage dieses Buches steht der Einsatz von Testautomatisierung auf der To-do-Liste von vielen, in Wirklichkeit fast allen Unternehmen, die Software produzieren oder implementieren. Die versprochenen und erwarteten Ziele wurden aber kaum erreicht. Im Gegenteil: Es gibt eine große Diskrepanz zwischen den in den Hochglanzprospekten der Werkzeughersteller dargestellten Errungenschaften und der in vielen Unternehmen vorhandenen Unsicherheit bezüglich des erfolgreichen und nachhaltigen Einsatzes von Testautomatisierung.

Dieses Buch soll eine umfassende, praktische Einführung mit ausreichendem Tiefgang geben und somit einen Leitfaden durch das Thema »Testautomatisierung« für eine Vielzahl von Rollen in diesem Tätigkeitsfeld bieten. Auch im Hinblick auf den schnelllebigen IT-Markt hat sich die Testautomatisierung in den letzten Jahren sowohl als technische als auch als inhaltliche Disziplin rasant

entwickelt. Skalierbare Agilität, Continuous Deployment und DevOps machen das Thema zu einer erfolgskritischen Komponente der Softwareentwicklung.

Diese Dynamik betrifft insbesondere auch die Testautomatisierungswerkzeuge - ob kommerziell oder Open Source. Daher verzichten wir in dieser Auflage darauf, in einem eigenen Kapitel auf konkrete Werkzeuge näher einzugehen, denn deren funktionale Beschreibung und Bewertung durch die Autoren würden bereits zwischen der Finalisierung des Textes und der Drucklegung an Aktualität verlieren. Hinzu kommt, dass es mittlerweile sowohl im Open-Source-Umfeld als auch im kommerziellen Bereich so viele wertvolle Werkzeuge gibt, dass jede Selektion der Autoren gegenüber anderen Herstellern und Communitys unfair wäre. Stattdessen listen wir einige passende Werkzeuge in den jeweiligen Kapiteln über die Testautomatisierungsarchitektur auf, in denen auf ihre Hauptrolle in einer Testautomatisierungslösung Bezug genommen wird. Werkzeugvergleiche und Marktanalysen finden sich rasch und in großer Menge im Internet - wobei selbst diese oft nicht tagesaktuell gepflegt sind.

Der Bedeutung der Disziplin Testautomatisierung trägt auch die internationale Tester-Community Rechnung: Ende 2019 wurde die deutsche Fassung des »ISTQB® Certified Tester Advanced Level Testautomatisierungsentwickler« [\[ISTQB 19c\]](#) freigegeben. Damit ist Testautomatisierung noch mehr als zuvor und endgültig Kernbestandteil der Disziplin »Softwaretest« mit einer eigenen Zertifizierung und dem dazugehörigen Lehrplan. Die dritte Auflage unseres Buches ist nicht nur eine Aktualisierung und Erweiterung der vorhergehenden Versionen. Vielmehr ist es eine komplette Neugestaltung entlang der Struktur dieses jungen und sich weiterentwickelnden Lehrplans. Auch die Kernkapitel über die

Testautomatisierungsarchitektur und über den Umgang mit Testautomatisierung in Projekten und Organisation wurden wesentlich überarbeitet. Sie halten also quasi die Erstauflage eines neuen Buches in den Händen.

Gründe für diese Neugestaltung gibt es mehrere: Die erste Auflage des Buches »Basiswissen Softwaretest« erschien im Jahr 2011, also fünf Jahre vor der Veröffentlichung der ersten englischen Fassung des ISTQB® Advanced Level Syllabus Test Automation Engineer Version 2016. Die zweite Auflage unseres Buches, 2015, war diesem Standard ebenfalls der Zeit voraus. Mit der Freigabe der deutschen Fassung 2019 sahen wir nun die Zeit gekommen, uns an diesem internationalen Standard zu orientieren, der die gemeinsame Sprache und den Wissensaustausch im Bereich Testautomatisierung unterstützen soll. Des Weiteren verfolgen wir auch das Ziel, unsere Leser bestmöglich in die Inhalte des Lehrplans einzuführen und sie auf die Zertifizierungsprüfung vorzubereiten. Der Lehrplan ist umfangreich und ausführlich und für sich ein eigenständiges Nachschlagewerk – wir sind aber überzeugt, mit unserem Buch einen deutlichen Mehrwert zu geben. Und zwar dadurch, dass wir mit praxisbezogenem Kontext, einer leicht lesbaren Form und praktischen Beispielen das Erfassen der Lehrinhalte deutlich einfacher und nachhaltiger ermöglichen, als es das Studium des Lehrplans allein zulässt.

Daher bereitet dieses Buch nicht nur auf die relevanten Inhalte für die Zertifizierungsprüfung vor, sondern vermittelt dem Leser auch die praktische Anwendung der Testautomatisierung.

Die Inhalte des Lehrplans (Version 2019) sind in diesem Buch in abweichender Reihenfolge und mit unterschiedlichen Schwerpunktsetzungen aufgearbeitet

und durch viele essenzielle Themen ergänzt, die wir als Exkurse deutlich ausweisen.

**Die Grundlage für die Zertifizierungsprüfung ist jedoch immer der zum jeweiligen Zeitpunkt offizielle Lehrplan.**

Daher empfehlen die Autoren für die Vorbereitung auf die Prüfung neben der Lektüre des vorliegenden Werkes den Besuch eines entsprechenden Trainings sowie einen Blick in die aktuelle Version des Lehrplans [\[ISTQB 19c\]](#).

Die Abdeckung des Lehrplans ist nur einer von mehreren Aspekten, die wir in diesem Buch behandeln wollen – die drei Hauptziele, die wir uns bereits in der ersten und zweiten Auflage von »Basiswissen Testautomatisierung« gesetzt hatten, behalten ihre Gültigkeit:

Erstens wollen wir Sie davor bewahren, dass Sie aufgrund überzogener Erwartungshaltungen enttäuscht werden. Testautomatisierung ist nicht eine Frage der Werkzeuge, nicht ein Auftrag, die Marketing-Schlagworte diverser Hersteller umzusetzen, sondern lediglich ein Instrument, das es Ihnen ermöglicht, die stetig wachsenden Anforderungen an den Softwaretest besser zu bewältigen.

Zweitens wollen wir Ihnen eine Anleitung geben, wie Sie dieses Instrument bestmöglich nutzen können. Dabei stehen insbesondere die Langfristbetrachtung, die Nachhaltigkeit der Investition und der tatsächliche Business Value im Vordergrund. Diese Aspekte messen sich nicht in einer Codeüberdeckung oder in einer Anzahl von Testskripten, sondern an der Total Cost of Ownership der Applikationsentwicklung und -evolution sowie am Nutzen und Anwenderfeedback am Markt.

Drittens haben wir erneut wesentliche Aspekte der Testautomatisierung eingearbeitet, wie zum Beispiel die

Rolle von Testautomatisierung im Kontext von Systemen mit künstlicher Intelligenz oder im DevOps-Umfeld.

Wird mit Testautomatisierung automatisch alles besser? Nein! Eine Fertigungsmaschine, die falsch justiert ist, liefert nur Ausschuss. Wenn sie unzulänglich bedient wird, erzeugt sie zufällige und unbrauchbare Ergebnisse. Wenn sie nicht oder unpassend gewartet wird, kommt sie zum Stillstand oder wird gar unbrauchbar. Ausgebildete Mitarbeiterinnen und Mitarbeiter, nachhaltige Konzepte, das Bewusstsein, dass Testautomatisierung ein wesentlicher Produktionsfaktor ist, und der verantwortungsvolle Umgang mit diesem sind Voraussetzungen dafür, dass die Potenziale und Möglichkeiten dieser Technologie auch tatsächlich realisiert werden können. Und dies ist in den meisten Fällen auch erfolgskritisch, denn Testautomatisierung ist unabdingbar, um in agilen Projektumwelten stabile Qualität liefern zu können, um mit der Geschwindigkeit moderner Continuous-Delivery-Verfahren Schritt zu halten und gleichzeitig die langfristige Wirtschaftlichkeit von Softwareentwicklungsprojekten insgesamt zu gewährleisten.

Für die Umsetzung in Ihrem Unternehmen wünschen wir Ihnen viel Erfolg. Begleitende Informationen zu diesem Buch und weitere wertvolle Hinweise zum Thema Testautomatisierung finden Sie auf unserer Internetseite [www.software-test-automation.at](http://www.software-test-automation.at).

## **Danksagung**

Die Autoren danken den tatkräftigen Unterstützern Michael Hombauer, Sonja Baumgartner, Dominik Schildorfer, Anita Bogner, Christian Mastnak, Roman

Rohrer, Martin Schweinberger, Stefan Denner, Stephan Posch, Yasser Aranian, Georg Russe, Vincent Bayer, Andreas Lenich, Cayetano Lopez-Leiva, Bernhard König, Jürgen Pointinger sowie dem Unternehmen Nagarro.

*Manfred Baumgartner*, Wien 2020

*Stefan Gwihs*, Wien 2020

*Richard Seidl*, Essen 2020

*Thomas Steirer*, Brunn am Gebirge  
2020

*Marc-Florian Wendland*, Berlin 2020

# Geleitwort zur 3. Auflage

Die zweite Welle ist da! Ich denke, wir befinden uns gerade mitten in der zweiten Welle der Testautomatisierung. Die große erste Welle habe ich in den ersten Jahren der 2000er-Jahre beobachten können. Die Projekte hatten zunächst große Erfolge zu verbuchen im Sinne einer Verbesserung von Effektivität und vor allem Effizienz der Testprozesse in punktuellen Einsatzbereichen. Allerdings, ganz im Sinne des Gartner-Zyklus: Das »Tal der Enttäuschungen« war schnell erreicht und das »Plateau der Produktivität« wurde - in meinem Sichtfeld - von der Mehrheit dann nicht erklimmen.

Was ich damals beobachten konnte, waren Projekte, die sich über mehrere Jahre und mit enormem Aufwand zu einem hohen Grad an Testautomatisierung vorgearbeitet hatten. Und dann kamen Technologiewechsel wie der Umstieg zu .NET-Plattformen oder Prozesswechsel wie die Umstellung auf Agile. Und viele der Testautomatisierungsframeworks haben diese Umstellungen nicht überstanden. Also bin ich in dieser Zeit gerne mit Vorträgen an die Öffentlichkeit gegangen, die provokante Titel trugen, wie »Testautomatisierung schlägt immer fehl«.

Zwei Kernprobleme waren zu beobachten: Erstens sind Organisationen daran gescheitert, die punktuellen Erfolge auf das gesamte Projekt oder die Organisation zu skalieren.

Und zweitens waren Testautomatisierungs-Plattformen nicht adäquat in der Lage, disruptive Änderungen an der technologischen Basis flexibel aufzufangen.

Kein Wunder also, dass das Thema Testautomatisierung mit der Zeit an Akzeptanz verlor. Hier spielen dann auch Managementaspekte eine tragende Rolle. Auf lange Sicht konnten die hohen wirtschaftlichen Erwartungen einer einmaligen Investition, die dann Regressionsaufwände deutlich reduziert, oft nicht erfüllt werden.

Seit Mitte des 2. Jahrzehnts können wir nun eine weitere Trendwelle der Testautomatisierung in den Projekten beobachten. Wird die Testautomatisierung nun wieder unter ihren Erwartungen bleiben? Ich glaube nein. Einerseits haben sich die Rahmenbedingungen für die Automatisierung von Tests geändert und andererseits die Erwartungen, die daran gestellt werden. Testautomatisierung hat sich inzwischen wieder als unverzichtbarer Erfolgsfaktor von Projekten in den aktuellen technologischen Szenarien etabliert. Warum dieser Unterschied?

Mit dem Einzug von agilen Prozessen haben sich hoch automatisierte und werkzeuggestützte Entwicklungsprozesse mittlerweile als Standard etabliert und deutlich weiterentwickelt. Continuous-Integration-Konzepte werden stetig zu DevOps-Prozessen weiterentwickelt, um eine nahtlose Plattform für die Integration automatisierter Projektschritte von der Idee bis zur Produktion und zum Betrieb zu schaffen. Die durchgehende Automatisierung von Prozessen bildet damit in natürlicher Weise eine hervorragende Basis zur Integration der Testautomatisierung in den Gesamtprozess. Die Skalierung von Prozessen hat mit dem agilen Vorgehen einen neuen, hohen Stellenwert erreicht. Dies ist eine Entwicklung, die für die Einführung und langfristige

Etablierung von Testautomatisierungslösungen ein essenzieller Erfolgsfaktor ist.

Ein wesentlicher Faktor für die Bedeutung und Notwendigkeit der Testautomatisierung ist aber die technologische Plattform, auf der wir uns derzeit bewegen. Disruptive Technologien wie IoT und künstliche Intelligenz drängen aus ihren jahrzehntelangen Nischen rasant in die Breite in unsere Produkte. Dies bringt eine deutliche Verschiebung der Prioritäten für die Qualitätsmerkmale mit sich, die wir testen müssen. Während vor 20 Jahren noch 90% aller Tests funktionale Tests waren, setzt sich die Bedeutung der nicht funktionalen Tests wie Usability, Performanz, IT-Sicherheit usw. langsam, aber sicher durch. Die Anzahl der Testfälle, die zur Bewertung der Produktqualität erforderlich sind, steigt daher rapide an, und nur mit automatisierten Tests können Qualitätsmerkmale wie die Performanz effektiv abgesichert werden.

Die Entwicklung und Wartung von Produkten erfolgt in immer kürzer werdenden Zeitintervallen. Aufgrund der steigenden Varianz in den Hardware- und Softwarekonfigurationen müssen die (Gesamt-)Systeme in einer steigenden Anzahl an Varianten getestet werden. Ein nicht automatisierter Regressionstest wird somit zunehmend zu einer Belastung für das Projekt; beziehungsweise es wird immer schwieriger, die geforderte Testabdeckung mit adäquatem Aufwand zu erreichen.

Und glücklicherweise haben wir auch methodisch dazugelernt: Testarchitekturen als ein wichtiger, wenn nicht der wesentlichste Faktor für die Qualität in der Wartbarkeit der automatisierten Tests sind mittlerweile so gut etabliert, dass Organisationen die Rolle eines Testarchitekten einführen. Dies nur als Beispiel. Aber Vorsicht: Das richtige Vorgehen und das Wissen über

Fallstricke und Best Practices bei der Einführung und Pflege der Testautomatisierung sind ein Schlüssel zum nachhaltigen Erfolg. Entsprechende Expertisen in die Projekte und die Organisation zu bringen, ist nicht immer einfach. Hier unterstützt das Zertifizierungsschema des Certified Tester, das in der Community schon seit Langem als Standard und als gemeinsames Glossar etabliert ist. Der für fortgeschrittene Tester gedachte »Test Automation Engineering«-Kurs, den dieses Buch begleitet, setzt die Schwerpunkte und Erfolgsfaktoren einer nachhaltig erfolgreichen Testautomatisierung in einen Kanon an Expertisen um - wie zum Beispiel über Testautomatisierungsarchitekturen. Und das vorliegende Buch mit seinen vielen Verbesserungen und Änderungen zur zweiten Auflage zeigt deutlich, dass sich dieser Kanon an Skills ständig weiterentwickelt.

Wir sind also gut gerüstet und meines Erachtens mit den Themen der Testautomatisierung einen deutlichen Schritt weiter, und ich wünsche Ihnen gutes Gelingen und auch kreativen Spaß dabei, Testautomatisierung als Schlüsselfaktor für Ihren professionellen Erfolg einzusetzen!

*Dr. Armin Metzger*  
Geschäftsführung  
German Testing Board, 2020

# Geleitwort zur 2. Auflage<sup>1</sup>

Seit es uns Menschen auf diesem Planeten gibt, haben wir immer danach getrachtet, mithilfe von Werkzeugen unser Leben zu vereinfachen, Aufgaben schneller zu erledigen und unsere Fähigkeiten zu erweitern. Vom Faustkeil des Steinzeitmenschen über die Erfindung des Rades bis hin zu den hoch technisierten Produktionsmaschinen von heute: Werkzeuge sind ein integraler Bestandteil und Ausdruck unserer Evolutionsgeschichte. Daher ist es kein Wunder und nur ganz natürlich, dass wir solche auch beim Testen von Software einsetzen möchten. Allerdings gibt es über den natürlichen Instinkt hinaus noch weitaus überzeugendere Gründe für den Einsatz von Testwerkzeugen.

Das Erfordernis, Software in besserer Qualität immer schneller und billiger liefern zu können, treibt die Suche nach einfacheren Methoden des Testens voran. Werkzeuge sind dabei eine der effektivsten Möglichkeiten, genau diese Ziele zu erreichen.

Testen ist eine herausfordernde Tätigkeit, die sowohl Denken als auch Aufwand erfordert. Das Denken ist notwendig, um einerseits das zu testende System zu analysieren und andererseits zu identifizieren und zu priorisieren, was getestet werden soll. Darüber hinaus müssen Testfälle geschrieben werden, die Fehler aufzeigen, und der Tester muss entscheiden, wann diese Tests und

etwaige Wiederholungen durchgeführt werden. Ohne entsprechende Gedankenarbeit wird der Test kaum erfolgreich Fehler finden und lediglich dazu führen, dass er als ineffektiv, zu teuer und aufwendig wahrgenommen wird.

Softwaresysteme sind immer größer und komplexer geworden. Dies gilt in gleicher Weise für die Anforderungen an das Testen. Das Wachstum für den Testbedarf verläuft somit nicht linear, sondern exponentiell. Wenn zu einem bestehenden System ein neues Feature hinzugefügt wird, ist nicht nur dieses Feature und seine Funktionsfähigkeit zu testen, sondern auch die kombinatorische Explosion der Interaktionen zwischen dem bestehenden System und dem neuen Feature. Und mit jeder Erweiterung wächst die Anzahl möglicher Interaktionen.

Eine Veränderung in einem Teil der Software kann zu neuen Fehlfunktionen führen bzw. bisher unbekannte Defekte in anderen Teilen der Software aufdecken. Daher darf sich der Testumfang nicht nur auf Regressionstests der Veränderung selbst beschränken, sondern muss das gesamte System, inklusive Interaktion zwischen bestehendem System und neuen Features, berücksichtigen. Darüber hinaus müssen viele Tests für mehrere Umgebungen wiederholt werden. Tests werden nicht nur einmal durchgeführt, sondern mehrmals mit jeder neuen Version der Software über die gesamte Nutzungsdauer hinweg.

Diese repetitive Arbeit kann einen solch großen Anteil am Testeinsatz einnehmen, dass für den kreativen Teil des Testens nur mehr wenig Raum bleibt. Darunter leidet die Qualität des Tests und in der Konsequenz die Qualität der Software.

Es ist daher nur verständlich, dass Werkzeuge insbesondere dort eingesetzt werden, wo Tests immer und

immer wieder erneut durchgeführt werden müssen – und mag es nur dem Ziel dienen, ausreichendes Testen innerhalb eines akzeptablen Zeitrahmens sicherzustellen. Darüber hinaus vermag der Werkzeugeinsatz dabei zu helfen, immer gründlichere Regressionstests innerhalb eines vorgegebenen Zeitraums durchzuführen. Am reizvollsten ist jedoch der Aspekt, dass Werkzeuge die Regressionstests übernehmen und den Testern damit mehr Freiraum für die kreativen Aspekte des Testens verschaffen. Damit wird neben der Qualität des Tests auch die Effizienz und Effektivität vom Test als Gesamtheit erhöht (der Werkzeuggebrauch kann dabei helfen, mehr Fehler zu finden).

Testwerkzeuge für den Regressionstest zu nutzen, ist nur der erste und augenscheinlichste Schritt, wenn es darum geht, Werkzeuge einzuführen, die letztlich das gesamte Spektrum der Testaktivitäten unterstützen können.

Zusätzlich zu den Vorteilen, die durch die Automatisierung von Regressionstests erreicht werden können, helfen uns Testwerkzeuge dabei, Tests durchzuführen, die rein manuell nicht möglich sind. Damit unterstützen sie uns, detaillierter, tiefgreifender und vielfältiger zu testen.

Es ist leicht nachzuvollziehen, dass der Werkzeugeinsatz im Testbereich großes Potenzial hat. Die Verwendung von Werkzeugen ermöglicht es, Tests mit kürzeren Zeitaufwänden durchzuführen. Außerdem können Tests zu Zeiten erfolgen, zu denen menschliche Tester normalerweise nicht verfügbar sind – etwa in der Nacht und am Wochenende.

Bei all diesen überragenden Vorteilen, warum nutzt dann nicht jeder Werkzeuge für den Softwaretest?

Der Erfolg beim Einsatz von Testwerkzeugen ist nicht automatisch garantiert. In der Tat haben viele

Organisationen versucht, Testwerkzeuge einzuführen, und sind dabei kläglich gescheitert. Der unzweckmäßige und uninformede Werkzeugeinsatz kann großen Schaden anrichten. Selbst das einfachste Tool kann missbräuchlich verwendet werden. Halten Sie sich dafür ein dreijähriges Kind vor Augen, dem wir einen Hammer in die Hand geben - ein einfaches und effektives Werkzeug, das bei unsachgemäßer Handhabung jedoch sehr gefährlich und schädlich sein kann. Wenn wir nicht verstehen, wie wir Werkzeuge gut und adäquat einsetzen, kann das Endresultat darin bestehen, dass wir eher Zeit und Ressourcen verschwenden, als sie zu sparen. Falsch eingesetzte Tools können zudem falsche oder irreführende Informationen liefern.

Das Problem ist, dass Testwerkzeuge - trotz ihres Namens - uns das Testen nicht abnehmen. Sie sind letztlich Werkzeuge, leblose Dinge, des Denkens und der kreativen Analyse nicht fähig. Sie können nicht die Verantwortung für das Testen übernehmen. Sie können lediglich die Aktivität der handelnden Personen unterstützen. Ohne menschlichen Einsatz haben sie keinen Wert.

Um Testwerkzeuge erfolgreich einsetzen zu können, müssen Testautomatisierungsentwickler, Tester und Testmanager die Einsatzmöglichkeiten und Beschränkungen eines jeden verwendeten Werkzeugs kennen. Ein verbreiteter Grund von Testautomatisierungsversagen liegt in unrealistischen Erwartungen begründet. Beispielsweise wird ein Testmanager, der davon ausgeht, im ersten Monat nach Einführung eines Testwerkzeugs doppelt so effektiv bei lediglich der Hälfte der Kosten zu sein, sehr enttäuscht werden.

Testmanager müssen auch den Unterschied zwischen Test- und Automatisierungsfähigkeiten verstehen. Um

automatisierte Testfälle zu schreiben braucht es Zeit, Wissen und Fähigkeiten, die sich grundlegend von jenen unterscheiden, die notwendig sind, um zu testen. In gleicher Weise sollten sich die Verantwortlichkeiten für das Testen und die Automatisierung unterscheiden. Testmanager müssen aus ihrer Rolle heraus sicherstellen, dass Einzelpersonen wissen, wann sie sich auf die Automatisierung von Tests und wann auf das Testen als solches konzentrieren sollen.

Testwerkzeuge einzuführen braucht Zeit und führt letztlich dazu, dass sich Testprozesse ändern. Diese Veränderung muss kontrolliert stattfinden, um zu gewährleisten, dass ein koordinierter und konsistenter Zugang die erwünschten Vorteile des Werkzeugeinsatzes sicherstellt und gleichzeitig Fallstricke vermeidet.

Erfolg in der Testautomatisierung bedeutet den Einsatz derselben im Rahmen einer gesamtheitlichen Teststrategie. Damit ist Werkzeugunterstützung und nicht die Übernahme der Testtätigkeiten durch Werkzeuge gemeint.

Qualifizierte Menschen werden immer ein integraler Bestandteil des Testprozesses sein. Jedoch führen die wachsende Größe und Komplexität unserer Systeme und der gestiegene Anspruch an hohe Qualität in kurzer Zeit und mit geringen Kosten dazu, dass der Einsatz von Testwerkzeugen nicht mehr optionaler, sondern essenzieller Bestandteil eines reifen Testprozesses ist.

Wie können wir sicherstellen, diese Werkzeuge gut und weise einzusetzen? Das ist der Sinn und Zweck des vorliegenden Buches.

Egal, ob Sie erst mit Testautomatisierung beginnen wollen oder nach Wegen zur Verbesserung bereits bestehender Automatisierungsansätze suchen - hier sind Sie richtig. Lesen Sie weiter und lernen Sie von den

Autoren – erfinden Sie das Rad nicht neu. Nutzen Sie die aus jahrelanger Erfahrung destillierte Weisheit.

Ich wünsche Ihnen viel Erfolg bei Ihren Automatisierungsvorhaben!

*Mark Fewster*

Grove Software Testing Ltd., 2015

# Inhaltsübersicht

- 1 Einführung in die Testautomatisierung und ihre Ziele**
- 2 Vorbereitungen für die Testautomatisierung**
- 3 Die generische Testautomatisierungsarchitektur**
- 4 Risiken und Eventualitäten bei der Softwareverteilung**
- 5 Berichte und Metriken**
- 6 Überführung des manuellen Testens in eine automatisierte Umgebung**
- 7 Verifizierung der Testautomatisierungslösung**
- 8 Fortlaufende Optimierung**
- 9 Ausblick**

## **Anhang**

---

- A Softwarequalitätsmerkmale**
- B Last- und Performanztest**

**C Kriterienkatalog zur Testwerkzeugauswahl**

**D Glossar**

**E Abkürzungen**

**F Quellen**

**Stichwortverzeichnis**

# **Inhaltsverzeichnis**

## **1 Einführung in die Testautomatisierung und ihre Ziele**

- 1.1 Einleitung
  - 1.1.1 Standards und Normen
  - 1.1.2 Der Einsatz von Maschinen
  - 1.1.3 Mengen und Massen
- 1.2 Was ist unter Testautomatisierung zu verstehen?
- 1.3 Ziele der Testautomatisierung
- 1.4 Erfolgsfaktoren für die Testautomatisierung
  - 1.4.1 Testautomatisierungsstrategie
  - 1.4.2 Testautomatisierungsarchitektur
  - 1.4.3 Testbarkeit des SUT
  - 1.4.4 Testautomatisierungsframework

- 1.5 Exkurs: Teststufen und Projektarten
  - 1.5.1 Testautomatisierung auf unterschiedlichen Teststufen
  - 1.5.2 Einsatzgebiet nach Projektart

## **2 Vorbereitungen für die Testautomatisierung**

- 2.1 SUT-Faktoren mit Einfluss auf die Testautomatisierung
- 2.2 Bewertung und Auswahl von Werkzeugen
  - 2.2.1 Verantwortlichkeiten

- 2.2.2 Exkurs: Evaluierung von Automatisierungswerkzeugen
- 2.2.3 Exkurs: Evaluieren leicht gemacht

- 2.2.4 Typische Herausforderungen
- 2.3 Auslegung auf Testbarkeit und Automatisierung

### **3 Die generische Testautomatisierungsarchitektur**

- 3.1 Einführung in die generische Testautomatisierungsarchitektur (gTAA)
  - 3.1.1 Warum eine gute Testautomatisierungsarchitektur so wichtig ist
  - 3.1.2 Entwicklung von Testautomatisierungslösungen
  - 3.1.3 Die Schichten der gTAA
  - 3.1.4 Projektmanagement einer TAS
  - 3.1.5 Konfigurationsmanagement einer TAS
  - 3.1.6 Unterstützung des Testmanagements und anderer Zielgruppen
- 3.2 Der Entwurf einer TAA
  - 3.2.1 Grundlegende Fragestellungen
  - 3.2.2 Welcher Ansatz zur Automatisierung von Testfällen soll unterstützt werden?
  - 3.2.3 Welche technischen Überlegungen zum SUT sind zu beachten?

- 3.2.4 Überlegungen zu Entwicklungs- und Qualitätssicherungsprozessen
- 3.3 TAS-Entwicklung
  - 3.3.1 Kompatibilität zwischen TAS und SUT
  - 3.3.2 Synchronisierung zwischen TAS und SUT
  - 3.3.3 Wiederverwendbarkeit in einer TAS
  - 3.3.4 Unterstützung verschiedener Zielsysteme
  - 3.3.5 Exkurs: Realisierung in unterschiedlichen Vorgehensmodellen und Methoden

## 4 Risiken und Eventualitäten bei der Softwareverteilung

- 4.1 Auswahl des Testautomatisierungsansatzes und Planung von Verteilung/Rollout
  - 4.1.1 Die Erprobung oder der Pilotversuch
  - 4.1.2 Die Verteilung oder das Deployment
- 4.2 Strategie für die Bewertung und Begrenzung von Risiken
  - 4.2.1 Spezifische Risiken bei der Erstverteilung
  - 4.2.2 Spezifische Risiken bei der Wartungsverteilung
- 4.3 Wartung der Testautomatisierung
  - 4.3.1 Auslöser und Arten von Wartungsaktivitäten
  - 4.3.2 Überlegungen zur Dokumentation der automatisierten Testmittel
  - 4.3.3 Der Umfang von Wartungsaktivitäten
  - 4.3.4 Wartung von Fremdkomponenten
  - 4.3.5 Wartung von Schulungsmaterial

### 4.3.6 Verbesserung der Wartbarkeit

- 4.4 Exkurs: Einsatzgebiet nach Systemarten
  - 4.4.1 Desktop-Applikationen
  - 4.4.2 Client-Server-Systeme
  - 4.4.3 Webapplikationen
  - 4.4.4 Mobile Applikationen
  - 4.4.5 Webservices
  - 4.4.6 Data Warehouse
  - 4.4.7 Dynamische GUIs: Formularlösungen
  - 4.4.8 Cloud Based Systems
  - 4.4.9 Künstliche Intelligenz und Machine Learning

## 5 Berichte und Metriken

- 5.1 Exkurs: Metriken und Validität
- 5.2 Beispiele für Metriken
- 5.3 Konkrete Implementierung und Realisierbarkeit in einer TAS
  - 5.3.1 Exkurs: TAS und SUT als Quellen für Protokolle
  - 5.3.2 Exkurs: Zentralisierte Verwaltung und Auswertung von Protokollen
  - 5.3.3 Implementierung der Protokollierung in einer TAS
- 5.4 Erstellung von Berichten zur Testautomatisierung