

[Editorial]



In dieser Ausgabe begrüßen wir einen neuen Goldpartner des ATB, die Seqis Software Testing GmbH. Anhand der wachsenden Zahl von professionellen Anbietern der ISTQB® Kurse erkennt man den immer größer werdenden Stellenwert des Tests und der Qualitätssicherung in der Softwareentwicklung. Neu ist auch eine Kooperation mit dem ASQF (Herausgeber des SQ-Magazins), die in Zukunft noch mehr Informationen für die Testing Community erwarten lässt.

Bereits fix etabliert sind die ATB Expertentreffs, bei denen an die Referenten stets die ATB Awards verliehen werden. Über die Produktion dieser Awards im Rahmen einer tollen Initiative des Lebenswerks Mödling berichtet ein lesenswerter Artikel. Nochmals danke an alle, die dem ATB Charity-Punsch zu einem tollen Ergebnis verholfen haben - über die Verwendung der Spenden wird ebenso berichtet.

Und natürlich gibt es jede Menge Fachbeiträge zu verschiedenen Themen: Agiler Test wird aus verschiedenen Aspekten betrachtet, auch die in Entstehung begriffenen ISTQB® Extensions (mit kritischen Anmerkungen). Weitere Beiträge befassen sich mit Mobile Apps, mit modellbasierter Testautomatisierung, mit Test Driven Development und mit TestSPICE.

Ich bin zuversichtlich, dass für jeden Tester und Testmanager etwas Interessantes dabei ist. Gut Test!

◀ (Karl Kemminger)

[Das ATB hat einen neuen Goldpartner]

Über  **Software Testing GmbH**

Die Kernkompetenz dieses IT-Consulting Unternehmens ist das Arbeiten für die Softwarequalität. Rund um die drei Geschäftsführer, Alexander Vukovic, Alexander Weichselberger und Reinhard Salomon sind es mittlerweile 25 Consultants, die sich dem Testen von Software und allen damit verbundenen Aufgaben verschrieben haben.

So unterstützt SEQIS nationale und internationale Kunden bei den verschiedensten Software Projekten. Ob im Testmanagement, im Requirements Engineering, Test-Consulting, bei der Testdurchführung, bei Schulungen oder bei der Auswahl eines geeigneten Testtools – bei SEQIS findet man immer die richtige Unterstützung.

**Das Leistungsportfolio der SEQISANER im Überblick:**



**Consulting:** Das Know-How der Testspezialisten wird bei der Testprozessberatung genau so hinzugezogen wie beim On-Demand Coaching oder

beim vorübergehenden Management eines IT-Projektes. Aufgrund der langjährigen Erfahrung sind komplexe Herausforderungen wie z.B. der Aufbau kompletter Testabteilungen oder interimistische Führung der Testabteilungen beim Kunden mehrfach praktisch erprobt.



**Outsourcing:** Ob direkt vor Ort beim Kunden oder im topausgestatteten Testlab der Firma. Im Testlab werden die gängigsten Tests umgesetzt. Lasttests oder diverse funktionale Tests, ob manuell oder automatisiert, die SEQIS Testprofis führen sie durch.



ISTQB, ISQI und IREB Ausbildungen, Methodenschulungen und Fachtesterausbildungen bilden das Fundament des **Education** Portfolios. SEQIS Trainer können auf Erfahrungen aus den vielfältigen Kundenprojekten zurückgreifen. Somit wird nicht nur theoretisches Wissen gelehrt, sondern es wird auf Best Practices aufgebaut.



Im SEQIS TCC, dem **Testtool Competence Center**, werden Testtools von vielen kommerziellen Anbietern und OpenSource genau unter die Lupe genommen. Spezialisten befassen sich mit der Evaluierung der Vor- und Nachteile der Programme. Diese Kenntnisse und die Produktunabhängigkeit von SEQIS ermöglichen eine optimale Beratung der Kunden.

**Mehr Informationen über SEQIS finden Sie hier:** [www.SEQIS.com](http://www.SEQIS.com)



[Inhalt]

■ Goldpartner	Seite 1 - 4
■ Mobile Apps	Seite 4 - 7
■ Agiler Test	Seite 8 - 9
■ Modellbasierte Testautomatisierung	Seite 10 - 14
■ Test Driven Development	Seite 15
■ ASQF, iSQI	Seite 16 - 17
■ ATB Expertentreff	Seite 18
■ ATB Award & Charity	Seite 19 - 20
■ Seminare	Seite 21 - 23
■ Impressum	Seite 22
■ Konferenzen, Silberpartner,	Seite 24
■ TestSPICE	Seite 25 - 28

## BDC-Trainings: Alles neu macht der Mai



- **First come, first serve: Unsere Frühbucherrabatte:** Bei Anmeldung **bis vier Wochen vor** Trainingsbeginn zahlen Sie für **ISTQB® CT - FL** nur **€ 1.450,-**.
- **Nächste Termine:**

<b>ISTQB® Certified Tester - Foundation Level:</b>	2.6.2014 - 5.6.2014 (Wien)
<b>CAT® Certified Agile Tester:</b>	5.5.2014 - 9.5.2014 (Wien)

Zusätzlich bieten wir die Trainings

- Software Test für IT-Manager,
- Software Test für Embedded Systems sowie **neu**
- **Automated Testing** und
- **Agile Testing** an.

>> [Details & Termine](#)

- **Inhouse-Trainings** bieten wir **österreichweit** an.

## SQS Academy - Für jede Zielgruppe das richtige Seminar



sqs.com

### Aktuelles Vorteils-Angebot

Tolle Aktionspreise auf ausgewählte Seminare für das erste Halbjahr 2014.

Buchen und sparen Sie jetzt!

Nähere Informationen finden Sie [hier](#)



### Lernen mit den Experten

SQS ist der weltweit führende Spezialist für Software-Qualität. Position und Kompetenz der SQS als Marktführer sind auch im Trainings-Bereich das Ergebnis von mehr als 30 Jahren Beratungsaktivität. Die Stärke der SQS-Seminare liegt in der Expertise der Trainer.

### Kommende Seminartermine der SQS in Wien:

- ISTQB® CT Foundation Level, 02.-05.06.
- IREB - CPRE - Foundation Level, 23.-25.06
- ISTQB® CT Foundation Level, 01.-04.09.
- ISTQB® CTAL Test Manager, 15.-19.09.
- Certified Agile Tester® (CAT), 29.09.-03.10.
- ISTQB® CT Foundation Level, 06.-09.10.
- ISTQB® CTAL Test Analyst, 17.-20.11.

Alle Seminare von A-Z finden Sie [hier](#)

**Wir sind gerne für Sie da!** Telefonisch: +43 (0) 1 319 35 23-13 oder per E-Mail: [academy-austria@sqs.com](mailto:academy-austria@sqs.com)



Hier bloggen Mitarbeiter. Mit Expertenbeiträgen bieten die Autoren Erfahrungen und Tipps für alle Bereiche. Der ANECON Blog berichtet über Methoden, Prozesse und Tools aus den Gebieten Software-Entwicklung, Software-Test und Projektmanagement. Darüber hinaus informieren wir rund um Karriere, Veranstaltungen und IT-Trainings. Hier sind alle richtig, deren Alltag direkt oder indirekt durch IT beeinflusst ist. **Check IT out!** [www.anecon.com/blog](http://www.anecon.com/blog)



Bezahlte Einschaltung

## Aktuelle Seminartermine



ISTQB® Certified Tester Foundation Level (4-tägig)  
» 12.–15.05.2014: [Lustenau](#)

Software Usability (3-tägig)  
» 20.–22.05.2014: [Wien](#)

IREB® CPRE Foundation Level (3-tägig)  
» 20.–22.05.2014: [Wien](#) [Linz](#) [München](#)

Aufwandsschätzung in Softwareprojekten (2-tägig)  
» 10.–11.06.2014: [Wien](#) [Linz](#) [Graz](#) [Lustenau](#)

» [alle Seminare](#)

## Auszeichnung bei Great Place to Work® 2014



### Software Quality Lab ist ein Great Place to Work®

Im Rahmen der großen Great Place to Work® Award Ceremony im März 2014 wurde Software Quality Lab der Preis für den 3. Platz in der Kategorie „Small“ (Unternehmen mit 20-49 Mitarbeitern) verliehen. Wir freuen uns sehr über diese besondere Auszeichnung!

Bezahlte Einschaltung



## Call for Papers

Reichen Sie Ihre Vorträge und Workshops online ein!

20. – 22.01.2015  
Hotel Savoyen, Wien

» [Einreichen](#)

A word cloud centered around the logo 'SEQSIS'. The logo consists of the letters 'S', 'E', 'Q', 'S', 'I', 'S' in a stylized font, with the 'Q' being a large, orange-to-red gradient circle. Surrounding the logo are various terms related to software testing and development, including: 'Anforderungsmanagement', 'Mobile testing', 'Testlab', 'Testtool Vergleich', 'TDD', 'IIBA CAT', 'ISTQB', 'InitTests', 'xecution', 'Testdriven', 'Know-How', 'Agile Tester', 'Coaching', 'Testmanagement', 'Loadtest', 'Business Analyse', 'Agil', 'Testprozess', 'Testautomation', 'Requirements Engineering', 'Edu', 'Test Driven Development', 'Prozessoptimierung', 'Funktionstest', and '10 things I wished they' told me'. There is also a circular seal on the right side of the word cloud that says 'www.SEQIS.com' and '100% Qualität'.

Bezahlte Einschaltung

**[Tosca Mobile+]**  
von Georg Thurner

Die mobile Revolution hat unseren Alltag verändert – in den modernen Industrieländern besitzen viele zumindest ein Smartphone wenn nicht auch zusätzlich ein Tablet.

Glaubt man den Analysten, so werden spätestens 2015 Tablets sämtliche andere PC-Arten wie Notebooks oder klassische Standgeräte in den Verkaufszahlen überholt haben.

Nutzer mobiler Applikationen erwarten sich intuitive, einfach zu nutzende und vor allem fehlerfreie Apps. Im Gegensatz zu klassischer PC Software werden Fehler viel weniger toleriert - Apps werden bei Nichtfunktionieren gelöscht auch oder vielleicht gerade weil sie kaum etwas kosten oder gar gratis sind.

Unternehmen stehen daher in der Zwickmühle, so rasch wie möglich mobile Lösungen mit der erforderlichen hohen Qualität auf den Markt zu bringen.

Ohne robuste Testautomatisierung lässt sich dieses Unterfangen nicht bewerk-

stelligen. Moderne Applikationen sind jedoch zumeist ‚lediglich‘ neue Clients bestehender Infrastruktur. Zum Beispiel haben Banken bereits jetzt eine etablierte Onlinebanking Lösung für herkömmliche Webbrowser – die neuen mobilen Apps erweitern dieses Angebot und stützen sich dabei auf bereits existierende Schnittstellen.

Tosca Mobile+ erweitert die Tricentis Tosca Testsuite durch mobile Testautomatisierung. Damit steht Unternehmen eine komplette Testautomatisierungslösung zur Verfügung, welche es erlaubt, logische Testfälle auf allen Ebenen moderner Businessapplikationen durchzuführen.

Derselbe Testfall kann für das Testen mobiler Applikationen aber auch für das Testen der Web-Oberfläche sowie beim Testen vom Web Service Schnittstellen verwendet werden.

Mithilfe der risikobasierten Testfallgewinnung können Sie sich auf die ‚wertvollsten‘ Testfälle konzentrieren – also jene Testfälle, welche kritische Funktionalität abtesten.

Tosca Mobile+ unterstützt Android Geräte ab 2.2 sowie iOS Geräte ab 5.0. Dabei können native und hybride Applikationen auf physischen und emulierten Geräten gleichermaßen getestet werden. Tricentis' einzigartige modellbasierte Testautomatisierungslösung erlaubt auch Testern ohne technischen Background den Aufbau und die Wartung von komplexen Testszenerien, ohne zu skripten.

Lesen Sie mehr unter [www.tricentis.com/mobile-plus](http://www.tricentis.com/mobile-plus)

◀ (Georg Thurner)

**[Der Autor]**

Georg Thurner ist seit über 12 Jahren im Softwaretestingbereich tätig und ist Product Marketing Manager bei Tricentis.



## [8 Faktoren für den Erfolg Ihrer mobilen App]

von Ing. Stefan Feßl

Mobile Anwendungen sind in aller Munde. Egal ob es sich um **Business to Customer** Anwendungen, **Business to Business** Lösungen oder auch Umsetzungen im **Business to Employee** Bereich handelt, stehen Unternehmen vor zahlreichen Herausforderungen, welche es zu Bedenken gibt. Ich möchte Ihnen hier acht Aspekte näher bringen, denen Sie im mobilen Bereich besondere Beachtung schenken sollten.



### 1. Fokus auf Nutzen/Mehrwert der App legen

Jede Applikation sollte einen Nutzen stiften und zwar sowohl für den Anwender der mobilen App als auch für das Unternehmen, welches die App zur Verfügung stellt. Praktisch ist es leider so, dass viele Apps von Anwendern nur einmal gestartet werden und dann nie wieder aufgerufen werden. Steht bei Endkunden hier die Einfachheit und **Usability** im Vordergrund, ist es bei Applikationen für die eigenen Mitarbeiter hauptsächlich die Arbeitserleichterung, welche die tatsächliche Verwendung einer App begünstigt.

### 2. Passende Plattformstrategie für die mobile App wählen

Im mobilen Umfeld ist man mit unterschiedlichen Plattformen konfrontiert. So beherrschen Android und iOS den Markt, allerdings will man ja auch Windows und BlackBerry Anwender nicht ausschließen. Außerdem lässt sich der schnelllebige Mobilmarkt kaum vorhersagen. Wer hätte in der Pre-iOS Zeit gedacht, dass Palm untergeht und BlackBerry in ernsthafte Schwierigkeiten kommt? Im Zuge dieser Überlegungen stellt sich daher die Frage, ob man eher eine **HTML5** Lösung anstrebt, eventuell sogar als **responsive Designlösung** einer mobilen Website oder ob mehrere **native Implementierungen** notwendig sind.

Letztere Lösungen ermöglichen zwar den Zugriff auf alle Features der Endgeräte und Plattformen, letztlich muss man ein und dieselbe App aber für jede Plattform entwickeln, was einem bei einer HTML5 Lösung erspart bleibt. Da beide Ansätze ihre Vor- und Nachteile haben, gibt es den Versuch, die Vorteile der beiden Lösungen durch sogenannte **Hybrid Apps** zu vereinen, wodurch bis zu 80% des Codes für alle Plattformen gemeinsam genutzt werden können.

### 3. Architektur festlegen

Sobald die grundsätzliche Plattformscheidung feststeht, sollten Sie rasch die wichtigsten Architekturentscheidungen treffen. Wie sollen die einzelnen Plattformen an den **Backend** angebunden werden? Wie kann die mobile App am besten in die eigenen **Geschäftsprozesse** integriert werden? Oft ist auch eine Anpassung der Geschäftsprozesse erwünscht oder nötig, um die Anwendung mobiltauglich zu machen. Wie kann die Zahl der Schnittstellen gering gehalten werden, um Implementierungs- und Wartungsaufwand gering zu halten?

### 4. Sicherheitsaspekte beachten

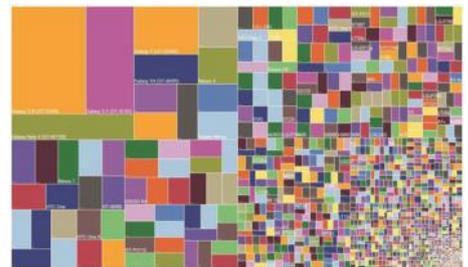


Sobald Unternehmensdaten oder Apps, welche den Zugriff auf der Unternehmensnetzwerk ermöglichen, auf einem mobilen Endgerät verfügbar sind, erlangt das Security Thema eine hohe Bedeutung. Dies beginnt beim Device Management und Themen wie "**Bring Your Own Device**" und führt bis zu Datenverschlüsselung und Absicherung der Apps innerhalb eines Containers. Führen Sie sich allerdings bewusst vor Augen, dass die 100%ige **Sicherheitslösung** für die mobile App noch nicht verfügbar ist und stets Kompromisse eingegangen werden müssen.

### 5. Usability & User Experience

Bei mobilen Apps kommt der Usability und User Experience **noch mehr Bedeutung** zu als bisher. Schlechte Usability führt rasch dazu, dass eine mobile App nicht mehr benutzt wird. Hier verzeihen Kunden viel weniger als vielleicht die eigenen Mitarbeiter. Um eine gute Usability zu erreichen, sind einerseits die gewählten Architektur- und Multiplattform-Ansätze von Relevanz, andererseits aber auch, von welcher Seite über die Geschäftsprozesse nachgedacht wurde. (Stichwort: mobile First)

### 6. Gerätevielfalt & Testherausforderung bedenken



Device Fragmentation, July 2013 (Quelle: <http://opensignal.com/reports/fragmentation-2013>)

Im mobilen Bereich gibt es eine Vielzahl unterschiedlicher Endgeräte, welche sich nicht nur in der Hardware, sondern auch in der installierten Betriebssystemversion unterscheiden. Häufig kann nicht vorhergesagt werden, für welche Geräte noch ein Update des Betriebssystems vom Hersteller veröffentlicht wird und für welche nicht. **Herstellerspezifische Anpassungen** der Software erhöhen zusätzlich die Komplexität. Im Hardwarebereich kommen unterschiedliche CPUs, Displaygrößen und Auflösungen sowie unterschiedliche Größen des Arbeitsspeichers hinzu. Dies erhöht nicht nur den Anteil an Device- und Plattformabhängigen Programmcode, sondern auch massiv die Herausforderungen beim Testen der mobilen Apps. Welche Endgeräte sollen getestet werden? Wann reicht eine Emulation aus?

### 7. Online/Offline-Datenhaltung festlegen

Wir leben in einer "always on" Gesellschaft und dennoch gibt es Situationen, in denen das gewohnte Datennetz oder WLAN nicht, oder nur eingeschränkt, verfügbar ist. >

Sei es, weil hohe Roamingkosten auf der Geschäftsreise anfallen oder weil ein Funkloch in der städtischen U-Bahn einen Datenaustausch nicht möglich macht. Wie soll sich die App hier verhalten? Wie wichtig ist die Aktualität der Daten? Sollen diese am Gerät gecacht werden? Wie soll mit der **Nicht-verfügbarkeit von Backendsystemen** umgegangen werden ohne die User Experience zu stark negativ zu beeinflussen?

## 8. Update-Strategie überlegen

Im Gegensatz zu nicht verteilten Anwendungen ist es bei Anwendungen, die über Plattform App Stores verteilt werden, nicht einfach möglich Updates sofort auf allen Endgeräten verfügbar zu machen. Grund hierfür ist die lange Zeitdauer, da eine Prüfung durch die jeweiligen Plattformstores Tage, wenn nicht sogar Wochen, in Anspruch nehmen kann. Außerdem sollten Sie bedenken, in den Applikationen eine Versions- bzw. Gültigkeitsprüfung einzubauen, um so ein Update der Applikation erzwingen zu können. Andernfalls laufen Sie Gefahr, dass alte Versionen der App im Umlauf bleiben und zu Problemen im Backend führen. Abschließend sei gesagt, dass es je nach Situation unterschiedlich passende Lösungen gibt und daher auch im mobilen Bereich ein **professionelles und geplantes Vorgehen** und die Berücksichtigung und Überlegungen zu den oben ausgeführten Punkten von zentraler Bedeutung sind.

## Checkliste für Ihre mobile App

Das 19. ANECON Expertenfrühstück informierte am 2. April rund um das Thema „Enterprise goes mobile“. Im Rahmen eines World Cafés fand ein reger Austausch an Fragen und Erfahrungen zu den größten Herausforderungen statt und die intensiven Gespräche zeigten eindrucksvoll Relevanz und Allgegenwärtigkeit des Themas. Die Ergebnisse sind als Checkliste erhältlich. Wollen auch Sie davon für Ihre mobilen Entwicklungsvorhaben profitieren? Dann fordern Sie jetzt die Checkliste kostenlos an unter: [marketing@anecon.com](mailto:marketing@anecon.com)

◀ (Ing. Stefan Feßl)

## [Selenium & Test-Automatisierung von Mobile Apps]

von Manouchehr Mirzaei Bsc.

### Einleitung:

Derzeit gehören die mobilen Lösungen zu den Faktoren, mit denen sich ein IT-Unternehmen langfristig seinen Erfolg sichern kann.

Mobile Apps sind mittlerweile im Aufbau nicht weniger komplex als Desktop-Anwendungen. Um eine hohe Qualität sichern zu können, ist es wichtig, Apps systematisch und automatisiert zu testen. Die Gründe dafür sind einerseits, dass auf dem Markt viele mobile Endgeräte mit unterschiedlicher Hardware und diversen Betriebssystem-Versionen angeboten werden, auf denen die Apps laufen müssen. Andererseits erfordern die Entwicklungsmethoden von Apps (z. B. agile Methoden) durch kurze und iterative Entwicklungszyklen eine intensivere Test-Automatisierung.

Damit die Testautomatisierung der Apps auf den mobilen Endgeräten effizient durchgeführt werden kann, werden Automatisierungs-Tools benötigt. Selenium bietet sich dabei als ein Open-Source Tool zur Test-Automatisierung an.

### Selenium

Seit der Entstehung von Selenium im Jahr 2004 (Selenium Core) wurden seine Funktionen und Leistungen zur Automatisierung von Benutzerinteraktionen mit Browsern erweitert. Wegen der Einfachheit der Bedienung von Selenium ist es möglich, ohne Vorkenntnisse von Script-Sprachen die Features und Tools wie z. B. Selenium IDE (als Firefox Plug-In) zu benutzen. Selenium IDE wird zur Aufnahme von Benutzerinteraktionen mit dem Browser und zur Bearbeitung, Verwaltung und Ausführung von daraus erstellten Testfällen verwendet. Für verteiltes, paralleles Testen sowie Skalierbarkeit der Tests bietet Selenium die Lösung Selenium Grid.

Den großen Durchbruch hat Selenium mit der Einführung der WebDriver API in Selenium 2 geschafft. Dabei kann durch WebDriver API der Browser direkt angesprochen und die nativen Methoden des Browsers zur Automatisierung genutzt werden. Selenium ist mittlerweile auf dem Weg zum Standard in der Browser-Automatisierung zu werden, hier liegt schon ein W3C-Entwurf vor [2].

Auch die heutigen Mobile Browser unterstützen moderne Webtechnologien wie HTML5, CSS3 und JavaScript. Diese Technologien und Unterstützung von „Rich Media“-Inhalten haben es ermöglicht, viele Applikationen als webbasierte Applikation für die Ausführung in einem Mobile Browser zu entwickeln. Selenium hat das Wachstum des Mobile Webs erkannt und schon 2009 mit der Implementierung der WebDriver API für iOS (iPhoneDriver) und 2010 für Android (AndroidDriver) begonnen.

Außerhalb vom Selenium Projekt haben drei weitere (Open Source) Projekte, nämlich „Selendroid“, „Appium“ und „ios-driver“, den Ansatz verfolgt, Selenium WebDriver so zu erweitern, dass damit nicht nur Mobile Web Applikationen, sondern auch hybride und native Applikationen (Apps) automatisiert getestet werden können.

Die Idee und der Erfolg von diesen drei Projekten wurden von Selenium anerkannt und unterstützt, so dass Selenium die Weiterentwicklung von eigenen Mobile Drivern eingestellt hat und für das Mobile App Testing die Verwendung von diesen drei Frameworks empfiehlt, die alle Selenium WebDriver unterstützen [3]. Zu zwei von diesen Frameworks, nämlich Selendroid und Appium, soll in diesem Artikel einen kurzen Überblick gegeben werden.

>

## [Der Autor]

Ing. Stefan Feßl ist Kompetenzfeldleiter JAVA bei ANECON und bereits seit über 15 Jahren für das Softwarehaus tätig. Aus aktuellem Anlass gab er beim letzten ANECON Expertenfrühstück mit seinem Vortrag „Enterprise goes mobile – haben Sie an alles gedacht?“ informative Einblicke in das Trend-Thema Mobility.

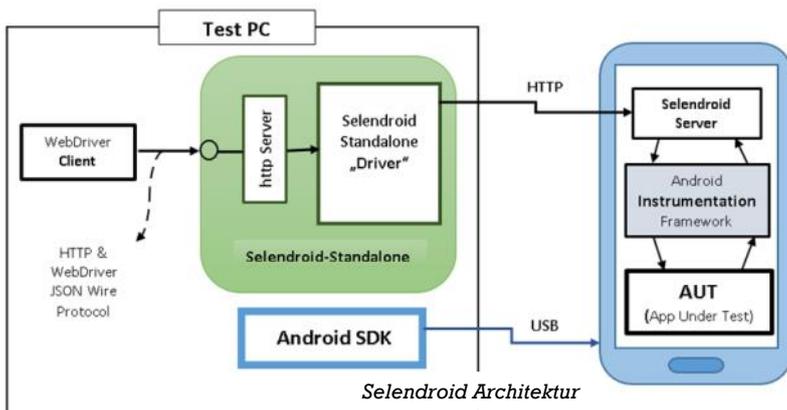
Kontakt: [Stefan.Feßl@anecon.com](mailto:Stefan.Feßl@anecon.com)



# Mobile Apps / Testautomatisierung

## Selendroid (Selenium für Android):

Selendroid ist ein Framework für die Testautomatisierung von Hybrid-, Native- und Web Apps auf Android über die Benutzerschnittstelle (UI) vom jeweiligen App-Typ. Selendroid funktioniert auf Basis des Client-Server Prinzips. Test-Skripts werden unter Verwendung von WebDriver Bibliotheken erstellt, die dann mit dem Selendroid „Standalone“-Server kommunizieren. Der „Standalone“-Server installiert und startet den Selendroid-Server auf dem Zielgerät und agiert dann als Proxy zwischen dem Test-Skript und dem Selendroid Server. Auf dem Zielgerät steuert und automatisiert Selendroid Server die Interaktionen mit dem UI der App und bringt die Testfälle zur Ausführung. Selendroid-Server verwendet das „Android Instrumentation Framework“, um die Komponenten der App (unabhängig von ihren normalen LifeCycles) zu steuern.



Selendroid bietet folgende Features und Möglichkeiten an:

- Unterstützung von vielen Programmiersprachen, wie Java, Ruby, Python und C#
- Unterstützung von Gestures
- Einen Inspektor für die Untersuchung von UI-Elementen der App und erleichterte Testfallerstellung
- Testausführung sowohl in Emulatoren als auch auf echten Mobile Devices
- Keine Modifikation der zu testenden App (Application Under Test AUT) nötig
- Unterstützung von mehreren Android Versionen (API 10 bis 19)
- Gleichzeitige Interaktion mit mehreren Android Geräten
- Hot Plugging von Android Mobile Devices
- Volle Integration als Node in Selenium Grid für Paralleltest

## Appium:

Appium ist ein weiteres Open Source Projekt, das sich auf Selenium WebDriver stützt, und die Testautomatisierung von Apps sowohl in iOS bzw. Android Plattform ermöglicht. Grundidee von Appium ist es, die Automatisierung von jeglicher Art Mobile App unter Benutzung von jedem beliebigen Testframework zu erlauben. Appium bietet u. a. folgende Vorteile und Möglichkeiten an:

- Unterstützung von vielen Programmiersprachen wie, Java, Ruby, Python, Perl, C# sowie, PHP
- Jedes Testframework kann dabei verwendet werden (JUnit, TestNG usw.)
- Hybrid-, Native- und Web Apps können automatisiert getestet werden
- Testen auf echten Geräte und Emulatoren wird unterstützt

- Integration als Node in Selenium Grid wird unterstützt
- Gestures werden unterstützt
- Ein Inspektor für die Lokalisierung von UI-Elementen
- Keine Re-Kompilierung bzw. Änderung der App nötig

Grundsätzlich ist Appium ein http-Server, der WebDriver-Sessions erzeugt und behandelt. Appium agiert als ein Proxy-Server zwischen Test-Skript und dem Mobile Device. Für die Automatisierung von Android Apps verwendet Appium das Android UiAutomator Framework, welches seit der Einführung von Android API16 (entspricht Version 4.1.x) in Android Plattform inkludiert ist. UiAutomator ist eine Java Bibliothek und beinhaltet APIs für die Testfallerstellung und eine Ausführung-Engine um die Tests automatisiert auszuführen.

Um auch ältere Android Versionen unterstützen zu können (bzw. für das Testen von webbasierten Teilen einer Hybrid-App), greift Appium auf

Selendroid zurück. Mit einer kleinen Konfigurationsänderung wird dann Selendroid in einer Appium Session gestartet.

Für Testautomatisierung auf iOS, verwendet Appium Apple's UIAutomation Framework. Appium startet das UIAutomation auf dem Zielgerät und agiert dann als ein Proxy-Server, der Anfragen (Commands) von Test-Skript über UIAutomation an die App auf dem Zielgerät weiterleitet bzw. die Antworten von der App an das Test-Skript zurück-schickt.

Mit UIAutomation alleine, können Tests nur in JavaScript erstellt und mit Apple's Tool „Instruments“ ausgeführt werden. Appium ermöglicht es aber die Tests in vielen Programmiersprachen zu erstellen und dabei jedes beliebige Testframework wie JUnit bzw. TestNG zu verwenden.

## Fazit

Der Erfolg einer Mobile Solution, in diesem Fall einer Mobile App ist dann gesichert, wenn die App die beabsichtigten und spezifizierten Funktionen erbringt. Für die Qualitätssicherung und ein effizientes App-Testing bieten die hier eingeführten Open Source Frameworks Appium und Selendroid nützliche Features und Möglichkeiten an, Apps automatisiert zu testen und all dies ohne Kosten und ohne an eine proprietäre Technologie festgebunden zu sein. Sie stützen sich auf einen freien „de facto Standard“, nämlich Selenium 2 WebDriver [2].

## Literatur:

- [1] David Burns: *Selenium 2 Testing Tools Beginner's Guide*. Packt Publishing Ltd, 2012
- [2] <https://dvc.w3.org/hg/webdriver/raw-file/default/webdriver-spec.html>
- [3] <http://seleniumhq.wordpress.com/2013/12/24/android-and-ios-support/>

◀ (Manouchehr Mirzaei)

## [Der Autor]

Manouchehr Mirzaei hat an der TU Wien Elektrotechnik und Informationstechnik studiert und ist bei BDC IT-Engineering GmbH als Software Test Engineer tätig. Er hat Berufserfahrung im Bereich Embedded Systems und im Telekommunikationsbereich.



Kontakt: [manouchehr.mirzaei@bdc.at](mailto:manouchehr.mirzaei@bdc.at)

## [Agiler Test - ein Vortrag von Tilo Linz]

von Markus Schlatzer

Am 1. April referierte der bekannte Software Test-Experte Tilo Linz beim dritten BDC Acceptance Cafe im Cafe Griensteidl über das Testen in Software Projekten die nach dem agilen Entwicklungsmodell „Scrum“ realisiert werden. In einem kurzen Repetitorium zu Beginn betonte Linz die wesentlichen Merkmale von Scrum (Sprints & Timeboxing, Product & Sprint Backlog und Transparenz) sowie dessen Ziele (Schnellere Time-to-Market und bessere Softwarequalität), um dann in einem spannenden Vortrag näher auf die Herausforderungen beim Testen in Scrum einzugehen.

Gleich vorweg: eine umfassende Abhandlung zum Thema „Testen in Scrum“ kann natürlich weder in einem Vortrag von 90 Minuten, noch in einem Aufsatz zu diesem Vortrag geboten werden. Aber das muss es auch nicht, dazu gibt es entsprechende Fachliteratur (z.B. das von Linz vor Kurzem erschienene Buch „Testen in Scrum-Projekten“). Vielmehr sollte es für Interessierte einen kurzen Einblick bieten und für Wissende die Möglichkeit, ihre Kenntnisse zu prüfen, und mit gezielten Fragen zu erweitern. Beides schaffte Linz in seinem Vortrag - natürlich auch mit der Hilfe des fragenden Publikums - zur vollen Zufriedenheit desselben.

Unter dem Publikum hielten sich die angewandten Entwicklungsmodelle (konventionell aka V-Modell oder Abwandlungen davon versus Agil und Derivate) ungefähr die Waage. Scrum selbst war entsprechend einer weiteren Publikumsbefragung zu Beginn des Vortrags hingegen dem Großteil des Publikums bekannt.

Was bedeutet also Testen in Scrum? Testen in Scrum heißt, die Testaktivitäten vom Unit- und Integrationstest bis hin zum Systemtest so gut wie möglich im Zyklus von Scrum abzudecken. Wie auch im V-Modell bilden die Anforderungen in Scrum die Basis für das Testen. Ein wesentlicher Unterschied aber ist, dass in Scrum der gemeinsame Review der Requirements aller am Entwicklungsprozess Beteiligten sicherstellt, dass ein gemeinsames Verständnis darüber herrscht, und diese korrekt und testbar formuliert wurden.

Damit nun alle Testaktivitäten vom Unit bis zum Systemtest innerhalb eines Sprints erfolgreich abgewickelt werden können, müssen die bestgeeigneten Test-Techniken angewendet werden. Für den Unit Test ist eine von diesen Techniken Test First, bei der vor der Entwicklung eines Code-Fragments (sei es eine Klasse oder eine Methode) die zugehörigen Unit-Tests entwickelt werden. Wird anschließend der Code nach und nach entwickelt, dann werden auch Schritt für Schritt die anfangs fehlschlagenden Tests erfolgreich sein. Vorteile dieser Technik sind die verbesserte Qualität der APIs sowie der Wegfall der Prosa-Spezifikationen, da die Tests als Spezifikationen dienen, und die verbesserte Testbarkeit. Des Weiteren liefern die Testfälle ein objektives Feedback für den Fortschritt der Programmierung.

Die Herausforderungen beim Integrationstest werden am besten durch Continuous Integration gemeistert. Gerade in Scrum haben diese Tests eine besondere Bedeutung, da hier das Risiko für Missverständnisse und nicht kommunizierte Codeänderungen zwischen verschiedenen Scrum-Teams besonders hoch ist. Continuous Integration mindert diese Risiken, da Fehler in Schnittstellen frühzeitig erkannt werden können.

Als Teststufe, bei der das System aus Nutzerperspektive betrachtet wird und das System möglichst der Einsatzumgebung entsprechen soll, ist der Systemtest eine ganz spezielle Herausforderung für Scrum. Aufgrund der hohen Komplexität des Systemtests stellt sich die Frage, zu welchem Zeitpunkt dieser durchgeführt werden soll, und von wem. Für eine konsequente Umsetzung des agilen Prinzips empfiehlt sich hier die Durchführung des Systemtests automatisiert und nonstop. Dabei sollen in einem Sprint für neue Features die Systemtests zuerst manuell definiert und durchgeführt werden, und im folgenden Sprint wird dann eine Auswahl getroffen, welche dieser Tests als Regressionstest automatisiert wird.

Die Anwendbarkeit des Lösungsansatzes für den Systemtest hängt im Wesentlichen von der Automatisierbarkeit der Tests ab, das wurde auch in der anschließenden Diskussion noch thematisiert.

Abschließend betonte Linz noch einmal den Punkt den erfolgreiches Testen in Scrum Projekten auszeichnet: An die Stelle sequenzieller Testphasen (mit langer Feedbackzeit) tritt kontinuierliches Testen (mit täglichem Feedback) in jedem Sprint und unter parallelem Einsatz von Tests jeder Teststufe.

Die Präsentation selbst kann übrigens von der Homepage von BDC IT Engineering GmbH heruntergeladen werden: <http://www.bdc.at/aktuelles/events.html>

◀ (Markus Schlatzer)

### [Der Autor]



Markus Schlatzer studierte Maschinenbau an der TU Wien und ist seit 2003 bei BDC IT-Engineering als Software Engineer, Test Experte und Consultant in verschiedenen Projekten aus dem Umfeld Digitale Signatur, Security und Finanzdienstleistungen tätig.



Der Vortragssaal war wieder gut gefüllt



Tilo Linz in Diskussion mit dem Publikum

## [ISTQB goes Agile]

von Martin Klonk

Der Erfolg des Certified Agile Tester (CAT), des agilen Ausbildungsprogramms des iSQI, hat beim ISTQB nun zum Umdenken geführt und dabei auch das bisherige Zertifikatsprogramm schlanker gemacht. Es gibt nun für Zertifizierte des Foundation Levels die Möglichkeit, sog.

„Extensions“ (Erweiterungen) ihres Zertifikats um spezielle Themenstellungen zu absolvieren. Dazu ist es nicht notwendig erst den Advanced Level zu erreichen. Während eine Extension für modellbasiertes Testen derzeit in Arbeit ist, wurde an der Extension für Agile Testing mit Hochdruck im letzten Herbst gearbeitet, so dass nun auch schon die Beta-Phase abgeschlossen worden ist. Wie es aussieht, kann ab Mai die englischsprachige Version unterrichtet und absolviert werden. An der Übersetzung arbeitet bereits auch eine Arbeitsgruppe unter Tilo Linz, an der von Seiten des ATB Alexander Weichselberger und ich teilnehmen.

Die Teilnehmer werden zunächst mit den typischen Prinzipien agiler Vorgehensweisen vertraut gemacht, wobei das Agile Manifest natürlich nicht fehlen darf. Vorgestellt werden als konkrete Vorgehensmodelle Extreme Programming, Scrum und Kanban. Wirklich konsequent weiter verfolgt wird im Kurs hingegen dann nur noch Scrum. Die Arbeitsgruppe begründet das zwar mit der Dominanz von Scrum, doch kann man getrost auch behaupten, dass man wenig Ahnung in der Arbeitsgruppe von anderen Modellen und in der ehrgeizigen Umsetzungsphase nicht mehr Zeit genug hatte, die Reviewergebnisse ordentlich einzuarbeiten. Dennoch ist dieses Kapitel recht gut abgestimmt und macht die Wesensmerkmale Agilen Denkens gut und prägnant sichtbar.

Als nächstes wird thematisiert, wie man als Tester aus sog. „klassischen“ Projektumfeldern umstellen muss, wenn man sich auf ein agiles Projekt einlässt. Es wird genau dargelegt, wie sich Rollenverständnis und Zusammenarbeit komplett ändern muss. Das Selbstbewusstsein, Tester zu sein, wird aber dennoch hoch gehalten.

Ein Tester hat eine ganz wichtige Rolle im Team als Coach und Förderer des Qualitätsgedankens. Dieser Teil des zukünftigen Trainings bereitet vor allem Neulinge in agilen Projekten exzellent auf ihre Aufgabe vor.

Im Dritten Teil des Lehrgangs ist nun alles hinein geworfen worden, was sich an speziellen Techniken in Agilen Projekten bewährt hat, wie man risikobasiert testet und wie die Werkzeugunterstützung in agilen Projekten üblicherweise aussieht. Hier fehlt der rote Faden bzw. muss vom Trainer später gut ausgearbeitet werden. Grund ist wieder einmal die Überarbeitungszeit nach dem ersten Alpha-Review. Die Arbeitsgruppe hatte ursprünglich einen 80-seitigen Syllabus mit 5 Kapiteln vorgelegt, der in 2 Tagen hätte durchgepeitscht werden müssen. Richtigerweise hat man daher die Kapitel 3-5 auf ein Kapitel und den Syllabus damit auf 50 Seiten eingedampft. In nur 3 Monaten eine reife Leistung. Leider ist dabei eine bunte Kollage der wichtigsten Themen entstanden, als welche man sie auch sehen muss. Es fehlt weder die Testpyramide (die darlegt, wieviel Testautomatisierung überhaupt an Benutzeroberflächen nötig ist), noch fehlen die vier Testquadranten von Crispin/Gregory (die zur Orientierung im Test eine gute Alternative zum V-Modell in agilen Teams bieten), noch das explorative Testen (das im Foundation Level ja noch relativ knapp behandelt wird).

Insgesamt ein löblicher Vorstoß des ISTQB in Richtung Agile Projekte, der meiner Meinung nach schon längst überfällig war. Leider unter so einem Zeitdruck zusammengestellt, dass keine Zeit mehr war, die Sicht der Außenstehenden gründlich einzuarbeiten. Was aber eine Besonderheit dieser Extension sein wird, ist, dass sie eine der ersten deutschsprachigen Zertifikate sein wird, wo doch alle Literatur zum agilen Testen bis vor kurzem nur auf Englisch zu lesen war. Interessant ist auch unsere Arbeit bei der Übersetzung (bei der die drei deutschsprachigen Boards eng zusammen arbeiten).

Viele Begriffe in der Agilen Welt sind englisch vorgeprägt und sie zu übersetzen ist nicht einfach: Während man „hardening iteration“ sicher nur ungern mit „Festigungsiteration“ übersetzen würde, so wäre „test-driven development“ sicher gut mit „testgetriebener Entwicklung“ zu übersetzen. Viele Begriffe werden im Deutschen gerne verwendet, ohne dass dem Sprecher die Bedeutung des Begriffes klar wäre.

Dennoch jeden „Markenbegriff“ gewaltsam ins Deutsche zu übersetzen ist sicher auch nicht zielführend. Auch werden wir ganze Absätze sehr frei übersetzen müssen, weil die englische Art der Herleitung oder Darlegung im Deutschen die Leser zwingen würde, zweimal zu lesen. Im Deutschen hängen wir vom Sprachgebrauch gerne Erläuterungen hinten an, während sie im Englischen gerne vorweg kommen.

Wenn alles nach Plan verläuft, werden wir im September bereits soweit sein, auch deutschsprachige Prüfungen für die Agile Extension anzubieten – vorausgesetzt die Akkreditierung der interessierten Trainingsprovider geht genauso schnell voran, wie die Übersetzungsarbeit.

◀ (Martin Klonk)

### [Der Autor]

Martin Klonk ist seit 2011 Senior Berater für Software-Test bei ANECON in Wien. Davor war er seit 1996 als Softwaretester, -testautomatisierer und -testmanager für SQS zunächst in Deutschland und später in Österreich tätig. Er trainiert seit über 12 Jahren Experten für die ISTQB Foundation- und Advanced-Zertifikate und hat als ATB-Mitglied schon in diversen Arbeitsgruppen des ISTQB mitgewirkt. Zusammen mit Manfred Baumgartner, Helmut Pichler, Richard Seidl und Siegfried Tanczos hat er 2013 das Buch „Agile Testing – Der agile Weg zur Qualität“ (Hanser Verlag) herausgebracht, das die Best Practice zum Testen in agilen Projekten auch einem deutschsprachigen Leserkreis vermittelt.

Kontakt: [martin.klonk@anecon.com](mailto:martin.klonk@anecon.com)



## [Modellbasierte Testautomatisierung: Von der Anforderungsanalyse zu automatisierten Testabläufen]

von Achim Krallmann und Markus Lingelbach

*Das in diesem Artikel beschriebene Vorgehen einer modellbasierten Testautomatisierung versucht, aktuelle Testansätze auf die Phase der Anforderungsanalyse zu übertragen. Wenn in den Fachabteilungen Anforderungen mit Hilfe von Modellen erstellt werden, ist es sinnvoll und machbar, diese Modelle so zu erweitern, dass auch Testabläufe erstellt werden können. Aus diesen modellierten Testabläufen können mit Hilfe von Generatoren Testskripte generiert werden, die dann automatisiert ablaufen.*

In der Disziplin des Softwaretestens hat sich in den letzten Jahren einiges getan. Die Testautomatisierung hat insbesondere über schlüssel- oder aktionswort-getriebener Frameworks einen so hohen Reifegrad erreicht, dass einem größeren Einsatz in Projekten nichts mehr im Wege steht. Über agile Projektmethoden ist die testgetriebene Softwareentwicklung ins Gespräch gekommen und weiter professionalisiert worden. Neben den modellbasierten Ansätzen zur Anforderungsanalyse und Softwareentwicklung gibt es nun auch das Vorgehen des modellbasierten Testens.

Allerdings sind diese angesprochenen Methoden und Techniken sehr entwicklungsintensiv. Bei der Testautomatisierung liegt dies in der Natur der Sache, bei der testgetriebenen Softwareentwicklung liegt es schon im Namen. Spätestens wenn es beim modellbasierten Testen um die Generierung von Testfällen geht, bleibt ein gewisses Entwicklungswissen unabdingbar. Somit besteht immer noch eine Kluft zwischen Fach- und Entwicklungsabteilungen, die schwer zu überbrücken ist. Wenn aber in den Fachabteilungen Anforderungen mit Hilfe von Modellen erstellt werden, ist es sinnvoll und auch machbar, diese Modelle so zu erweitern, dass auch Testabläufe modelliert werden können. Im Idealfall werden dadurch Geschäftsanalysten in die Lage versetzt, Testdurchläufe zu modellieren, die – ohne weitere Programmierung bzw. Skripterstellung – automatisiert ausgeführt werden können.

Für jeden durchzuführenden Test werden zwei Informationen benötigt:

- Der Testfall, zum Beispiel als eine Beschreibung des Durchlaufes durch eine Anwendung
- Die Testdaten.

Dabei beschreiben die Testdaten zum einen die Eingabedaten und zum anderen das erwartete Ergebnis, mit dem die Korrektheit des Testes verifiziert werden kann. Dies gilt selbstverständlich auch für automatisierte Tests – dabei werden die Testdaten meistens in Dateien (Excel) ausgelagert und der Ablauf des Tests wird als Skript programmiert, das in einem Automatisierungswerkzeug mit den jeweiligen Testdaten ausgeführt wird. Dieses Vorgehen besitzt mehrere Nachteile:

- Die Pflege der Testdaten-Dateien ist unübersichtlich und fehleranfällig.
- Der Testablauf ist nicht dokumentiert, da er nur als Skript existiert.
- Änderungen und Erweiterungen am Testablauf müssen programmiert werden.
- Änderungen an den Anforderungen (Ablauflogik, Geschäftsobjekte, Oberflächen) müssen umgehend in die Testdaten und Skripte eingearbeitet werden.

Ein erster Schritt, um diese Nachteile zu umgehen, ist die Entwicklung eines Frameworks mit Schlüssel- oder auch Aktionsworten (vgl. [Seil 1]). Die Testabläufe werden dabei über sequenziell angeordnete Schlüsselwörter beschrieben (ähnlich wie in **Listing 1**).

Hinter diesen Schlüsselwörtern verbergen sich dann entsprechende Skripte, die die gewünschte Funktion ausführen. Dadurch sind die Testabläufe einfacher lesbar und auch erweiterbar. In einem nächsten Schritt bietet es sich an, die Testdaten und -abläufe zu modellieren und die entsprechenden Skripte (mit den Schlüsselwörtern) aus diesen Modellen zu generieren. Basiert auch die Anforderungsanalyse auf Modellen, so können die Testdaten mit den Geschäftsobjekten und die Testabläufe mit den Oberflächendefinitionen verknüpft werden. Dadurch können Änderungen in den Anforderungen bis in den Testablauf nachverfolgt und eingearbeitet werden. Gleichzeitig entsteht eine lesbare Dokumentation der Testabläufe mit den Testdaten. Unterstützt wird somit auch eine frühzeitige Validierung der Anforderungen, da die Testabläufe parallel mit der Anforderungsanalyse erstellt werden können. Der Hauptvorteil liegt allerdings darin, dass aus diesen Modellen die Testskripte automatisch generiert werden können.

Voraussetzung für diese modellbasierte Testautomatisierung ist neben einem Testautomatisierungswerkzeug mit einem intelligenten Framework auch ein Modellierungswerkzeug. Die Möglichkeiten, die ausgeschöpft werden können, sind grundsätzlich abhängig vom Reifegrad des eingesetzten Modellierungswerkzeuges.

Dieses Vorgehen möchten wir im Folgenden an einem Beispiel verdeutlichen. Dabei handelt es sich um eine maskenbasierte Anwendung – allerdings können die Ergebnisse ohne Weiteres auch auf Schnittstellen, Batch-Abläufe usw. übertragen werden.

>

```
Öffne Maske (Kunde.ID)
Fülle Feld (Kunde.Vorname.ID, „Karl“)
Fülle Feld (Kunde.Name.ID, „Schmidt“)
Drücke Schaltfläche (Kunde.Speichern.ID)
Validiere ist Maske offen (Kundenstatus.ID)
Validiere Feld (Kundenstatus.Status.ID, „Angelegt“)
```

Listing 1: Skript (Pseudocode)

# Modellbasierte Testautomatisierung

## Ergebnisse der Anforderungsanalyse

Die Ergebnisse einer jeglichen Anforderungsanalyse sollten zumindest die Geschäftsobjekte mit den zugehörigen Attributen sein. Bei der Definition der Oberflächen gibt es häufig einen Streit, ob dies nun zur Anforderungsanalyse oder zum IT-Design der Anwendung gehört. Für die hier angestellten Überlegungen ist dies irrelevant: Es wird davon ausgegangen, dass diese Oberflächendefinitionen existieren. Des Weiteren ist es möglich, die Attribute eines Geschäftsobjekts auf entsprechende Eingabefelder einer Oberfläche abzubilden. Im Idealfall ist das Ergebnis der Anforderungsanalyse ein Modell, wie in **Abbildung 1** dargestellt. Für eine modellbasierte Testautomatisierung müssen die Geschäftsobjekte und die Oberflächen in der Anforderungsanalyse definiert und mit einem Modellierungswerkzeug abgebildet werden. Ist dies nicht der Fall, so bleibt die Möglichkeit, aus den bestehenden, natürlich-sprachlich formulierten Fachkonzepten die entsprechenden Geschäftsobjekte und Oberflächen zu extrahieren und sie in einem Modellierungswerkzeug, wie abgebildet, zu modellieren. Sind selbst die Geschäftsobjekte nicht zu extrahieren, so kann zu jeder Oberfläche ein Datencontainer als „Geschäftsobjekt“ definiert werden. Wichtig ist dies, weil über diese Geschäftsobjekte die Struktur der Testdaten für die Testautomatisierung definiert wird.

## Beschreibung eines Testdurchlaufes

Nach der reinen Lehre sind Testfälle von den Testdaten strikt zu trennen. In einem ersten Schritt werden logische Testfälle definiert, die dann mit Testdaten in konkrete Testfälle überführt werden (vgl. [Spi12]). Im Zusammenhang mit der Testautomatisierung wird von einem Testdurchlauf gesprochen.

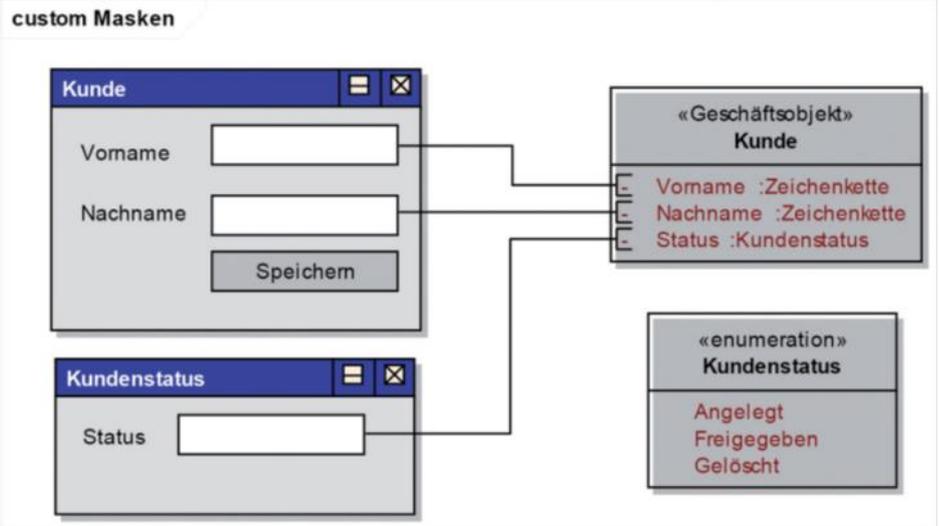


Abb. 1: Ergebnisse aus der Anforderungsanalyse

Ein Testdurchlauf entspricht dabei einem logischen Testfall, der mit genau einem Testdatum automatisiert ausgeführt wird. Zur Vereinfachung betrachten wir im Folgenden nur Testdurchläufe, die als Modell abgebildet werden – ein modellierter Testdurchlauf beinhaltet also zum einen den logischen Testfall und zum anderen auch die Testdaten.

**Abbildung 2** verdeutlicht einen beispielhaften Testdurchlauf, der die modellierten Elemente aus der Anforderungsanalyse in **Abbildung 1** wiederverwendet.

Hinter dem Testdurchlauf verbirgt sich diese fachliche Anforderung: Jeder neu angelegte Kunde hat den Status „Angelegt“. Der fachliche Testdurchlauf ist: Öffne die Oberfläche „Kunde“, trage die Testdaten aus „Kunde 1“ in die Eingabefelder ein, aktiviere die Schaltfläche „Speichern“ und überprüfe in der neuen Oberfläche, ob der Status „Angelegt“ angezeigt wird.

Entscheidend ist dabei, dass durch die Modellierung sichergestellt wird, dass bei den Testdaten auch wirklich die Attribute belegt werden, die im Geschäftsobjekt definiert sind und die dadurch mit einem Eingabefeld der Oberfläche verbunden sind. (An dieser Stelle wird die UML-Notation für Klassen und Objekte verwendet, die genau diese Beziehung definiert, vgl. [Rum10].)

Mit dieser Methode können auch Fehler-Testfälle erzeugt werden, indem zum Beispiel Attribute nicht belegt werden und eine Fehlermeldung als erwartete Maske modelliert wird.

## Generierung des Testskripts

Grundvoraussetzung für eine Generierung von Testskripten ist, dass ein Metamodell für die Modellierung von Testdurchläufen erstellt wird. Dieses Metamodell ist abhängig von der zu testenden Applikation, den Anforderungen an die Testautomatisierung, vom eingesetzten Testwerkzeug und vielen weiteren Parametern. Für den hier dargestellten Testdurchlauf wurde zum Beispiel in einem Metamodell definiert, wie das Eintragen von Testdaten in eine Oberfläche zu modellieren ist: über eine Abhängigkeitsbeziehung mit dem Kennzeichen <<fill>>.

Diese Verpflichtungen bezüglich der Notation müssen von allen Modellierern eingehalten werden, damit ein Generator die Informationen eindeutig auswerten kann. Das gilt grundsätzlich für jegliche Art der modellbasierten Methoden, da nur so sichergestellt werden kann, dass die Modelle auch von jedem Stakeholder gleichartig interpretiert und verstanden werden.

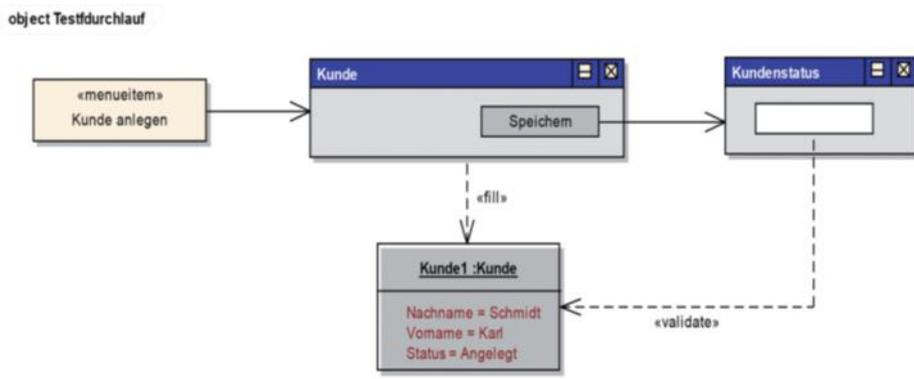


Abb. 2: Testdurchlauf

# Modellbasierte Testautomatisierung

Viele Werkzeuge für die Testautomatisierung funktionieren in zwei Schritten:

- In einem ersten Schritt werden die Elemente, die eine Oberfläche beinhaltet, ausgewertet und an einer zentralen Stelle so abgelegt, dass sie eindeutig identifiziert werden können (so zum Beispiel die GUI-Map bei dem Werkzeug „Quick Test Professional“ oder UIMap bei der Automatisierung mit „CodedUI Test für Microsoft Visual Studio“).
- In einem zweiten Schritt werden diese Elemente dann über das Erstellen eines Skripts zu einem Testdurchlauf angesprochen und bei Bedarf aktiviert, mit Werten ausgefüllt oder der Inhalt wird ausgelesen.

Beide Schritte – die Definition der Oberflächen und die Testdurchlaufskripte – können durch einen Generator erzeugt werden. Die Oberflächenelemente werden aus der Definition der Oberflächen aus dem Modell übernommen und vom Generator in eine Datei mit den Oberflächen-Definitionen geschrieben (siehe **Listing 2**). Die eindeutigen IDs für die Elemente können beliebig aus dem Modellierungswerkzeug ermittelt werden.

Dabei bietet sich die eindeutige ID an, die das Werkzeug dem Element zugeordnet hat. Wichtig ist, dass diese IDs auch von den Entwicklern der Applikation verwendet werden, damit die Elemente durch das Testautomatisierungswerkzeug gefunden werden können (siehe weiter unten).

```
Kunde.ID = M#001
Kunde.Vorname.ID = M#001F#001
Kunde.Nachname.ID = M#001F#002
Kunde.Speichern.ID = M#001#003
Kundenstatus.ID = M#002
Kundenstatus.Status.ID =
M#002F#001
```

Listing 2: Oberfläche und Elemente (Pseudocode)

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <title id="M#001">
    Kunde
  </title>
</head>
<body>
  <form action="kundenstatus.html">
    <input type="input" id=" M#001F#001" />
    <input type="input" id=" M#001F#002" />
    <input type="submit" id="M#001F#003" />
  </form>
</body>
</html>
```

Listing 3: Beispiel HTML-Quellcode

Anhand des modellierten Testdurchlaufs und des definierten Metamodells kann nun der Generator auch automatisch das Testskript erstellen. In **Listing 1** ist das zu **Abbildung 2** gehörende Skript in Pseudocode dargestellt. Dieses Skript muss natürlich an die Skriptsprache des verwendeten Automatisierungswerkzeugs, die verwendeten Schlüsselwörter und das implementierte Framework angepasst werden. Dabei bleibt ein weiterer Vorteil der modellbasierten Testautomatisierung anzumerken: Die Testdurchläufe sind sprachunabhängig und können durch konfigurierbare Generatoren auf unterschiedliche Testautomatisierungswerkzeuge abgebildet und ausgeführt werden.

## Anforderungen an die Entwicklung

Damit die entsprechenden Oberflächenelemente bei der Testdurchführung gefunden werden, müssen bei der Entwicklung der Oberflächen bestimmte Regeln eingehalten werden. Die Oberflächen werden vor der Umsetzung in der Entwicklung mit den Entwicklern besprochen. Dabei werden der Typ jedes Elements und dessen Abhängigkeit zu anderen Aktionen, wie Eingaben oder das Drücken eines Buttons, eindeutig definiert. Außerdem werden die IDs der Elemente, die später für die Zuweisung und Erkennung der Elemente einer Seite für die Testautomatisierung notwendig sind, im Modell festgelegt. Die Entwickler müssen sich bei der Umsetzung an diese Vorgaben aus dem Modell halten, wobei der Aufbau der Seite und die Platzierung der einzelnen Elemente nicht relevant für die Testautomatisierung sind, da sich der Testablauf aus den Testfällen im Modellierungswerkzeug und den dahinter liegenden IDs der Elemente ergibt. Ein Beispiel für eine HTML-Oberfläche, die die in **Listing 2** definierten Felder verwendet, ist in **Listing 3** dargestellt.

>

# Modellbasierte Testautomatisierung

## Werkzeugunterstützung und der Testskript-Generator

Die Erfolge bzw. die Einsatzmöglichkeiten der hier beschriebenen modellbasierten Testautomatisierung stehen und fallen mit der Mächtigkeit der eingesetzten Werkzeuge. Das hier verwendete Beispiel wurde mit dem „Enterprise Architect“ der Firma Sparx Systems erstellt. Dieses Modellierungswerkzeug unterstützt zum einen die Modellierung von Objekten mit Werten für die entsprechenden Attribute der zugeordneten Klasse, die zur UML Spezifikation gehört (vgl. [Rum10]). Zum anderen verfügt es über zusätzliche Eigenschaften, wie die Möglichkeit der Modellierung von Oberflächen und die Verknüpfung von Eingabefeldern zu Attributen einer Klasse (siehe **Abbildung 1** und **Abbildung 2**). Diese zusätzlichen Eigenschaften gehören allerdings nicht zur UML-Spezifikation. Somit ist bei der Auswahl eines Modellierungswerkzeugs darauf zu achten, dass auch die für die Modellierung der Testdurchläufe benötigten Elemente und Beziehungen – die wiederum im Metamodell definiert werden – unterstützt werden.

Des Weiteren sind die Auswirkungen auf den Testskript-Generator zu berücksichtigen. Dieser muss die für das Testautomatisierungswerkzeug benötigten Dateien aus den Informationen, die im Werkzeug modelliert sind, und mit Hilfe des definierten Metamodells erzeugen. Dadurch, dass das Metamodell jeweils von den definierten Testanforderungen und dem Modellierungswerkzeug abhängt, wird es mit hoher Wahrscheinlichkeit keine Standardgeneratoren auf dem Markt geben. Der Generator muss also selbst konzipiert und implementiert werden. Als Eingabe benötigt der Generator die im Werkzeug modellierten Informationen, die er entweder über eine Programmierschnittstelle zu dem Werkzeug abfragen kann, oder indem er geeignete Exportfunktionen, die das Werkzeug zur Verfügung stellt, ausnutzt. Der Weg über eine Programmierschnittstelle ist robuster, einfacher zu handhaben und somit als Standardweg zu empfehlen. Der Weg über eine Exportfunktion könnte den Charme haben, dass – falls es sich um eine XMI-Schnittstelle und somit um eine standardisierte Schnittstelle für den Austausch von UML-Modellen (vgl. [Gro02]) handelt – der Generator unabhängig vom verwendeten Werkzeug programmiert werden kann.

Das gelingt allerdings nur, wenn für die modellbasierte Testautomatisierung Modellierungselemente verwendet werden, die zum UML-Standard gehören bzw. die über den Erweiterungsmechanismus der UML-Profile abgebildet werden können (vgl. [Rum10]). Zusätzlich muss das Modellierungswerkzeug bei diesem Vorgehen auch das Modell korrekt exportieren. Der Enterprise Architect exportiert zum Beispiel die Objekte mit den Attributwerten nicht wie in der XML-Spezifikation vorgesehen, sondern in einem proprietären Format, was dann die Übertragbarkeit sehr stark einschränkt.

## Ablauf

Wie sieht nun der konkrete Ablauf bei der modellbasierten Testautomatisierung aus? Es ist zu unterscheiden zwischen dem Ablauf während der Designzeit und dem Ablauf zur Laufzeit (siehe **Abbildung 3**). In der Designzeit können der Geschäftsanalyst oder auch Testanalyst beliebige Testabläufe erstellen und pflegen (Sequenz 1 in **Abbildung 3**). Alle Änderungen werden im Modellierungswerkzeug gespeichert, haben aber noch keine Auswirkungen auf die bereits existierenden, automatisierten Testdurchläufe.

Zu einem definierten Zeitpunkt (beispielsweise „Anforderungen/ Testfälle sind abgenommen“ oder „zu testende Anwendung steht bereit“) können dann alle gespeicherten Testabläufe in das Automatisierungswerkzeug übertragen werden (Sequenz 2 in **Abbildung 3**). Dies kann direkt aus dem Modellierungswerkzeug heraus angestoßen werden, z. B. über einen zusätzlichen Menüpunkt, der den Generator aktiviert, oder indem der Generator direkt gestartet wird. Die Aufgabe des Generators ist es nun, die entsprechenden Skripte zu erzeugen, diese am korrekten Ablageort abzulegen und gegebenenfalls die Kompilierung der Skripte durch das Testautomatisierungswerkzeug anzustoßen. Als Ergebnis stehen somit automatisiert ausführbare Testdurchläufe zur Verfügung. Das Ausführen dieser Testdurchläufe geschieht zur Laufzeit. Jeder berechtigte Stakeholder kann über das Modellierungswerkzeug die Ausführung eines Testdurchlaufs starten (Sequenz 4 in **Abbildung 3**). Auch dies kann über entsprechende Erweiterungen in den Menüs des Werkzeugs realisiert werden.

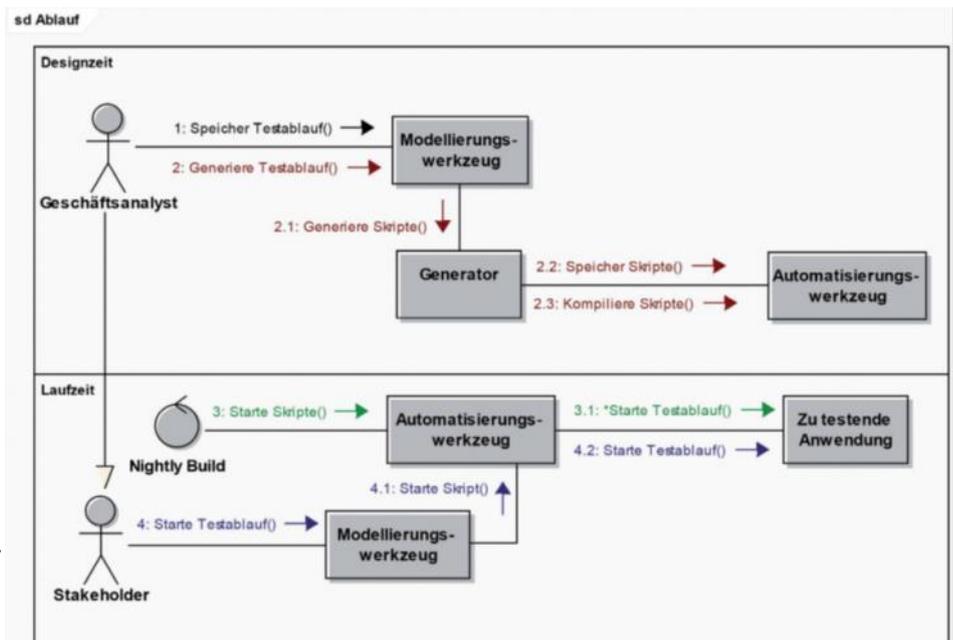


Abb. 3: Ablauf bei der modellbasierten Testautomatisierung

# Modellbasierte Testautomatisierung

Eine zeitpunktgesteuerte Ausführung der Testdurchläufe ist auch möglich (Sequenz 3 in **Abbildung 3**). Existiert zum Beispiel im Projekt ein nächtlich erstellter Versionsstand (*Nightly Build*), der im nächsten Schritt automatisch auf einem Testsystem bereitgestellt wird, kann im Anschluss an diese Bereitstellung die Ausführung aller automatisierten Testdurchläufe gestartet werden.

## Erfolgsfaktoren

Wie bereits angedeutet, benötigt die modellbasierte Testautomatisierung die modellierten Geschäftsobjekte und die definierten Oberflächen aus der Fachkonzeption. Erstellt der Fachbereich diese Informationen bereits mit einem Modellierungswerkzeug und ist dadurch mit dem Arbeiten eines Werkzeugs vertraut, so ist es ein Leichtes, den Fachbereich zu befähigen, auch die Testdurchläufe zu modellieren. Wenn dem nicht so ist, ist die Einarbeitung des Fachbereichs in ein Modellierungswerkzeug eine nicht zu unterschätzende Aufgabe. Insbesondere ist zu beachten, dass die Geschäftsobjekte und Oberflächen nicht annähernd so simpel sind, wie hier in den Abbildungen dargestellt. Der Aufbau von komplexen Geschäftsobjektmodellen, Oberflächendefinitionen und deren Verknüpfung über die Instanziierung von Geschäftsklassen erfordert ein grundlegendes Verständnis der UML-Notation und deren korrekte Anwendung in einem Modellierungswerkzeug.

Der Einsatz eines Modellierungswerkzeugs sollte somit nur dann angegangen werden, wenn die erstellten Modelle primär für die Fachkonzeption Verwendung finden und nicht ausschließlich für die Testautomatisierung benutzt werden. Zusätzlich profitiert allerdings auch der Entwicklungsbereich von diesen Modellen, z. B. durch die Möglichkeit der Generierung von Quellcode-Dateien oder die Möglichkeit der Verknüpfung von DV-Konstrukten mit entsprechenden Anforderungen. Eine modellbasierte Testautomatisierung ist also nur in Verbindung mit einer modellbasierten Anforderungsanalyse sinnvoll.

Ein weiterer kritischer Erfolgsfaktor ist die Implementierung des Generators, bei dem das Wissen aus mehreren Bereichen nötig ist: Das Modellierungswerkzeug und das Testautomatisierungswerkzeug müssen beherrscht und die Schnittstelle zum Modellierungswerkzeug muss verstanden werden (dies impliziert ein tiefgehendes Verständnis der UML).

Der Entwickler des Generators muss somit in der Welt der Anforderungen, der Implementierung sowie der Testfallerstellung und -durchführung zu Hause sein.

Aus dem bisher Gesagten wird deutlich, dass das Vorgehen einer modellbasierten Testautomatisierung keine Aufgabe ist, die ein einzelnes Projekt für sich allein angehen sollte. Eine strategische Verankerung des Vorgehens in einem Unternehmen ist sehr zu empfehlen.

## Fazit

Die modellbasierte Testautomatisierung ist ein weiterer notwendiger Baustein modellbasierter Teststrategien (zu weiteren modellbasierten Teststrategien vgl. [Roß10]). Mit entsprechenden Werkzeugen und einer durchdachten Konzeption können aus der Anforderungsanalyse mit wenig zusätzlichem Aufwand passende, automatisierte Testdurchläufe zur Verifizierung der Anforderungen erstellt werden. Das kann bei entsprechender Einweisung sogar direkt durch die Geschäftsanalysten geschehen. Die Auswirkungen von Anforderungsänderungen auf die Testdurchläufe werden im Werkzeug umgehend identifiziert, die Testdurchläufe können sofort angepasst und die Testskripte neu generiert werden. Damit sind auch die nicht zu unterschätzenden Kosten, die für die Konzeption und Realisierung eines Testskript-Generators aufzubringen sind, sinnvoll begründet und eingesetzt.

Des Weiteren unterstützt und erweitert die Methode der modellbasierten Testautomatisierung auch den Ansatz einer testgetriebenen Softwareentwicklung (vgl. [Bec02]). Dieser Ansatz – Schreibe erst die Tests (z.B. Unit-Tests), automatisiere diese Tests und entwickle dann solange, bis alle Tests erfolgreich ablaufen – beschränkt sich hauptsächlich auf Entwicklertests (Modul- und Integrationstest). Mit dem hier beschriebenen Ansatz kann dieses Vorgehen auf System- und Abnahmetests ausgeweitet werden: Modellierte erst die Testfälle (Fachbereich), generiere die automatisierten Testabläufe und entwickle dann die Anwendung, bis alle Testdurchläufe erfolgreich ablaufen.

◀ (Achim Krallmann, Markus Lingelbach)

Erstmals erschienen in  
OBJEKTSpektrum 02/2014  
<http://www.objektspektrum.de>

## Literatur

- [Bec02] K. Beck, Test Driven Development, Addison-Wesley 2002
- [Gro02] T. J. Grose, G.C. Doney, S.A. Brodsky, Mastering XMI, John Wiley & Sons 2002
- [Roß10] T. Roßner, C. Brandes, H. Götz, M. Winter, Basiswissen modellbasierter Test, dpunkt.verlag 2010
- [Rum10] J. Rumbaugh, I. Jacobson, G. Booch, The Unified Modeling Language Reference Manual, Addison Wesley 2010
- [Sei11] R. Seidl, M. Baumgartner, T. Bucsecs, Basiswissen Testautomatisierung, dpunkt.verlag 2011
- [Spi12] A. Spillner, T. Linz, Basiswissen Softwaretest, dpunkt.verlag 2012

## [Die Autoren]

Achim Krallmann ist als Test- und Projektmanager bei dem Unternehmen SQS tätig. Er ist graduiert systemischer Organisationsmanager und unterstützt Unternehmen bei der Einführung und Umsetzung von innovativen Testmethoden.



Markus Lingelbach ist als Testanalyst und Testautomatisierer bei der SQS beschäftigt und unterstützt Projekte im Testdesign und bei der Testdurchführung.



# Test Driven Development

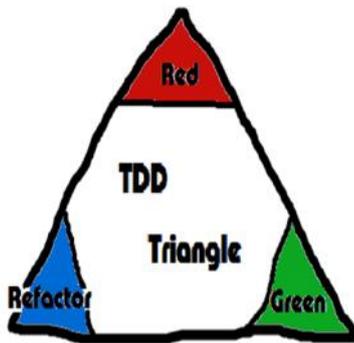
## [Test Driven Development]

von Klemens Loschy

### Agile rocks Business

Flexibel auf geänderte (Business-) Anforderungen reagieren können, die Forderung nach raschen time-to-market erfüllen können, trotzdem im Umgang mit Menschen respektvoll sein und Spaß an den Erfolgen des Teams haben können! So oder so ähnlich kann man agile Entwicklung zusammenfassen. Wichtig bei all dem ist die Identifikation des gesamten Teams mit der geteilten Aufgabe der Qualitätssicherung („collective quality ownership“). Eine der mittlerweile immer interessanter werdenden Standard Methoden die hierbei zum Einsatz kommen ist Test Driven Development, kurz TDD. Hierbei geht es um das entwicklungsnahe Testen bzw. das testnahe Entwickeln. Bei dieser testgetriebenen Entwicklung erstellt der Programmierer die Tests bevor er die jeweiligen dazu gehörige Business Logik erstellt.

### TDD in der Praxis - "Red, Green, Refactor"



Test Driven Development wird oft zu Unrecht als "unnötiger zusätzlicher Aufwand", sowohl bei Entwicklern als auch im Management, eingeschätzt. TDD zwingt den Entwickler vor der eigentlichen Umsetzung der Business Logik die zugehörigen Unit Tests zu schreiben - "Es darf keine Business Logik implementiert werden ohne zugehörigen, fehlschlagenden Unit Test". Zusätzlich ist eine fixe Verbesserungsphase, so genanntes "Refactoring" eingeplant, die für qualitativ hochwertige Ergebnisse sorgen soll. Diese drei Phasen der Entwicklung - Unit Test schreiben, Business Logik umsetzen, Code verbessern - werden iterativ bis zur fertigen Lösung fortgesetzt.

Das führt im ersten Blick zu mehr Aufwand, aber auch nur aus dem Grund, dass Unit Tests im Allgemeinen viel zu wenig Aufmerksamkeit in der Vergangenheit bekommen haben. Aber TDD geht noch viel weiter als einfach nur die Testfälle vor der Business Logik zu schreiben: TDD ist eine Methode, die den Fokus auf qualitativ hochwertige Ergebnisse legt! Naming, Duplication und ein gutes Design sind nur einige wichtige Punkte, die in TDD groß geschrieben werden.

### Die Vorteile von TDD

Ein Vorteil liegt auf der Hand: Durch den Einsatz von TDD erreicht man eine theoretische Code Coverage der Business Logik von 100%. Oben drauf bekommt man mit dem Fokus auf Naming und Design einen auf lange Sicht wartbaren Code. Die Entwickler können durch das dicht gewebte Sicherheitsnetz der Unit Tests ohne Gefahr große Teile der Business Logik bei Bedarf neu erstellen, denn: Die Anforderungen an die Business Logik sind in den Unit Tests persistiert und die korrekte Umsetzung kann laufend sichergestellt werden.

Ein weiterer und nicht unwesentlicher Vorteil wird durch TDD erreicht: die Entwickler sollen wieder zufrieden und stolz auf deren Machwerk blicken können. Software Craftmanship wird groß geschrieben, weg von Programmiermaschinen. Und zufriedene Arbeiter leisten doch zu meist auch gute Arbeit.

Interessiert TDD besser kennenzulernen? <http://www.seqis.com/TDD>



### [Der Autor]

Klemens Loschy ist SEQIS Managing Consultant. Zahlreiche Lasttest- und Testautomationsprojekte prägten seine berufliche Laufbahn. Seit 2002 ist er als Consultant, Testmanager und Trainer bei SEQIS tätig.



### Wussten Sie schon...?

..., dass die Methode Test Driven Development bereits vor ~15 Jahren entstanden ist?

Aufgrund der aktuellen Prominenz von TDD liegt die Vermutung nahe, dass es sich um eine neue Methode handelt. Tatsächlich wurden Unit Tests allgemein in den letzten Jahren zu sehr missachtet, wie der ice cream cone als Antipattern verdeutlicht. Die Testabdeckung wurde alleine durch Oberflächentests erreicht. Aktuell findet ein Umdenken statt: Unit Tests sind ein wichtiger Bestandteil der Qualitätssicherung und mit TDD gibt es die passende Methode dafür!

Mehr zu TDD:  
<http://blog.seqis.com>



### Aufruf zur Teilnahme an Studie zum Thema „Der Einfluss von Testautomatisierung auf Software Qualität“

Die Automatisierung von Testfällen ist ein wichtiges Thema in vielen Software Projekten. Mit dieser Studie möchte ich den Nutzen und die Bedeutung von Testautomatisierung für Software Projekte untersuchen.

Sie finden die Umfrage zu meiner Master Thesis unter folgendem Link:

<https://survey.technikum-wien.at/index.php?sid=84342&newtest=Y&lang=de>

Der Fragenbogen besteht aus 11 Fragen und die geschätzte Dauer beträgt ca. 10 Minuten. Selbstverständlich ist diese Umfrage komplett anonym.

Ich bedanke mich herzlich für Ihre Teilnahme!

Roman Gablek  
([roman.gablek@gmx.at](mailto:roman.gablek@gmx.at) / <mailto:roman.gablek@gmx.at>) / FH Technikum Wien



## [ASQF-Fachgruppen-Abende]



Die regionale Fachgruppe Software-Test Österreich des ASQF veranstaltet regelmäßig Fachgruppenabende zu ausgewählten Themen. Die Teilnahme ist kostenlos, Anmeldung unter angegebenem Link. Im Anschluss an jeden Abend gibt es einen Imbiss und Gelegenheit zur Diskussion und zu Networking.

Nächster Fachgruppenabend: Di., 10. Juni 2014, 18:00 - 20:00

### Software-Qualitätssicherung: von den Anfängen bis heute und darüber hinaus

Bei dieser Fachgruppe werden zwei Referenten vortragen: Prof. Dr. Karin Vosseberg und Prof. Dr. Andreas Spillner

Veranstaltungsort: FH Technikum Wien, Raum HS\_A1.04A, Höchstädtplatz 5, 1200 Wien

#### Prof. Dr. Karin Vosseberg

arbeitet seit 1987 als Informatikerin schwerpunktmäßig in den Bereichen Betriebssysteme und Softwaretechnik. Sie hat 1996 an der Universität Bremen zum Thema »Sicherheit in Informationssystemen« promoviert und war im Projekt »Informatica Feminale« in Curricula-Entwicklungen der Informatik aktiv mit dem besonderen Blickwinkel ein Informatikstudium für Frauen und Männer gleichermaßen attraktiv zu gestalten. Anschließend war sie acht Jahre in der Unternehmensentwicklung der Commerz Systems GmbH mit dem Schwerpunkt Qualitätsmanagement beschäftigt. Seit 2009 arbeitet sie als Professorin an der Hochschule Bremerhaven im Bereich IT Systemintegration mit dem Fokus auf Qualität in der Softwareentwicklung.



#### Prof. Dr. Andreas Spillner

arbeitet seit 1979 praktisch und in der Forschung im Bereich der Softwareentwicklung und -prüfung. Er studierte Informatik an der TU Berlin und promovierte an der Universität Bremen und ist seit 1993 an der Hochschule Bremen in der Fakultät Elektrotechnik und Informatik als Hochschullehrer tätig. Sein Lehrgebiet ist Softwaretechnik mit den Schwerpunkten Qualitätssicherung und Programmierung. Die Forschungsthemen sind Softwaretechnik, Validation von Software, Testmethoden sowie Vorgehensmodelle bei der Softwareentwicklung und die geschichtliche Entwicklung des Softwaretests. Andreas Spillner ist Gründungsmitglied und Ehrenmitglied des German Testing Board e.V. und war Gründer und Sprecher der Fachgruppe Test, Analyse und Verifikation von Software der Gesellschaft für Informatik (1990-2003) sowie Mitglied im Leitungsgremium der GI Fachgruppe Softwaretechnik (1997-2003). Im Jahr 2007 wurde er zum »Fellow der Gesellschaft für Informatik« ernannt.



#### Abstract:

Die ersten Veröffentlichungen zu Vorgehensmodellen für die Softwareentwicklung gehen auf die Anfänge der elektronischen Datenverarbeitung zurück. In diesen frühen Veröffentlichungen wird die Bedeutung des Prüfens und Testens bereits deutlich und entsprechend hervorgehoben. Die grundlegenden Arbeiten zum systematischen Testen wurden in den 70er Jahren publiziert und haben ihre Gültigkeit bis heute nicht verloren.

In der heutigen Praxis scheinen viele der Vorgehensweisen zur Qualitätssicherung nicht bekannt oder zumindest nicht praktiziert zu werden, aktuelle Untersuchungen belegen dies und werden im Vortrag vorgestellt.

Ein Blick in die Zukunft soll dazu beitragen, auch für die kommenden Aufgaben und Anforderungen gerüstet zu sein. Im Vortrag soll deutlich werden, dass es sich lohnen kann, aus der Vergangenheit für die Gegenwart und die Zukunft zu lernen.

Weitere geplante Termine der ASQF Fachgruppenabende (Themen noch offen): Di., 30.09.2014 und Di., 25.11.2014

Reservieren Sie sich diese Termine schon einmal im Kalender!

◀ (Mohsen Ekssir)

## [ASQF-Fachgruppenleitung Software-Test Österreich]

Dr. Mohsen Ekssir-Monfared ist Bereichsleiter Software Test und Qualitätssicherung bei BDC EDV-Consulting.

Besonders Augenmerk gilt bei seiner Arbeit der Entwicklung und dem Einsatz der Test Services wie Testkonzeption, -metrik, -automatisierung, -management und Testprozess-Assessment. Die Zertifizierung zum ISTQB® Certified Tester, Full Advanced Level und QAMP erlangte er 2010. Außerdem ist er seit 2012 iTACS™ Certified ISO/IEC 15504 Provisional Assessor TestSPICE und Certified Agile Tester (CAT). Er leitet seit 2010 die ASQF-Fachgruppe Software-Test Österreich.



Kontakt: [mohsen.ekssir@bdc.at](mailto:mohsen.ekssir@bdc.at)

## Neues von ASQF e.V. und iSQI GmbH

### **FLEX© – The Flexible Exam erhält Prädikat „Best of 2014“ des „INNOVATIONSPREIS-IT“**

Nur wenige Monate nach der Markteinführung hat FLEX© – The Flexible Exam auch die Experten überzeugt. Eine 100-köpfige hochkarätig besetzte Jury zeichnete das Produkt des Potsdamer International Software Quality Institute (iSQI GmbH) mit dem Prädikat „INNOVATIONSPREIS-IT ‚Best of 2014‘“ in der Kategorie „E-Learning“ aus. Der INNOVATIONSPREIS-IT wird von der Initiative Mittelstand vergeben, um innovative IT-Lösungen mit einem hohen Nutzwert für den Mittelstand zu würdigen. Das selbstbestimmte, zeit- und ortsunabhängige Prüfungsverfahren überzeugte die Jurymitglieder mit seinem hohen Innovationsgehalt. Es basiert auf einem maßgeschneidertem Konzept, das sich den individuellen Bedürfnissen eines jeden Prüfungsteilnehmers anpasst und dabei hilft, Kosten zu sparen.

Mit der erneuten Auszeichnung erhält das Potsdamer Institut bereits zum zweiten Mal das begehrte Prädikat. „Die Bedürfnisse der Kunden sind Treiber von Innovationen und damit Basis für den Erfolg am Markt.

Die Auszeichnung von FLEX© – The Flexible Exam mit dem „Best of 2014“-Prädikat bestätigt die Qualität unserer Arbeit“, freut sich iSQI-Geschäftsführer Stephan Goericke über den Erfolg. Das International Software Quality Institute zählt seit Jahren zu den innovativen Vorreitern im Bereich Personalzertifizierungen für Groß- und mittelständische Unternehmen. Als Deutschlands Marktführer ist das Potsdamer Institut vielgefragter Ausbildungspartner sowohl für Global Player als auch mittelständische Firmen in über 90 Ländern auf 6 Kontinenten in 10 Sprachen. Drei Viertel der DAX-30-Unternehmen vertrauen auf die Expertise des iSQI. Das Institut setzt sich für die stetige Weiterentwicklung und Verbesserung der Personal-Industriestandards ein, indem es als „Think Tank“ aktuelle Forschungsergebnisse mit Best-Practice-Erfahrungen verbindet. Hierfür pflegt das iSQI rund 100 internationale Partnerschaften und Kooperationen und arbeitet aktiv in internationalen Organisationen wie der ISO (International Organization for Standardization) mit.

**Hier erfahren Sie mehr über FLEX© – The Flexible Exam.**

◀ (Felix Winter)

### **[Öffentliche ISTQB® Prüfungstermine]**

Termine und nähere Informationen zu den öffentlichen Prüfungen in Österreich, sowie die Anmeldung finden Sie auf der [Homepage der iSQI](#)



### **ASQF e.V. gründet neue Fachgruppe für mobile Quality**

Der Arbeitskreis Software-Qualität und -Fortbildung e.V. (ASQF) ist seit März mit seiner neuen Fachgruppe „Mobile Devices & Apps“ in Österreich präsent. Die Entwicklung und der Test von Devices und Apps ist eine eigene anspruchsvolle Disziplin, die Experten mit Pioniergeist und Ideenreichtum erfordert. Die Fachgruppe hat sich hierbei die mobile Qualitätssicherung zum Ziel gesetzt. Für Interessierte bietet sie auf der Plattform „MeetUp“ unter dem Namen „Vienna Mobile Quality Crew“ ein Forum zum Austausch rund um das Thema Qualitätssicherung im mobilen Umfeld an. Infos und Termine unter <http://www.meetup.com/Vienna-Mobile-Quality-Crew-Meetup/>

◀ (Felix Winter)

### **ASQF e.V. Fachgruppe Mobile Devices & Apps / Vienna Mobile Quality Crew**

Datum: 15.05.2014  
Uhrzeit: ab 19:00 Uhr  
Ort: OBJENTIS Software Integration GmbH, Mariahilfer Straße 3, Wien  
Thema: **Mobile Quality Hangout: Best Practices in Mobile App Testing**  
Referent: Daniel Knott (via Google Hangout)  
Anmeldung unter: <http://www.meetup.com/Vienna-Mobile-Quality-Crew-Meetup/events/166187192/>

Datum: 24.06.2014  
Uhrzeit: ab 18:00 Uhr  
Ort: noch offen  
Thema: **Android als Zielplattform für Cybercrime**  
Referent: FH-Prof. DI Dr. Thomas Sommer  
Anmeldung unter: <http://www.meetup.com/Vienna-Mobile-Quality-Crew-Meetup/events/174010632/>

◀ (Felix Winter)

### **[Der Autor]**

Felix Winter hat Wirtschaftswissenschaften und Politische Wissenschaft studiert. Seit 2011 ist er Geschäftsführer des ASQF e.V. . Der ASQF e.V. ist mit über 1.200 Mitgliedern das größte Kompetenznetzwerk für Software-Qualität im deutschsprachigen Raum.



### **[Umfrage]**

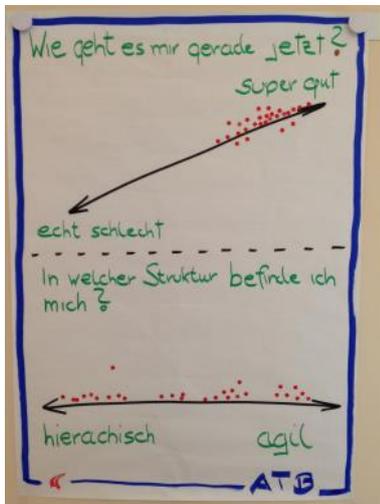
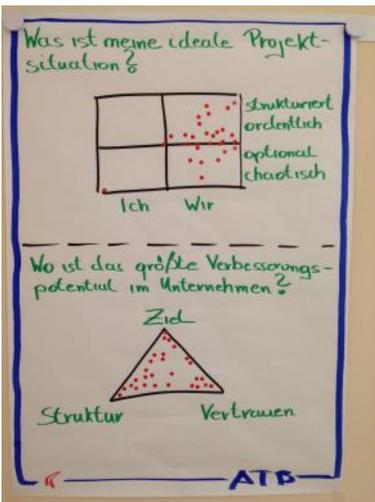
Auf Ihre Meinung kommt es an! Noch bis 04. Mai können Sie sich unter <https://de.surveymonkey.com/s/PG5BD8G> an der aktuellen ASQF-Umfrage beteiligen. Nehmen Sie sich **10 Minuten** Zeit und helfen Sie mit, die Etablierung von Qualitätsstandards in der Softwareentwicklung zu untersuchen.

## [ATB Expertentreff – mal *soft*, mal *destruktiv*]

von Alexander Weichselberger

... dass beim ATB Expertentreff die Community mit umfassendem Softwaretest-relevantem Knowhow „unterhalten“ wird, zeigen die Schwerpunkte des vergangenen und des kommenden ATB Expertentreffs:

**Mal soft...** Der erste ATB Expertentreff in diesem Jahr stand ganz im Zeichen der „Soft Skills“: Wie wir wissen, scheitern rd. 2/3 der IT Projekte immer noch – zumindest eine Dimension (Leistung, Einsatzmittel oder Zeit) aus dem Projektmanagement-Dreieck wird nicht erreicht. Hier sind sich die Analysten von Gartner und Co einig: Durch Verbesserung der „Soft Skills“ in den Projekten kann man mehr rausholen. Was sind nun diese „Soft Skills“? Nach der üblichen Einleitung und der Fragenrunde beim Expertentreff



gab es eine kleine Abstimmung, was „Soft Skills“ sein können:

Ausdrucksvermögen, Teamfähigkeit, Initiative, Flexibilität, Kommunikationsfähigkeit, Einfühlungs- und Durchsetzungsvermögen, Konfliktfähigkeit, Koordinationsfähigkeit, Kooperationsfähigkeit, Kontaktfreudigkeit,... und, und, und. Kurz zusammengefasst geht bei Soft Skills darum, sich selbst oder andere zu motivieren und im Team Ziele zu erreichen.

Nach dieser kurzen Einleitung startete **Günter Strobl**, geschäftsführender Gesellschafter von „DOCK12 – Offene Gesellschaft für menschengerechte Innovation“, mit seinem Vortrag „Soziokratie – Eine kompakte Einführung in das innovativste Organisationsmodell unserer Zeit“. Grundtenor des Vortrags: „Klassische Macht- und Entscheidungssysteme produzieren stets Verlierer. Die Soziokratie nutzt das Informationspotenzial aller Beteiligten – wobei Kritiker wichtige Impulsgeber sind – und lenkt deren Wissen in konstruktive Lösungen um.“

Nach dem Vortrag ging es wieder in die Diskussionsrunde, hier wurde die Runde mit Vera Stepan (DOCK12) und Karin Schmitzer (Leiterin Human Resources, ANECON) erweitert.

**Vera Stepan's** Eingangsstatement unterstrich die Bedeutung von Kommunikation und Austausch auf gleicher Augenhöhe: „Bespprechungen brauchen einen klaren Rahmen. Wenn alle Beteiligten aktiv teilhaben und es - ganz deutlich - Platz für unterschiedliche Meinungen gibt kann echte Kooperation entstehen.“ Frau Stepan ist, neben ihrem sonstigen IT Background, auch zertifizierte ISTQB Testerin ☺

Aber was macht eine Vertreterin aus dem HR Bereich in dieser Runde? Nun, ganz einfach: Bei diesem Thema ging es nicht nur um das WAS und WIE, sondern auch WER Teil des Systems ist. Und wer wäre da nicht profunder als der Personal-Gatekeeper eines Unternehmens? **Karin Schmitzer** machte mit ihrem Eingangsstatement („Soft Skills sind das essentielle i-Tüpfelchen für erfolgreiche Projekte.“) klar wie wichtig „Soft Skills“ im Projekt- und Unternehmensalltag sind und verriet auch, dass bei der Auswahl neuer Mitarbeiter „im Zweifelsfall fachliche Skills zu lernen sind; wichtiger ist es einfach, das der/die Bewerber/in grundsätzlich zum Team passt!“

Ein gelungener Auftakt ins heurige Jahr!

**... und dann wieder destruktiv:** Gleich nach Ostern, am 7.5., steht das TechGate Vienna wieder ganz im Zeichen des Softwaretests: Hans Hartmann, OBJENTIS, führt Ihre Software an die Grenzen: „Destruktives Testing“ konstruktiv und zielorientiert.

Alle Unterlagen zum letzten Expertentreff finden Sie unter [http://www.austriantestingboard.at/expertentreff\\_rueckblicke](http://www.austriantestingboard.at/expertentreff_rueckblicke) .

Ihr  
Alexander Weichselberger

Und hier noch ein paar Fotos vom Treff:



Merken Sie sich auch gleich die weiteren Termine für die ATB Expertentreffs im Jahr 2014 vor:

- **Mi., 02.07.2014**
- **Mi., 10.09.2014**
- **Mi., 05.11.2014**

## ATB-Award

Seit dem Jahr 2011 besteht eine gute geschäftliche Verbindung zur Firma SEQIS, welche sich nicht nur als hervorragende Profis rund um das Testen von Software einen Namen gemacht, sondern, und das freut uns sehr, das „kreative Potential“ der KlientInnen und MitarbeiterInnen unserer Werkstätte entdeckt haben.

Die Lebenshilfe NÖ vertritt die Rechte und Bedürfnisse von Menschen mit intellektuellen Beeinträchtigungen und setzt sich dafür ein, die Behinderungen aus dem Weg zu räumen, die diese Menschen oftmals „behindern“ können – wenngleich die außerordentliche menschliche Originalität mancher unserer Persönlichkeiten niemals wegzuleugnen ist. Dennoch ist es eine Frage des Umganges miteinander und der persönlichen Achtung und Wertschätzung, die ein gleichgestelltes Zusammenleben und Ergänzen in der Gesellschaft ermöglichen.

Laut dem Leitbild der Lebenshilfe NÖ haben...

**„Menschen mit intellektueller Beeinträchtigung das Recht, ein Leben entsprechend ihren Wünschen und Zielen zu führen. Wir respektieren dabei die Selbstständigkeit, Eigenständigkeit und Einzigartigkeit der Person und unterstützen sie bei der dazu notwendigen Entscheidungsfindung. Unser Blick liegt dabei in erster Linie auf der Förderung von Fähigkeiten und Stärken der einzelnen Person.“**

Aus diesem Auftrag heraus versucht die Werkstätte Mödling den Alltag zu gestalten und ist dankbar, sich in der Welt der Wirtschaft immer wieder eine kreative und individuelle Nische zu finden. Unsere Stärke liegt tatsächlich darin, auf die manchmal ausgefallenen Wünsche unserer Kunden eingehen zu können.

Dadurch konnten wir in den letzten Jahren einige Werkstücke und Awards genau auf den jeweiligen Anlass hin gestalten, wie in diesem Fall der ATB-Award zeigt. Ursprünglich hatte unsere Metallgruppe das Logo als Skulptur umgesetzt. Da aber nun eine höhere Stückzahl gefordert ist, haben wir befunden, dass dies mit dem Werkstoff Holz besser möglich ist. Hier werden die Einzelteile von einem Klienten maschinell ausgesägt und von weiteren KlientInnen mit Schleifpapier bearbeitet, zusammengeschraubt, verleimt und die Oberfläche konserviert.

Bei all diesen Arbeitsschritten geht es den Begleitern darum, einen möglichst hohen Arbeitsanteil durch die KlientInnen zu erzielen, diese in den Prozess miteinzubeziehen.

Was an der Zusammenarbeit mit SEQIS besonders schön ist, ist die Freundschaft, die sich mit Herrn Reinhard Salomon entwickelt hat und der dadurch einen persönlichen Bezug zur Kundenschaft herstellt. (Was der Motivation durchaus gut tut – nicht nur wegen der regelmäßigen Schokoladenspenden, die natürlich auch sehr willkommen sind.)



Letztlich sind wir dankbar für diese Zusammenarbeit, im Wissen, selbst an den Aufgaben wachsen zu können und der Öffentlichkeit zu zeigen, dass mit Einrichtungen wie der unseren immer zu rechnen ist. Dass hier oft verborgene Talente schlummern und geniale Werkstücke entstehen, von deren Existenz es sich zu wissen lohnt – und die man auch erstehen kann...Die Werkstätte Mödling sei ein Geheimtipp – dies haben wir sehr selbstbewusst immer für uns in Anspruch genommen! Bitte überzeugen Sie Sich selbst!

Bernd Hatter, 31.3.2014

### Kontakt:

Lebenshilfe NÖ, Werkstätte Mödling  
Untere Bachgasse 7, 2340 Mödling  
Öffnungszeiten: Mo – Fr, 08.00 bis 16.00 Uhr  
02236 22 66 8 Fax: 02236 22 66 8-9  
E-Mail: [moedling-wrk@noe.lebenshilfe.at](mailto:moedling-wrk@noe.lebenshilfe.at)

Fotos: Lebenshilfe Mödling



**lebenshilfe**  
**NIEDERÖSTERREICH**  
gemeinnützige GmbH



Wie berichtet, hat das ATB im Rahmen des Charity Punsch eine Spende von 4.600.- an die Stiftung Kindertraum übergeben. Dazu erreichte uns das unten abgebildete Dankschreiben. Somit konnte das ATB wieder einigen Kindern Herzenswünsche erfüllen.

Das ATB dankt allen Spendern, insbesondere den nebenstehend angeführten Firmen.

◀ (Helmut Pichler, Karl Kemminger)





## Dankeschön!



S. g. Herrn  
Herbert Pichler  
ATB Austiran Testing Board  
1090 Wien

★

Sehr geehrter Herr Pichler,

für Ihr Engagement im Rahmen des 3. ATB Charity Punsch im Dezember 2013 möchten wir uns sehr herzlich bedanken – Mit Ihrer Unterstützung haben Sie wieder zur Erfüllung von Herzenswünschen beigetragen!

**Muhammed, 9 Jahre, erhält Förderung durch eine Sprachpsychotherapie mit speziellem Hörtraining** - die Therapie wirkt entwicklungs-fördernd und bietet eine gute Grundlage, damit Muhammed besser am sozialen Leben teilnehmen kann:

Die Therapie hilft, Kommunikation, Konzentration und Wahrnehmung zu fördern. Beim Hörtraining wird mittels eines elektronischen Gerätes speziell aufbereitete Musik über einen eigens konstruierten Kopfhörer, mit einer zusätzlichen Schallquelle für die Knochenleitung, dem Gehör zugeführt. Dabei wird ein Prozess angeregt, der die neuronale Vernetzung positiv beeinflusst. Dadurch werden Hörwahrnehmungsstörungen beeinflusst, verändert und gemildert.

Bei Muhammed wurde frühkindlicher Autismus diagnostiziert, er verständigt sich ausschließlich durch Blickkontakt.




Muhammeds Schwester freut sich mit.



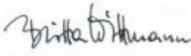
**Die Schülerinnen und Schüler eines sonderpädagogischen Zentrums in Niederösterreich freuen sich über einen Computer mit Spiel- und Lernsoftware.** Für das Erlernen von Farben, Formen, Buchstaben und Zahlen bietet gerade das Lernen in computerunterstützter Form einen großen Anreiz! Spezielle Lern- und Förderspiele fördern die kommunikativen Fertigkeiten, vermitteln Freude an Geschichten und wecken das Interesse für Buchstaben und Zahlen.

Der Computer motiviert die 8 bis 15-Jährigen (Entwicklungsverzögerungen, geistige Einschränkungen, Autismus): Sie wünschen sich nun „mehr Aufgaben“ auf dem Computer!



Mit herzlichen Grüßen





Britta Wittmann  
Stiftung Kindertraum



Wien, März 2014



Kurs	Termin	Ort	Anbieter
<b>ISTQB Certified Tester Foundation Level</b>	12.05. - 15.05.2014	<a href="#">Salzburg</a>	<b>BDC</b>
	12.05.-15.05.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	<b>Software Quality Lab</b>
	02.06. - 05.06.2014	<a href="#">Wien</a>	<b>ANECON</b>
	02.06. - 05.06.2014	<a href="#">Wien</a>	<b>BDC</b>
	02.06. - 05.06.2014	<a href="#">Wien</a>	<b>SQS</b>
	10.06.-12.06.2014	<a href="#">Mödling</a>	<b>SEQIS *)</b>
	23.06. - 26.06.2014	<a href="#">Graz</a>	<b>BDC</b>
	23.06.-26.06.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	<b>Software Quality Lab</b>
	21.07.-24.07.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	
	25.08.-28.08.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	
	25.08. - 28.08.2014	<a href="#">Wien</a>	<b>ANECON</b>
	01.09. - 04.09.2014	<a href="#">Wien</a>	<b>SQS</b>
	08.09.—11.09.2014	<a href="#">Wien</a>	<b>BDC</b>
	08.09.-11.09.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	<b>Software Quality Lab</b>
	22.09.—25.09.2014	<a href="#">Salzburg</a>	<b>BDC</b>
	06.10. - 09.10.2014	<a href="#">Wien</a>	<b>SQS</b>
	14.10.-16.10.2014	<a href="#">Mödling</a>	<b>SEQIS *)</b>
	27.10.—30.10.2014	<a href="#">Wien</a>	<b>BDC</b>
	03.11.—06.11.2014	<a href="#">St. Pölten</a>	
	03.11.—06.11.2014	<a href="#">Wien</a>	<b>ANECON</b>
<b>ISTQB Certified Tester Foundation Level englischsprachig</b>	19.05. - 22.05.2014	<a href="#">Wien</a>	<b>ANECON</b>
	02.06.–05.06.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	<b>Software Quality Lab</b>
	29.09.-02.10.2014	<a href="#">Wien, Linz, Graz</a>	
	30.09.-03.10.2014	<a href="#">Lustenau</a>	
<b>ISTQB Certified Tester Advanced Level Test Manager</b>	05.05.-09.05.2014	<a href="#">Mödling</a>	<b>SEQIS *)</b>
	11.06. –17.06.2014	<a href="#">Wien</a>	<b>ANECON</b>
	07.07.–11.07.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	<b>Software Quality Lab</b>
	15.09. - 19.09.2014	<a href="#">Wien</a>	<b>SQS</b>
	17.09. –23.09.2014	<a href="#">Wien</a>	<b>ANECON</b>
	22.09.-26.09.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	<b>Software Quality Lab</b>
<b>ISTQB Certified Tester Advanced Level Test Analyst</b>	23.06. - 26.06.2014	<a href="#">Wien</a>	<b>ANECON</b>
	30.06–03.07.2014	<a href="#">Wien, Linz, Graz</a>	<b>Software Quality Lab</b>
	01.07.-04.07.2014	<a href="#">Lustenau</a>	
	25.08.-28.08.2014	<a href="#">Mödling</a>	<b>SEQIS *)</b>
	06.10.-09.10.2014	<a href="#">Wien, Linz, Graz</a>	<b>Software Quality Lab</b>
	07.10.-10.10.2014	<a href="#">Lustenau</a>	
<b>ISTQB Certified Tester Advanced Level Technical Test Analyst</b>	06.05. - 08.05.2014	<a href="#">Wien</a>	<b>ANECON</b>
	23.06.–25.06.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	<b>Software Quality Lab</b>
	22.07.-24.07.2014	<a href="#">Mödling</a>	<b>SEQIS *)</b>
	28.07. - 30.07.2014	<a href="#">Wien</a>	<b>ANECON</b>
	14.10.-16.10.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	<b>Software Quality Lab</b>

\*) in Kooperation mit Diaz-Hilterscheid

Kurs	Termin	Ort	Anbieter
<b>Certified Agile Tester® Training</b> (Prüfung in deutsch oder englisch)	05.05. - 09.05.2014	<a href="#">Wien</a>	<b>BDC **)</b>
	12.05.-16.05.2014	<a href="#">Mödling</a>	<b>SEQIS</b>
	30.06. - 04.07.2014	<a href="#">Wien</a>	<b>ANECON</b>
	30.06.–03.07.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	<b>Software Quality Lab</b>
	14.07.-18.07.2014	<a href="#">Mödling</a>	<b>SEQIS</b>
	29.09.—03.10.2014	<a href="#">Wien</a>	<b>SQS</b>
	06.10. - 10.10.2014	<a href="#">Wien</a>	<b>BDC **)</b>
	20.10.-24.10.2014	<a href="#">Wien</a>	<b>ANECON</b>
	20.10.-24.10.2014	<a href="#">Lustenau</a>	<b>Software Quality Lab</b>
	27.10.-31.10.2014	<a href="#">Wien, Linz, Graz</a>	
<b>Certified Agile Test Driven Development® Training</b>	03.06.-05.06.2014	<a href="#">Mödling</a>	<b>SEQIS</b>
<b>IREB Certified Professional for Requirements Engineering Foundation Level</b>	20.05.–22.05.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	<b>Software Quality Lab</b>
	23.06.—25.06.2014	<a href="#">Wien</a>	<b>SQS</b>
	23.06. - 25.06.2014	<a href="#">Wien</a>	<b>ANECON</b>
	15.07.–17.07.2014	<a href="#">Wien, Linz, Graz, Lustenau</a>	<b>Software Quality Lab</b>
	02.09.-04.09.2014	<a href="#">Mödling</a>	<b>SEQIS</b>
	24.09. - 26.09.2014	<a href="#">Wien</a>	<b>ANECON</b>

\*\*) in Kooperation mit Agile Experts

## [Impressum]

Herausgeber: Austrian Testing Board Alser Straße 4/Hof 1/Eingang 1.5  
A-1090 Wien, Austria  
Telefon: +43 676 64 35 688  
Fax: +43 2256 65969  
Email: [office@austriantestingboard.at](mailto:office@austriantestingboard.at).

Dieses Magazin geht an alle zertifizierten Tester in Österreich, die ihre Zertifizierung dem ATB bekannt gemacht haben. Anregungen, Feedback, Kritik und ähnliches richten Sie bitte an [office@austriantestingboard.at](mailto:office@austriantestingboard.at)  
Wenn Sie dieses Magazin abbestellen wollen, senden Sie bitte eine Mail mit Betreff „Storno Magazin“ an [office@austriantestingboard.at](mailto:office@austriantestingboard.at).

Sämtliche in diesem Magazin zur Verfügung gestellten Informationen und Erklärungen geben die Meinung des jeweiligen Autors wieder und sind unverbindlich. Das ATB übernimmt keinerlei Haftung und Gewähr, insbesondere auch für die Richtigkeit oder Vollständigkeit der darin enthaltenen oder referenzierten Informationen oder deren Anwendung, sowie Druckfehler oder Irrtümer und es werden keinerlei Garantien, Zusicherungen oder sonstige Rechtsansprüche daraus begründet.

Die Redaktion behält sich Kürzungen vor. In keinem Fall spiegeln Leserbriefe die Meinung der Redaktion wieder.



Veranstaltung	Termin	Ort	Anbieter
<b>Tosca Certified User Foundation Level (TCUFL)</b>	03.06.-05.06.2014	<a href="#">Wien</a>	<b>Tricentis</b>
<b>Tosca Certified Quality Designer (TCQD)</b>	13.05.-15.05.2013	<a href="#">Wien</a>	<b>Tricentis</b>
	24.06.-26.06.2014	<a href="#">Wien</a>	
<b>Tosca Certified Administrator (TCA)</b>	17.06.-18.06.2014	<a href="#">Wien</a>	<b>Tricentis</b>
<b>Tosca Certified User Advanced Level (TCUAL)</b>	20.05.-22.05.2014	<a href="#">Wien</a>	<b>Tricentis</b>
<b>Agile Testing in a nutshell</b>	12.06.2014	<a href="#">Wien</a>	<b>ANECON</b>
	04.08.2014	<a href="#">Wien</a>	
<b>360 ° Testautomatisierung</b>	12.06.-13.06.2014	<a href="#">Wien</a>	<b>ANECON</b>
	04.08.-05.08.2014	<a href="#">Wien</a>	
<b>Funktionale Sicherheit</b>	04.-05.06.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	<b>Software Quality Lab</b>
<b>Scrum Master Professional</b>	25.06.-26.06.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	<b>Software Quality Lab</b>
<b>Testgetriebene Softwareentwicklung - Unit Testen in der Praxis</b>	06.05.-08.05.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	<b>Software Quality Lab</b>
	01.07.-03.07.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	
<b>Moderationstechniken im Requirements Engineering</b>	20.05.-21.05.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	<b>Software Quality Lab</b>
<b>Software Usability - FL</b>	20.05.-22.05.2014	<a href="#">Wien</a>	<b>Software Quality Lab</b>
<b>Professionelles Requirements Engineering und Management</b>	10.06.-12.06.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	<b>Software Quality Lab</b>
	23.-25.09.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	
<b>Aufwandsschätzung in Softwareprojekten</b>	10.06.-11.06.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	<b>Software Quality Lab</b>
	03.-04.09.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	
<b>ISAQB® Certified Professional for Software Architecture - FL</b>	23.06.-26.06.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	<b>Software Quality Lab</b>
	15.-18.09.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a>	
	16.-19.09.2014	<a href="#">Lustenau</a>	
<b>GUI-Testautomatisierung in Theorie und Praxis</b>	08.07.-09.07.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	<b>Software Quality Lab</b>
<b>Continuous Integration and Delivery</b>	11.09.2014	<a href="#">Wien</a> , <a href="#">Linz</a> , <a href="#">Graz</a> , <a href="#">Lustenau</a>	<b>Software Quality Lab</b>
<b>Risikomanagement in Softwareprojekten</b>	30.09.2014	<a href="#">Wien</a>	<b>Software Quality Lab</b>
<b>Automated Testing</b>	02.07.2014	<a href="#">Wien</a>	<b>BDC</b>
<b>Agile Testing</b>	03.07.2014	<a href="#">Wien</a>	<b>BDC</b>
<b>API Testing: Nur eine Schnittstelle und der passende Test</b>	26.06.2014	<a href="#">Wien</a>	<b>SEQIS</b>
<b>Test Driven Development – die Methode für Qualitätsbewusste</b>	18.09.2014	<a href="#">Wien</a>	<b>SEQIS</b>

## Konferenzen / Silberpartner

### [Konferenz Kalender]

Konferenz	Termin	Ort	Deadline Call for Paper
<a href="#">ignite</a>	20.-22.05.2014	Düsseldorf/ Deutschland	
<a href="#">National Software Testing Conference</a>	20.-21.05.2014	London/UK	
<a href="#">Mobile testing days</a>	04.-05.06.2014	Berlin/Deutschland	
<a href="#">6th World Congress for Software Quality</a>	01.-03.07.2014	London/UK	
<a href="#">German Testing Day</a>	02.07.2014	Berlin/Deutschland	
<a href="#">QA&amp;TEST</a>	22.-24.10.2014	Bilbao/Spanien	
<a href="#">Agile Testing Days</a>	10.11-13.11.2014	Potsdam/Deutschland	
<a href="#">Software Quality Days 2015</a>	20.-22.01.2015	Wien	31.05.2014

### Software Quality Days 2015

#### CALL FOR PAPERS

Gestalten Sie die Software Quality Days 2015 von 20. bis 22. Januar in Wien.

Begeistern Sie das Publikum mit Ihren Vorträgen und Workshops an der Seite von Rex Black und Gernot Starke als Keynote Speaker.

Zum Themenschwerpunkt "Software and Systems Quality in Distributed and Mobile Environments" wird ein gesamter Track gestaltet. Die weiteren Tracks sind zahlreichen anderen Themen zur Qualität, Effektivität und Effizienz in der System- und Softwareentwicklung gewidmet.

Reichen Sie jetzt ein! [www.software-quality-days.com/einreichen](http://www.software-quality-days.com/einreichen)



▶ Jetzt einreichen

CALL FOR PAPERS  
Nur bis 31. Mai 2014!

**TRICENTIS**  
Accelerate. Innovate. Automate.

Accelerate  
Business  
Innovation

Enterprise Software  
Test Automation

[www.tricentis.com](http://www.tricentis.com)

## [TestSPICE – Agiles Testen in SCRUM, Software Kanban & Co.]

Eine Einführung in die Welt von TestSPICE  
von Torsten Zimmermann

Das Ziel agiler Softwareentwicklung ist es, den Softwareentwicklungsprozess flexibler und schlanker zu machen, als es bei den klassischen Vorgehensmodellen der Fall ist. Man möchte sich mehr auf die zu erreichenden Ziele fokussieren und auf technische und soziale Probleme bei der Softwareentwicklung eingehen. In den letzten Jahren haben agile Entwicklungsmethoden in der Softwarebranche deutlich an Bedeutung gewonnen: VersionOne stellte beispielsweise in ihrer siebten jährlichen Umfrage zu agilen Methoden 2013 fest, dass bereits 84 % aller Unternehmen agile Prozesse einsetzen.

Doch wie sieht es neben der Entwicklung um die Softwaretests aus? Bislang finden sich in den gängigen agilen Entwicklungsmethoden nur ansatzweise Aussagen zu den Tests, oft in Form von Akzeptanztests am Ende einer Iteration. Um professionelle Softwaretests wie auch eine gute Harmonisierung zwischen Test und agiler Entwicklung zu erreichen, bedarf es jedoch eines Testprozessframeworks mit agilen Elementen. TestSPICE in der Version 3.0 erfüllt als einziges Framework beide Anforderungen und ist damit die erste Wahl, wenn es um die Etablierung agiler Testprozesse innerhalb von SCRUM, Software Kanban, Rapid Prototyping oder anderer agiler Methoden geht.

Der Begriff „Prozessframework“ im Zusammenhang mit TestSPICE ist jedoch nicht ganz richtig gewählt. TestSPICE ist in Wahrheit deutlich mehr als nur eine Ansammlung von Testprozessen. Vielmehr verbirgt sich dahinter ein Reifegradmodell oder Assessmentmodell, welches den Anforderungen von SPICE beziehungsweise der Norm ISO/IEC 15504 folgt. Hierbei existieren für die Prozessreife folgende Levels: siehe Abb. 1.

So erklärt sich auch der Name TestSPICE, welcher den Bezug zur SPICE Norm und den Fokus auf Testprozesse gleichermaßen herausstellen will.

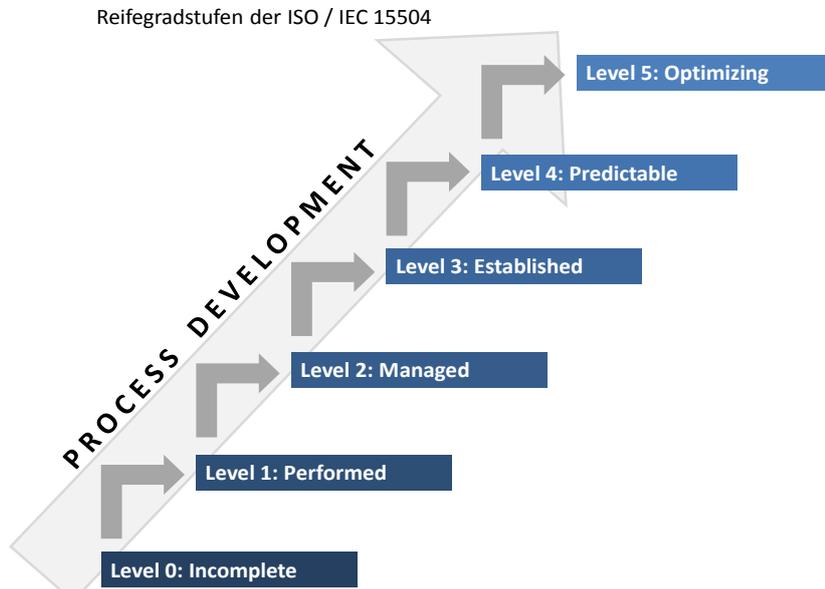


Abb. 1: Reifegradstufen der ISO/IEC 15504

Die Reifegradstufen werden bei ISO/IEC 15504 als Capability Levels bezeichnet und drücken den Grad der Fähigkeiten des betreffenden Prozesses aus.

### Entstehungsgeschichte von TestSPICE

Im Rahmen der Qualitätssicherung liegt stets das Hauptaugenmerk auf der Bewertung von Ergebnissen. Das heißt, es wird in Form von Tests entschieden, ob ein Ergebnis den zuvor gesetzten Anforderungen entspricht oder nicht. Dies führt dann bekanntlich zu Testergebnissen, welche „passed“ oder „failed“ sind. Diese Betrachtung hat den entscheidenden Nachteil, dass erst am Ende eines Entwicklungsschrittes auf Basis der besagten Testergebnisse Korrekturen eingeleitet werden. Effektiver wäre es jedoch, wenn man diesen Fehler erst gar nicht produzierte, sondern die Prozessqualität soweit verbesserte, dass Fehler nicht mehr entstehen können.

Ein zweiter Aspekt liegt in den methodischen Fehlern begründet, welche auch heute im Rahmen der Softwarequalitätssicherung häufig auftreten (siehe Infokasten zu methodischen Fehlern im Softwaretest).

Der Bedarf, Testprozesse angemessen bewerten und verbessern zu können, wurde bereits ab Mitte der 1990er Jahre erkannt. So starteten zu dieser Zeit europaweit unterschiedliche Initiativen, welche dieses Thema verfolgten. Mit zunehmender Verbreitung von SPICE kam der Bedarf nach einem zu ISO/IEC 15504 (SPICE) kompatiblen Testprozessmodell auf, welches zunächst unter dem Namen SPICE4TEST im Jahr 2010 auf mehreren Konferenzen vorgestellt wurde.

In der Folge schlossen sich führende Beratungsunternehmen (siehe Infokasten zu TestSPICE Mitgliedern) in Europa zur TestSPICE SIG (Special Interest Group) zusammen, um ein Assessmentmodell zu entwickeln, welches sich im Testumfeld mit der Frage nach der Prozessqualität beschäftigt. Hierbei ermöglicht TestSPICE die fokussierte Verbesserung aller testrelevanten Themen und Prozesse, welche in der gleichen Struktur wie bei den Entwicklungsprozessen vorliegen. Dies führte zur TestSPICE 1.0 (2010) und 2.0 (2011). Die Version 3.0, welche unter anderen agile Komponenten enthält, wird voraussichtlich Ende 2013 verfügbar sein. Damit ist TestSPICE die erste Norm, welche explizit auch agile Testprozesse als Pendant zu agilen Entwicklungsprozessen anbietet. >

## Allgemeiner Überblick und Anwendung

Die TestSPICE Prozesse decken das gesamte Testbusiness ab. In TestSPICE 3.0 sind folgende Prozesskategorien vorgesehen:

- **Business Life Cycle Processes:** In dieser Prozesskategorie sind alle Prozessgruppen zusammengefasst, welche für die Vorbereitung und Durchführung von Tests benötigt werden. Die betreffenden Prozesse sind so gestaltet, dass sie auch von Managed Services einsetzbar sind.
  - **Test Service Management** verwaltet alle Testaufträge. Diese Prozessgruppe fokussiert auf die Beauftragung und Akzeptanz von Testaufträgen.
  - **Testing** definiert die Prozesse zur Spezifizierung, Implementierung und Wartung von Tests.
  - **Test Process Management** fokussiert auf die Planung, die Steuerung und das Berichtswesen im Testumfeld.
  - **Regression and Reuse Test Engineering** beinhaltet Prozesse für die Wiederverwendung von Tests und das Verwalten von Regressionstests.
- **Technical Life Cycle Processes:** Hierin werden alle Prozesse zur Verwaltung und Pflege von Testumgebungen, Testdaten und automatisierten Testprozessen zusammengefasst.
  - **Test Environment Management** beinhaltet Prozesse für die Definition, die Planung, das Aufsetzen und den Support von Testumgebungen.
  - **Test Data Management** beschreibt die Prozesse für die Bereitstellung und Benutzung von Testdaten.
  - **Test Automation** beinhaltet Prozesse für die Definition, die Planung, das Aufsetzen und den Support von automatisierten Testprozessen.

Die TestSPICE Prozessdefinitionen sind nach ISO/IEC 15504-2 beschrieben und beinhalten für jeden Prozess die folgenden Informationen: siehe Abb.2.

Zentrales Element ist die Prozess-ID mit dem Prozessbezeichner zur eindeutigen Identifizierung des betreffenden Prozesses. Diese Prozess-ID steht in direkter Beziehung zu folgenden Informationselementen:

- **Prozessergebnisse** (Process Deliverables)
- **Prozesszweck**, das heißt: Was soll mit diesem Prozess erreicht werden? (Process Purpose)
- **Basis Aktivitäten**, das heißt die Beschreibung der wichtigsten Prozessschritte (Process Activities)
- **Ein- und Ausgangsschnittstellen** mit der entsprechenden, beschriebenen Ergebnischarakteristik (Input/Output)

Im Rahmen der Prozessentwicklung wurde das TestSPICE PRM (Process Reference Model) erstellt, welches – über seine verschiedenen Entwicklungsiterationen – die Erfahrungen mehrerer Tausend Experten aus dem Bereich Softwarequalität berücksichtigt. Das Referenzmodell, welches analog zur ISO/IEC 15504 strukturiert ist, berücksichtigt die Anforderungen und Best Practices aus eine ganzen Reihe bekannter Normen und Modelle im Qualitätsumfeld (siehe Infokasten).

TestSPICE kann in jeder Branche und Organisationsgröße angewendet werden. Das Referenzmodell ist so konzipiert, dass es im sequenzorientierten wie auch agilen Entwicklungsumfeld implementiert werden kann.

Da TestSPICE ein Assessmentmodell ist, enthält diese Norm neben dem bereits vorgestellten PRM auch das TestSPICE PAM (Process Assessment Model). Dieses Modell umfasst eine Reihe von Indikatoren für die Beurteilung der Prozessleistung und Prozessfähigkeit. Die besagten Indikatoren dienen als Grundlage zur Bewertung der aktuellen Prozessqualität beziehungsweise Prozessfähigkeit (Capability gemäß SPICE oder ISO/IEC 15504) im Testumfeld der betreffenden Organisation beziehungsweise des betreffenden Unternehmens.

Die TestSPICE PAM und PRM sind aufeinander abgestimmt, sodass eine gemeinsame Basis für die Durchführung des Prozessassessments auf einer harmonisierten Skala gegeben ist.

Das Prozess-Assessment-Modell definiert ein zweidimensionales Modell der Prozessfähigkeit. In einer Dimension werden die Prozesse definiert und in Verfahren eingeteilt. Innerhalb einer Prozesseskategorie werden Gruppenprozesse und auf einer zweiten Ebene nach der Art der Tätigkeit strukturiert.

Die andere Dimension beschreibt die Prozessfähigkeit (Capability), welche eine Reihe von Verfahrensattributen in Reifegradstufen gruppiert definiert. Die betreffenden Prozessattribute werden gemessen, und hieraus wird dann die besagte Prozessfähigkeit ermittelt.

>

Struktur einer Prozessbeschreibung

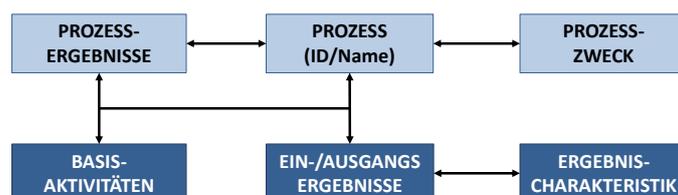


Abb. 2: Struktur einer Prozessbeschreibung

Das Ziel von TestSPICE ist die stetige Verbesserung der Prozessleistung und der Prozessfähigkeit, basierend auf einem standardisierten Assessmentansatz, welcher den ISO/IEC 15504 Anforderungen folgt. TestSPICE stellt sich also stets die Frage nach dem „was“: Das Assessment Modell fokussiert auf Prozesse und bewertet deren Fähigkeiten beziehungsweise den betreffenden Reifegrad.

## Grundzüge agiler Softwareentwicklung

Ich möchte hier keine umfassende Einführung über agile Softwareentwicklung schreiben. Hierzu gibt es bereits eine ganze Reihe umfassender Artikel aus meiner Feder wie auch der anderer Autoren, welche in diesem Magazin bereits erschienen sind. Als kleine Einstimmung nur so viel:

Agile Softwareentwicklung ist der Oberbegriff für den Einsatz von Agilität (lat. agilis: flink; beweglich) in der Softwareentwicklung. Je nach Kontext bezieht sich der Begriff auf Teilbereiche der Softwareentwicklung – wie im Fall von Agile Modeling – oder auf den gesamten Softwareentwicklungsprozess – exemplarisch sei Extreme Programming angeführt. Agile Softwareentwicklung versucht mit geringem bürokratischen Aufwand, wenigen Regeln und meist einem iterativen Vorgehen auszukommen. Ich spreche deshalb gerne von iterativen Prozessmodellen als Oberbegriff für die Entwicklungsmethoden wie SCRUM, Software Kanban etc. (siehe Infokasten) im Vergleich zu sequenzorientierten Prozessmodellen, wie zum Beispiel dem V-Modell.

Diese iterativen Prozessmodelle zielen auf einen – im Vergleich zu sequenzorientierten Prozessmodellen – flexibleren und schlankeren Softwareentwicklungsprozess ab. Auf diese Weise wird mehr auf die zu erreichenden Ziele fokussiert und auf technische wie auch soziale Probleme bei der Softwareentwicklung eingegangen.

## TestSPICE im agilen Umfeld

Bereits TestSPICE 2.0 bietet einige Ansätze, welche auch im agilen Umfeld eingesetzt werden können. Im Rahmen der Entwicklung zur Version 3.0 wurde jedoch ganz bewusst das Prozessmodell um agile Elemente erweitert, um so die einfache Anwendung in iterativen gleichermaßen wie in sequenzorientierten Prozesswelten zu ermöglichen. In diesem Zusammenhang wurde das Frameworkdesign im Vergleich zu TestSPICE 2.0 gestreamlined und an den novellierten ISO/IEC 15504 Anforderungen von 2012 ausgerichtet. Insbesondere besagtes Streamlining unterstützt den Lean-Management-Gedanken, welcher in allen agilen Methoden einen der zentralen Ansätze bildet. In diesem Rahmen ist es bei agilen Prozessen wichtig, das Maß der notwendigen Prozessdetaillierung exakt an die aktuellen Anforderungen im Entwicklungsumfeld anzupassen. Hier liefert das TestSPICE Process Assessment Model die entsprechenden Informationen darüber, ob der aktuell ermittelte Capability Level zu den geprüften Testprozessen den besagten Anforderungen genügt. Entsprechend der Ergebnisse aus dem Assessment können die Testprozesse an den richtigen Stellen weiterentwickelt beziehungsweise nachjustiert werden.

TestSPICE legt die entsprechenden Informationen (Input/Output) fest, welche bezüglich des Betriebs von Testprozessen notwendig sind. Jedoch wird die Art und Weise der Informationsdarstellung nicht reglementiert. Somit können auch die Regelwerke agiler Methodenansätze im Bereich der Dokumentation berücksichtigt werden.

### TestSPICE

Ein wichtiger Punkt bei der Entwicklung von Prozessmodellen ist die Berücksichtigung vieler Quasistandards, Normen und Modelle. Diese Eigenschaft unterstützt eine hohe Anwenderakzeptanz. TestSPICE berücksichtigt beispielsweise die folgenden Normen und Best Practices:

- IEEE 829
- ISO/IEC 15504
- ISO/IEC 29119
- ISTQB®

## Fazit

TestSPICE bietet sowohl ein Prozessreferenzmodell als auch ein Prozess-Assessment-Modell für Testprozesse an, welches die aktuellen Geschäftsanforderungen berücksichtigt. Diese Modelle sind branchenneutral entwickelt und können deshalb in jeder Branche gleichermaßen angewendet werden. Das zentrale Element ist natürlich das TestSPICE PAM. Hierüber lassen sich Prozesse über eine standardisierte Methode bewerten und weiterentwickeln. Die Prozessqualität wird in Capability Levels gemessen. Deren Kanon wie auch Wertmaß orientieren sich an SPICE beziehungsweise ISO/IEC 15504. Unternehmen, welche bereits die SPICE Norm einsetzen, können einfach die TestSPICE Norm in ihr Qualitätssystem integrieren, da Kompatibilitätsprobleme nicht zu befürchten sind. Weiterführende Informationen zu TestSPICE finden sich im Internet unter <http://testspice.de>.

◀ (Torsten Zimmermann)

### Was versteht man unter agilen Prozessen?

Allen agilen Prozessen ist gemeinsam, dass sie sich zahlreicher Methoden bedienen, um die Aufwandskurve möglichst flach zu halten. Inzwischen gibt es eine Vielzahl von agilen Prozessen. Zu den bekanntesten Vertretern dieser Prozessgattung zählen unter anderem:

- Adaptive Software Development (ASD)
- Crystal
- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Software Kanban
- Scrum
- Agiles Testen
- Behavior Driven Development (BDD)

Der Rational Unified Process (RUP) wird von vielen Vertretern agiler Methoden (viele von ihnen haben das Agile Manifest unterzeichnet) als nicht-agiler, schwergewichtiger Prozess aufgefasst. Das ist allerdings umstritten beziehungsweise es wurde versucht, mit dem Agile Unified Process eine agile Variante von RUP zu entwickeln.

## Die zehn größten methodischen Fehler im Softwaretest

Im Bereich des Softwaretests finden sich oft einer oder mehrere der nachfolgend aufgezählten Schwachpunkte in Bezug auf die Planung, Vorbereitung und Durchführung von Tests:

1. kein Testmanagement
2. Unterschätzung von Komplexität und Vorbereitungsdauer
3. fehlendes oder unzureichendes Testdatenmanagement
4. fehlende Testautomatisierung
5. Unterschätzung des Know-how-Aspekts
6. Verzicht auf methodisches Vorgehen
7. kein Tool-Einsatz oder Einsatz ungeeigneter Tools
8. falsche Testfälle beziehungsweise unzureichende Testüberdeckung
9. Last- und Performanztests erst am Projektende
10. Unterschätzung der Folgen, welche sich durch methodische Fehler einstellen

## Die Prozessreife gehört zu den erfolgskritischen Faktoren

Die Bedeutung der Prozessreife oder Prozessfähigkeiten im Bereich der Softwareentwicklung wird häufig unterschätzt. Es gibt jedoch gute Gründe, um die Prozessreife weiterzuentwickeln:

- Reduzierung des Risikos unerwünschter Ergebnisse
- Erhöhung der Prozesssicherheit
- Verbesserung der Prozesstransparenz
- Verantwortung kann tatsächlich delegiert werden
- ist die Basis für die Etablierung einer effektiven Prozesssteuerung
- ist die Basis für die Realisierung von Vorhersagemodellen.

TestSPICE unterstützt die nachhaltige, planvolle und gesteuerte Entwicklung der Prozessreife im Testumfeld.

## Prinzipieller Ablauf eines TestSPICE Assessments

Die TestSPICE Norm ist inzwischen von der INTACS hinsichtlich der ISO/IEC 15504 Anforderungen verifiziert. Unternehmen können sich somit nach TestSPICE zertifizieren lassen, um ihre besondere Kompetenz im Bereich Softwarequalität und Testmanagement am Markt herauszustellen. Das Prozessassessment ist wie bei ISO/IEC 15504 organisiert, und die Prozessbewertung entspricht der ISO/IEC 15504 Bewertung.

Das Assessment läuft in folgenden Stufen ab:

1. Preparation
  - Pre-Assessment/Briefing
  - Initialization
  - Assessment Team Identification
  - Collecting Supporting Material
  - Planning & Scheduling
  - Defining Confidentiality Agreement
2. Execution
  - Organizational Unit Briefing
  - SPU/Project Assessment
  - Data Collection
  - Data Validation
  - Process Attribute Rating
  - Interview Feedback Sessions
3. Reporting
  - Assessment Report Preparation
  - Final Report Preparation
  - On-Site Final Meeting/Presentation of the Assessment Results

## [Der Autor]

Bereits seit 1985 entwickelte Torsten Zimmermann Anwendungssoftware für Unternehmen und Behörden. Nach seinem vollendeten Studium als Diplom Wirtschaftsinformatiker (1993) kam er mit Qualitätsthemen innerhalb des Software-Lifecycles in Berührung. Ab dem Jahre 1995 berät er im Rahmen international angelegter Projekte in den Themen Software-Qualität und Qualitäts- / Test-Management. Im Laufe der Jahre wurde er zu einem der Experten in Europa. Im Rahmen seiner Arbeiten entwickelte er den risikobasierten Testansatz..

Weitere Ergebnisse und Erkenntnisse aus der Qualitäts-Management-Praxis führten zu dem T1 TFT (Test Framework Technologies).

Heute entwickelt Torsten Zimmermann neue Ansätze für leistungsfähigere Testkonzepte und -Frameworks. Als Referent auf Kongressen und Fachautor präsentiert er regelmäßig seine Erfahrungen, Ergebnisse und Konzepte in zahlreichen Vorträgen und Fachartikeln auf nationaler wie auch internationaler Ebene.

Kontakt: [http://www.xing.com/profile/Torsten\\_Zimmermann2](http://www.xing.com/profile/Torsten_Zimmermann2)



## [Redakteurinnen und Redakteure gesucht]

Haben Sie einen außergewöhnlichen Bug gefunden?

Kennen Sie ein Tool, von dem die Testercommunity unbedingt wissen sollte?

Haben Sie ein Buch gelesen, das andere auch lesen sollten?

Ja? Dann schreiben Sie an den ATB-Insider.

Wir suchen noch Redakteurinnen und Redakteure, die Spaß am Schreiben haben.

Einsendungen an:

[newsletter@austriantestingboard.at](mailto:newsletter@austriantestingboard.at)

◀ (Karl Kemminger)