

[Editorial]

Der Sommer war heuer mitunter sehr herausfordernd und brachte viele mächtig ins Schwitzen.

Um diesen Zustand in Projekten möglichst zu vermeiden, empfehle ich die tollen Beiträge, die wir wiederum erhalten haben. Neben einem guten Überblick zum Tooleinsatz im Agilen Umfeld, Requirements Engineering und dem QAMP Neu, freut es mich ganz besonders, dass es im universitären Umfeld ein starkes Lebenszeichen zum Thema Software-Qualität gibt.

Wir bekamen auch die Chance, einen Artikel aus einem internationalen Forum abzu- drucken: modellbasierte Ansätze für die Generierung von Testscripts gibt es zwar bereits seit Jahren, hier wird jedoch ein weiterer Ansatz sehr ausführlich beschrieben. Viel Spaß beim Lesen.

Im Herbst starten wir auch wieder mit den beliebten Netz- werktreffen – Details finden Sie auf unserer Website.

„Gut Test“
◀(Helmut Pichler)

[Der Autor]

Dipl.Ing.
Johannes
Hochrainer
ist Senior Test
Consultant bei
Software Quality Lab und
Leiter des Tool Evaluation
Centers. Unter anderem be-
treut er Kunden bei der Aus-
wahl von Testwerkzeugen und
begleitet bei der Umsetzung
von Test-
automatisierungsprojekten.



[Werkzeuge für agile Entwicklung]

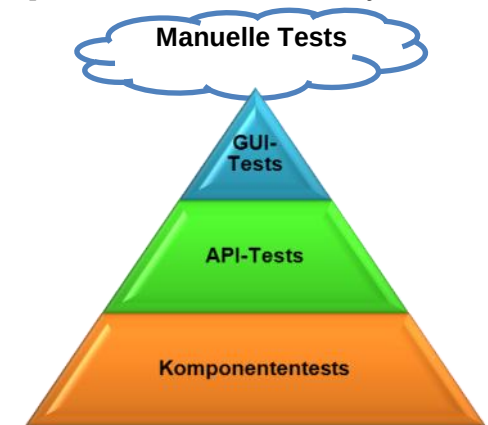
Von DI Johannes Hochrainer, Software Quality Lab

Agile Vorgehensmodelle, wie XP, Scrum oder ganz aktuell Kanban, haben andere Anforderungen an Testwerkzeuge als klas- sische Modelle. Kunden, Anforderungsma- nager, Entwickler und Tester müssen viel enger zusammenarbeiten, um auf Änderun- gen reagieren zu können und in kurzen In- tervallen Produktinkremente gute Qualität liefern zu können.

Anforderungen spezifizieren, Releases pla- nen, Aufgaben an Entwickler/Tester zuwei- sen, Testfälle aus Anforderungen ableiten, Tests durchführen, Fehlerberichte verwal- ten etc. – das alles und teilweise noch eini- ges mehr versprechen ALM-Werkzeuge (Application Lifecycle Management). IBM hat vor Jahren mit der Jazz-Plattform und den darauf aufbauenden neuen IBM Rational Werkzeugen einiges vorgelegt. Microsoft machte im Jahr 2010 aus der klassischen Entwicklungsumgebung Visual Studio einen großen Schritt in Richtung ALM. Mit Visual Studio Team System 2010 und Team Founda- tion Server 2010 als Backend bietet Micro- soft auch eine Vielzahl an Testautomatisie- rungswerkzeugen (Unit, Performanz, GUI, Datenbank, ...). Seit kurzem ist auch HP mit an Board: Mit HP ALM 11 gibt es Traceability von den Anforderungen bis zum Source Code. Auch Integrationen für die Entwick- lungsumgebungen Eclipse und Visual Stu- dio dürfen dabei nicht fehlen. Schneller als die Software-Riesen haben kleinere Anbie- ter passende Werkzeuge geliefert. Polarion, Atlassian oder CollabNet, um nur einige zu nennen.

Auch bei Testautomatisierungswerkzeugen hat sich einiges getan. Wie aus der Testpy- ramide erkennbar, verschiebt sich bei agiler Entwicklung der Fokus weg von GUI- Tests „runter“ zu API- und Komponentent- tests. Letztere versprechen nämlich schnel- lere Implementierung und weniger War- tungsaufwand. Unit-Testframeworks, Mo- cking, Code-Metriken, Behaviour-Driven Development (Cucumber), Testfall- und Testdatengeneratoren (Microsoft Pex). Bei diesen Worten zieht es die Mundwinkel ei- nes agilen Testers nach oben.

Aber auch GUI-Testautomatisierungs- werkzeuge passen sich an agile Software- entwicklung an. Proprietäre Sprachen sind kaum noch zu finden. Testfälle werden heu- zutage in den Sprachen spezifiziert, die auch Entwickler verwenden. Das erleichtert die Zusammenarbeit im Team. Micro Focus (früher Borland) verwarf beispielsweise die Sprache 4Test und setzt nun auf Java



und .Net.

Eine alte Binsenweisheit besagt, dass je frü- her ein Fehler entdeckt und behoben wird, desto kostengünstiger ist es. Ständige In- tegration (Continuous Integration), also das automatische Bauen des Codes, Testumge- bung aufsetzen, Tests durchführen, Metriken ermitteln, Entwickler informieren usw., setzt genau hier an. Continuous Integration Tools, wie etwa Jenkins, erfreuen sich dabei gro- ßer Beliebtheit. Zusätzlicher Nebeneffekt dabei: Entwickler werden mutiger und tra- uen sich auch komplizierte Codes umzuba- uen, denn sie wissen: Entsteht dabei ein Feh- ler, werden sie sehr rasch informiert und können sofort Nachbesserungen tätigen.

Wie man sieht, ist Testen in agilen Software- entwicklungen spannend und voll mit inte- ressanten Möglichkeiten. Die Wahl der rich- tigen Werkzeuge trägt erheblich zum Pro- jekterfolg bei. Also: Viel Spaß beim Evaluie- ren! ◀ (Johannes Hochrainer)

Erfolg braucht Training

SQS – Ihr erfolgreicher Weg zur Qualifikation



Über SQS Software Quality Systems

Die SQS-Gruppe ist der größte unabhängige Anbieter von Software-Test- und Qualitätsmanagement-Dienstleistungen. Position und Kompetenz der SQS als Marktführer sind auch im Trainings-Bereich das Ergebnis von fast 30 Jahren Beratungsaktivität. Die Stärke der SQS-Seminare liegt in der Expertise der Trainer.

SQS setzt neben ISTQB®-Themen nun auch vermehrt auf „Requirements Engineering“ und „Agile“

Neu bei SQS:

- Certified Professional for Requirements Engineering (CPRE) Advanced Level – Elicitation & Consolidation
- Certified Agile Tester® (CAT)

Mehr Informationen: www.sqs.com/de/at/training.php

Anmeldung über training-at@sqs-group.com oder +43 (0) 1/319 35 23-18. Wir freuen uns auf Sie.

SQS Training: individuell und professionell

Die nächsten Termine in Wien:

- CPRE Foundation Level:
26.–29.09. 2011
- CPRE Advanced Level – Elicitation & Consolidation:
07.–09.11.2011
- Certified Agile Tester® (CAT)
10.–14.10.2011
- Agile Software-Entwicklungsmethoden und -Tests:
31.10.2011

Alle weiteren Seminare und Termine finden Sie unter:
www.sqs.com/de/at/training.php

Bezahlte Einschaltung



Aufbauend auf dem profunden Mix aus langjähriger Erfahrung sowie der fundierten Ausbildung in den wesentlichen agilen Methoden, hat der österreichische Agil-Pionier ANECON die erfahrungs- und erlebnisorientierte Trainingsserie **Agile LIVE!** entwickelt.



Der erste Workshop der Trainingsserie fand Ende August statt und „Agile Testing In A Nutshell“ war ein voller Erfolg!

Testen im agilen Umfeld: EIN ERFAHRUNGSBERICHT

Agilität wurde bei dem **Agile LIVE!** Kurs „Agile Testing In A Nutshell“ von Beginn an gelebt und die acht Teilnehmer teilten sich nach dem „Stand Up Meeting“, bei welchem agile Vergangenheit und Kurs-Erwartungen eingebracht wurden, in zwei „SCRUM-Teams“ auf.

Aus dem priorisierten Workshop-Product-Backlog wurde zunächst von den Teams das Modul „Agile In A Minute“ gewählt: Kurz & prägnant wurde die agile Welt, vom Product-Backlog zum



abgenommenen Deliverable, kennengelernt und erarbeitet.

Dieses Basis-Know-how war das ideale Fundament für die darauffolgende „Story“, bei welcher die Teilnehmer mit LEGO® – in der Agile Community meist weit verbreitet, da agile Methoden einfach und effizient vermitteln werden können – spielten.

Die individuellen und sehr durchdachten „Deliverables“ zeugten durchwegs von Kreativität, die „Time-Box“ verging hierbei deutlich schneller als gewünscht.

Die weiteren „Stories“ behandelten die Rollen im Agilen Umfeld und deren Aufgaben - mit dem besonderen Augenmerk auf die erweiterten Skills. Im Rahmen der finalen „Story“ erfuhren die Teilnehmer, was konkret in einem Agile/Scrum-Projekt vor sich geht, lernten das Scrum-Vokabular zu deuten und bekamen einen Einblick, was sie nun als Tester konkret erwartet bzw. was von ihnen erwartet wird.



Conclusio der Teilnehmer: „WIR WOLLEN MEHR!“

Das Interesse für den folgenden 5-Tages Kurs „**Certified Agile Tester**“ (CAT) ist groß! Bei **ANECON**, Österreichs einziges Unternehmen **mit eigenen CAT-Trainern**, kann dies gestillt werden:

- **Wien: 19. - 23. September 2011**
- **München: 17. - 21. Oktober 2011**

Weitere Informationen erhalten Sie unter
www.anecon.com/angebot/trainings
oder trainings@anecon.com

Bezahlte Einschaltung



Software Quality Lab ist das führende herstellerunabhängige Unternehmen im Bereich Software-Prozesse und SW-Qualitätssicherung und Marktführer bei Tester-Seminaren in Österreich.

Software Quality Lab ist in Österreich vertreten an den Standorten Wien, Linz, Graz und Bregenz sowie in Deutschland in München.

Infos zu den Dienstleistungen und Seminar-Termine finden Sie unter www.software-quality-lab.at.

Software Quality Days 2012

**Der jährliche Kongress für Software Qualität & Testing
17.-19. Jänner 2012, Austria Trend Hotel Savoyen, Wien**

Abwechslungsreiche Keynotes, Fach-Vorträge, Fach-Ausstellungen, Tutorials und Workshops stehen auch heuer wieder auf dem Programm der Software Quality Days.

Die hohe Qualität der Vorträge und Tutorials sowie eine Mischung aus neuesten wissenschaftlichen Themen und hochwertigen Umsetzungs-Vorträgen aus der Praxis machen diese Veranstaltung zum Top-Event im Bereich Software-Qualität!

Sichern Sie sich Ihre Teilnahme und planen Sie jetzt schon die Software Quality Days 2012 ein - **bis 30.9.2011 zu ermäßigten Frühbucher-Konditionen.**

Einen kleinen Vorgeschmack auf die Konferenz erhalten Sie durch unseren [Rückblick auf die vergangene Konferenz](#).

Melden Sie sich jetzt an unter www.software-quality-days.at!

Treten Sie auch der XING-Gruppe [„Software Quality Days“](#) bei und profitieren Sie vom Know-How rundum das Thema Software Qualität und Testen!



Hier könnte Ihr Inserat stehen!

**Haben Sie ein interessantes Testtool?
Suchen Sie einen qualifizierten Tester für Ihr Team?**

Nutzen Sie die Möglichkeit, sich direkt an die Testcommunity zu wenden. Kontaktieren Sie das ATB, wenn Sie nähere Informationen zu den Konditionen für Inserate oder Partnerschaften haben wollen.

[Requirements Engineering]

von Dr. Thomas Harms

Warum Requirements Engineering?

Tester müssen oft nicht lange nach Motiven für Requirements Engineering (RE) suchen, kennen sie doch die Folgen von mangelhaftem RE aus eigener Erfahrung. So zeigen sich beispielsweise bei der Testfallermittlung oft unterschiedliche Vorstellungen der Projektbeteiligten über das korrekte Softwareverhalten. Eine typische Fehlerquelle im Integrationstest ist, dass Anforderungsänderungen nicht systemübergreifend konsistent umgesetzt wurden. Und im Abnahmetest wird zuweilen schmerzlich klar, dass wichtige Kundenbedürfnisse übersehen wurden, die dann nur noch per Change Request – oft verbunden mit massiver Zeit- und Budgetüberschreitung – nachzuholen sind.

Was leisten Requirements Engineers im Projekt?

RE begegnet diesen Risiken mit einem systematischen Prozess, der aus vier Hauptaktivitäten besteht:



Die eingangs beschriebenen Symptome sind deutliche Hinweise auf Lücken im RE-Prozess des Projekts. Häufig wird RE auf den Verwaltungsaspekt Requirements Management reduziert und die inhaltliche Arbeit mit den Anforderungen vernachlässigt. So könnte etwa in einem Projekt ein Tool bereitstehen, wo verschiedene Parteien formlos beschriebene Anforderungen hinterlegen, die dann bewertet und gegebenenfalls zur Umsetzung eingeplant werden. Ein solches Tool unterstützt zwar wesentliche Aufgaben des Requirements Managements trägt aber nicht dazu bei, dass Anforderungen systematisch erhoben und angemessen

sen dokumentiert werden. Ebenso wenig kann ein Tool dafür sorgen, dass Anforderungen bestimmte Qualitätskriterien erfüllen oder dass Konflikte in den Anforderungen aufgelöst werden. Der Erfolg eines Projekts hängt oft ganz entscheidend davon ab, solche Prozesslücken rechtzeitig zu erkennen und zu beseitigen.

Erste Schritte

Anforderungen bilden das Fundament für nahezu alle Aktivitäten im Software-Lebenszyklus. Insofern erstaunt es nicht, dass die Initiative zur Einführung oder Verbesserung des RE-Prozesses oft von Personen wie Fachspezialisten oder Entwickler ausgeht, für die RE nicht die primäre Aufgabe ist. Eine Zertifizierung im Foundation Level des Certified Professional for Requirements Engineering (CPRE) schafft das hierfür notwendige methodische Grundlagenwissen und schärft den Blick für mögliche Schwachstellen im RE-Prozess.

Optimierungen

Ist der grundlegende RE-Prozess erst einmal in einer Organisation etabliert, ergibt sich meist eine Fülle von Verbesserungspotentialen. Zwei wichtige Handlungsfelder sind dabei einerseits die eingesetzten Dokumentationstechniken und andererseits der Umgang mit Stakeholdern und anderen Anforderungsquellen bei der Ermittlung und Konsolidierung von Anforderungen.

Bei den Dokumentationstechniken geht es darum, dass Anforderungen in natürlicher Sprache häufig unvollständig oder mehrdeutig formuliert werden, da Kommunikation in natürlicher Sprache Transformationsprozessen wie Tilgung, Generalisierung und Verzerrung unterliegt. Diesen Effekten kann man durch Anwendung sprachwissenschaftlicher Erkenntnisse gezielt entgegenwirken. Gleichzeitig bietet sich die Verwendung grafischer Modelle wie Diagramme der UML an, um die Vollständigkeit und Eindeutigkeit der Anforderungen zu verbessern. Es gibt zwei Zertifikate im Advanced Level des CPRE. Das eine hat

Requirements Modeling, das andere Elicitation & Consolidation zum Gegenstand.

Bei der Ermittlung und Konsolidierung von Anforderungen geht es zunächst einmal um die relevanten Anforderungsquellen. Viele Projekte scheitern schon daran, dass wichtige Stakeholdergruppen, gesetzliche Auflagen oder Altsysteme gar nicht als Anforderungsquelle wahrgenommen werden. Bei der Ermittlung von Anforderungen setzt man oft einseitig auf die Befragung der Stakeholder und übersieht dabei leicht implizites Wissen, das den Befragten so selbstverständlich ist, dass sie es gar nicht mehr erwähnen. Hier wären Beobachtungstechniken effizienter, während sich für die Entwicklung innovativer Ideen Kreativitätstechniken wie Brainstorming anbieten. Im Anschluss an die Ermittlung müssen die relevanten Stakeholder natürlich auch in die Konsolidierung der Anforderungen einbezogen werden. Insbesondere sind dabei Konflikte bezüglich der Anforderungen aufzulösen.

Der richtige Umgang mit Stakeholdern erfordert neben Fingerspitzengefühl auch spezielle Techniken, die geübt sein wollen:

CPRE — Advanced Level – Elicitation & Consolidation: 07. - 09. November; SQS Wien

Dieses Seminar empfiehlt sich für alle, die die Grundlagen des RE beherrschen und nun die Zusammenarbeit mit den Stakeholdern effizienter gestalten wollen.

◀ (Dr. Thomas Harms)

[Der Autor]

Dr. Thomas Harms, Senior Consultant und Trainer bei SQS. Sein Schwerpunkt als Berater liegt in den Bereichen Testmanagement und Requirements Engineering. Diese Erfahrungen gibt er als Trainer in den Seminaren der SQS weiter.





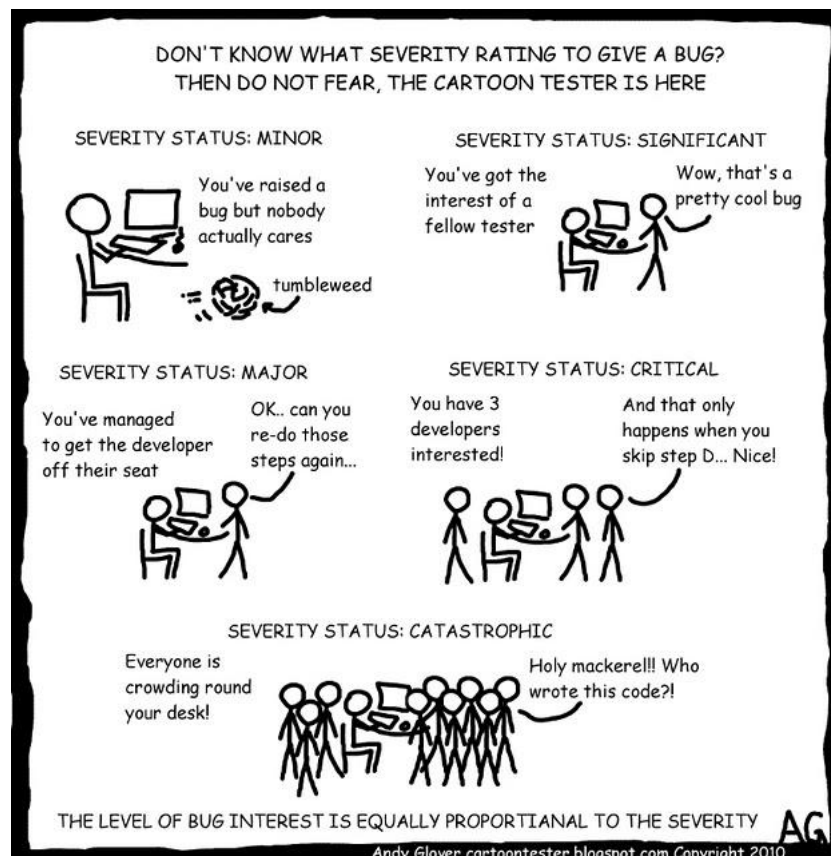
Kurs	Termin	Ort	Anbieter
ISTQB Certified Tester Foundation Level	19.-22.09.2011, Anmeldung	Wien	SQS
	19.-22.09.2011, Anmeldung	Wien	Software Quality Lab
	19.-22.09.2011, Anmeldung	Linz	Software Quality Lab
	7.-10.11.2011, Anmeldung	Wien	Software Quality Lab
	14.-17.11.2011, Anmeldung	Wien	SQS
	28.11-01.12.2011, Anmeldung	Wien	ANECON
ISTQB Certified Tester Foundation Level (engl.)	28.11-01.12.2011, Anmeldung	Wien	ANECON
ISTQB Certified Tester Advanced Level Test Manager	14.-20.09.2011, Anmeldung	Wien	ANECON
	10.-12.10. und 17.-18.10.2011, Anmeldung	Wien	Software Quality Lab
	16.-18.11. und 21.-22.11.2011, Anmeldung	Wien	Software Quality Lab
	12.-16.12.2011, Anmeldung	Wien	SQS
ISTQB Certified Tester Advanced Level Test Analyst	17-21.10.2011, Anmeldung	Wien	SQS
	09.-15.11.2011, Anmeldung	Wien	ANECON
	5.-6.12. und 12.-14.12.2011, Anmeldung	Wien	Software Quality Lab
ISTQB Certified Tester Advanced Level Technical Test Analyst	21.-25.11.2011, Anmeldung	Wien	SQS
IREB Certified Professional for Requirements Engineering Foundation Level	13.-15.9.2011, Anmeldung	Vorarl-	Software Quality Lab
	26.-29.09.2011, Anmeldung	Wien	SQS
	05.-07.10.2011, Anmeldung	Wien	ANECON
	10.-12.10.2011, Anmeldung	Wien	Software Quality Lab
IREB Certified Professional for Requirements Engineering – Advanced Level; Elicitation	07.-09.11.2011, Anmeldung	Wien	SQS
CAT- Certified Agile Tester® Training (Prüfung in englisch)	19.-23.09.2011, Anmeldung	Wien	ANECON
	10.-14.10.2011, Anmeldung	Wien	SQS
	23.-29.01.2012, Anmeldung	Wien	ANECON

Konferenzen

[Konferenz Kalender]

Konferenz	Termin	Ort	Deadline Call for Paper
QA & Test 2011	26.-28.10.2011	Bilbao	Bereits abgeschlossen
Agile Testing Days	14.-17.11.2011	Potsdam	Bereits abgeschlossen
Tricentis Solution Day	14.11.2011	Wien	Bereits abgeschlossen
EuroSTAR	21.-24.11.2011	Manchester	Bereits abgeschlossen
Software Quality Days 2012	17.-19.01.2012	Wien	Bereits abgeschlossen
Belgium Testing Days	12.-14.03.2012	Brüssel	Bereits abgeschlossen

Cartoon



Quelle: Andy Glover, <http://cartoontester.blogspot.com/>

[Software-Qualität im Informatikstudium der Uni Innsbruck]

Von Dr. Michael Felderer

Der steigenden Bedeutung des Themas Software-Qualität, insbesondere des Software-Testen sowohl für die industrielle Praxis als auch für die universitäre Forschung wird durch die Verankerung von entsprechenden Lehrveranstaltungen im Curriculum des Bachelor- und Masterstudium Informatik an der Universität Innsbruck Rechnung getragen. Im Bachelorstudium Informatik ist ein Wahlpflichtfach Software-Qualität und im Masterstudium ein darauf aufbauendes Wahlpflichtfach Advanced Software Quality vorgesehen. Der Ausrichtung der beiden Studiengängen entsprechend, ist die Bachelorlehrveranstaltung Software-Qualität eher praxisorientiert und baut auf den Vorerfahrungen der Studierenden im Bereich Softwareentwicklung auf. Die Masterlehrveranstaltung Advanced Software Quality hingegen ist eher forschungsorientiert und vermittelt formale Konzepte.

Im Sommersemester 2011 wurde die Lehrveranstaltung Software-Qualität im Bachelorstudium Informatik zum ersten Mal angeboten. Die Lehrveranstaltung besteht aus einer zweistündigen Vorlesung und einem begleitenden einstündigen Proseminar zur Vertiefung der Vorlesungsinhalte anhand praktischer Übungen.

Software-Testen im Rahmen der Lehrveranstaltung Software-Qualität

Testen bildet den inhaltlichen Schwerpunkt der Lehrveranstaltung Software-Qualität und es werden die wesentlichen Inhalte des ISTQB Certified Tester Foundation Level Syllabus wie Grundlagen des Testens, der Testprozess, Testtechniken, Testmanagement und Testwerkzeuge abgedeckt. Dies erleichtert den Studenten eine eventuelle spätere Zertifizierung als Certified Tester. Die behandelten Themen werden nicht nur theoretisch erarbeitet sondern auch anhand von praktischen Übungen mit Werkzeugen wie JUnit oder FIT vertieft und eingeübt. Für die Themen Testmanagement und Testen Sicherheitskritischer Systeme konnten wir den erfahrenen Testexperten Dipl.-Ing. Armin Beer als Gastvortragenden gewinnen. Der dadurch hergestellte

Praxisbezug wirkte auf die Studenten zusätzlich motivierend und gipfelte in der Erstellung eigener Testkonzepte durch die Studenten. Neben dem Themenbereich Software-Testen wurden auch die Bereiche Grundlagen der Software-Qualität, Software-Metriken, Software-Prozesse sowie Software-Infrastruktur näher behandelt.

Positive Rückmeldung durch die Studenten

Die studentische Rückmeldung aus der Lehrveranstaltung ist äußerst positiv (einige Studenten konnten die erlernten Techniken schon parallel zur Lehrveranstaltung in anderen Softwareprojekten einsetzen) und manifestiert sich auch im großen Interesse an Bachelorarbeiten im Themenbereich Software-Qualität im allgemeinen und Software-Testen im speziellen.

Ausblick auf die kommenden Semester

Die Bachelorlehrveranstaltung Software-Qualität wird im Wintersemester 2011/12 erneut abgehalten, die Masterlehrveranstaltung Advanced Software Quality wird 2012 das erste Mal angeboten. Beide Lehrveranstaltungen werden dann im Jahresrhythmus wiederholt. Gastvortragende, welche die LV mit Praxisberichten bereichern, sind sehr willkommen. Bei Interesse bitte ich Sie, direkt mit mir unter michael.felderer@uibk.ac.at in Kontakt zu treten.

◀(Michael Felderer)

[Über den Autor]

Dr. Michael Felderer ist wissenschaftlicher Mitarbeiter am Institut für Informatik der Universität Innsbruck. Zu seinen Arbeitsschwerpunkten zählen Testen von Software und Requirements Engineering. Neben seiner wissenschaftlichen Tätigkeit ist Dr. Felderer als Berater im Bereich Software-Qualität und Anforderungsmanagement tätig.



[Öffentliche ISTQB Prüfungstermine]

In Wien finden die öffentlichen Prüfungen im Normalfall am letzten Mittwoch in jedem geraden Monat statt.

Nächste Termine:

- 19.10.2011
- 15.12.2011

Für Anmeldungen und nähere Informationen wie Ort und Beginnzeiten kontaktieren Sie bitte katja.liebau@isqi.org ◀

[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

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
Wenn Sie dieses Magazin abbestellen wollen, senden Sie bitte eine Mail mit Betreff „Storno Magazin“ an 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.

[Introducing a Model-based Automated Test Script Generator]

Martin Steinegger, Hannu-Daniel Goiss

This article provides an introduction to a new way of generating test scripts for GUI testing (web based or rich clients) using a model-based testing approach and how this alters the test process. The described test code generation tool (Abstract Testing Tool) has been developed by the authors in cooperation with one of Accenture's clients. The generated automation code can then be used for performance testing or functional testing.

The abstraction layer, which is at the core of this approach, will be explained at the beginning. Afterwards we will give an overview of how this approach changes the testing process and discuss advantages and disadvantages.

Introduction

The most common way to create test scripts for test tools is (unfortunately) still „capture and replay“, even though the benefits of treating test automation like a software development project have been demonstrated repeatedly. In a capture and replay approach, the test tool simply records the users' actions and generates a script from them, which can be used to execute tests. This approach has several drawbacks. Every step has to be captured each time a script is generated, even if the step is contained in every script. Thus tools employing a modular approach have been developed but they still require in-depth knowledge of the testing tool and scripting language, as it is necessary to adapt the test scripts after capturing them from complex applications in order to make the scripts more robust.

These problems can be addressed by using an abstraction layer, which we will call the “Abstract Testing Tool (ATT)”. ATT introduces an abstraction layer, which separates the testing tool from the test script and introduces a representation of the System under Test (SUT), with all its pages and navigation paths and a meta-scripting language. The tester generates scripts by navigating through the SUT in a GUI, provided by ATT.

Abstraction Layer

This section gives a short overview of how to create the tool's abstraction layer. The following key topics will be explained: **test object model**, **test script** and **code generation**.

The test object model contains a representation of the SUT with all its pages and navigation paths. The test scripts are scripts, which test the application by navigating using the navigation paths that have been stored in the test

object model. Code generation is the process of creating scripts for the test tools, combining the test scripts and the test object model. See figure 1

The test object model is an abstract model representing the test object. The minimum requirement is that it contains the possible interactions within the SUT (actions), which will form the test scripts. For example, on a website these interactions would be clicks on links or buttons. The action in the object model has to contain every component (for example parameters of a link) necessary to be performed.

Pages and a mechanism for verification should also be part of the test object model. Pages represent start and end points for actions. For example, google.com and the search results could be pages, while the search button (with all the necessary parameters) would be an action, leading from the start page to

the search result page. Verification objects are a mechanism to check the accuracy of the end points of actions. Usually verification objects are a set of strings, pictures or other elements that are part of the end point. For example, for a Google search action, the expected result list could be a verification string. If the end point of the action doesn't contain the search action, the verification will not be successful. Verification objects can be connected either to actions or pages and there are many different possible types, like positive or negative checks, etc. The model can thus be arbitrarily complex. For example, events can be added to verifications to react on errors. Possible reactions could be error logging, creation of screenshots or environment snapshots.

Continued next page

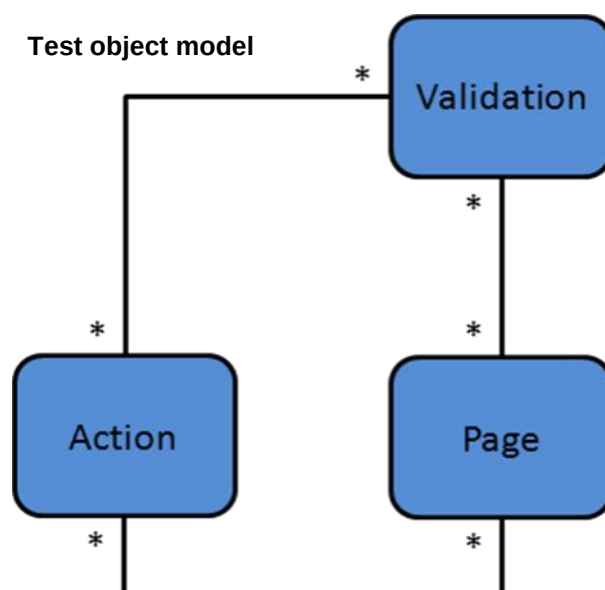


Figure 1: Test object model

Testautomatisierung

The test script

The test script is basically a list of actions that have been defined in the test object. The meta-scripting language itself can be created in many different ways (Domain Specific Language) but it should be able to be read by a human (not that developers are not humans, but we digress...).

The tester should be able to create test scripts using a GUI. Using UML activity diagrams for representing test scripts in the GUI could be a fast way to create such a GUI, but can lead to accuracy issues because the UML editors are not made for script creation. An own language will bring more flexibility and accuracy. But activity diagrams can also be used for visualization of your own language. See figure 2

The following is an example of how the scripting language could look. The object "script" is the root node and contains many steps. An example in XML could be:

```
<script>
  <step link="technical or functional key from the test object model"/>
</script>
```

A step is an action the user can perform within the application. Every step needs a reference to an action from the test object model. It is important to distinguish between steps and actions, because actions are unique (for example there is only one action for performing a Google search), but the same action can be referenced to many steps in a script (the same Google search can be performed multiple times in the same script).

Operations represent control structures that can affect the script flow or commands that have to be executed. Complex scripts have to be nested (e.g. if - else, loop exit, log, delay, run, waitUntil, skip, store, check). Functional scripts are often flat, because they have the focus to check exactly one functionality at a time. In contrast technical scripts are rather complex due to contingencies that appear because of the test data and necessary flow controls.

```
<operation type="definition" >
  <step link="..." />
  <step link="..." />
  <operation type="..." />
</operation>
```

When deciding which operations to implement the targeted test tools have to

be taken into account. Some operations might not be possible with all test tools, so they may require add-ons to the test tool. This is important as this approach can ensure test tool independence if the scripts are to be used with many different test tools.

See figure 3

Abstraction layer for the code generation

This section describes how the test object model and the test script is being transformed to source code (for the testing tools) or documents. A generator engine is needed to merge the test object model and test script. The output of the generator is an object graph, which will be used as input to create test source code itself. This mechanism is very useful to change content before it will be sent to a template engine (templates will be described in the next paragraph). Figure 4 is a visualization of a merged test object model and test script. See figure 4.

Continued next page

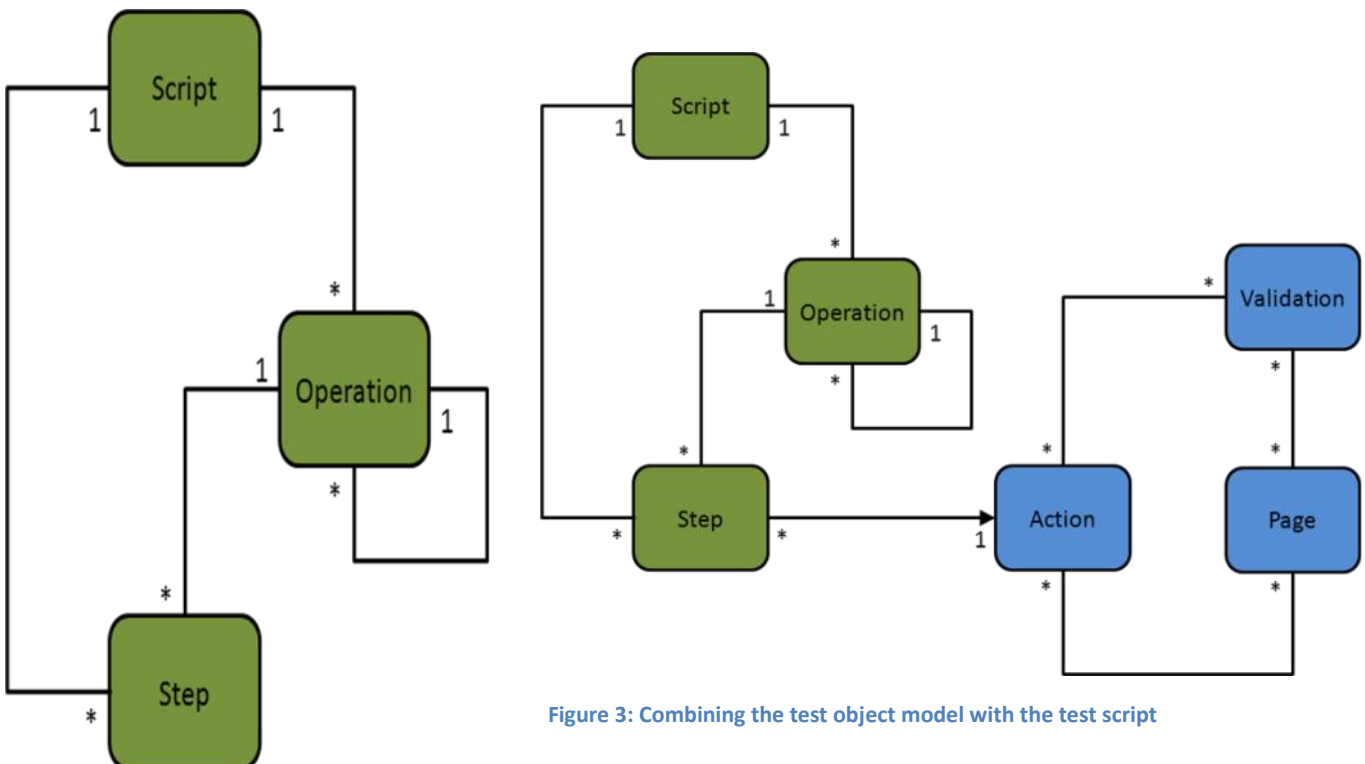


Figure 2: The test script

Figure 3: Combining the test object model with the test script

Testautomatisierung

Template engines (e.g. Velocity, Xpand or Hamlets for Java) can be used to generate code. This technology has the advantage of being very flexible. The engine can be used to convert a graph to the desired format. Templates have to be developed for every desired format. For example, one template could be created to generate HP LoadRunner code and another could be generated to create Micro Focus SilkPerformer code. However, template languages can be tempted to fudge (duplicate code, ...), therefore the template code should be kept structured and modular. Template languages are not made for complex logic. If it is necessary to create complex logic, it can be delegated to the target programming languages (for example C oder Java) by creating a function call.

Keep the following points in mind:

- Don't repeat yourself
The most important thing is to create a test object model that is maintainable by avoiding redundancy.
E.g. to reduce duplicate interactions that exist on many pages (e.g. main navigation), they have to be defined one time and, if needed, referenced.

- Keep it simple
Think about the complexity and your requirements. The scripting language should be as easy as possible. Only implement features that are really needed and cannot be replaced by a combination of simple features. Complex functions are rarely used and not required.
- Don't forget versioning
Changes to the model and the scripting language can occur with every release or deployment of the SUT. Many people have to be able to create scripts and make changes in the model. This is a frequent source of errors. These changes have to be traceable and enable rollback. Versioning is a key feature.

How does the tool change and improve the test process?

See figure 5.

The graphic illustrates the difference between conventional test script creation and test script generation using

ATT. The main difference is that in the conventional way the main task is to use the capture functionality to generate the script and to make manual adaptations until the script is capable of running. Using this approach, the most cost-and time-intensive part is manually adapting test scripts (and often multiple scripts for the same reason), as in large applications the test script usually has to deal with a wide variety of pages and test data.

Using the ATT approach, the most cost-and time-intensive task is filling and maintaining the test object model. Once the test object model is filled with the bulk of the SUT, almost all test scripts can be generated quickly.

In the conventional approach, the most cost-intensive task (manually adapting the code of the test scripts) has to be performed every time a script is generated, while in the ATT approach the most cost-intensive task (filling the test object model) has to be performed once, as during the test phase the test object model only has to be maintained in case of changes in the application.

The following are some **advantages** of ATT:

Continued next page

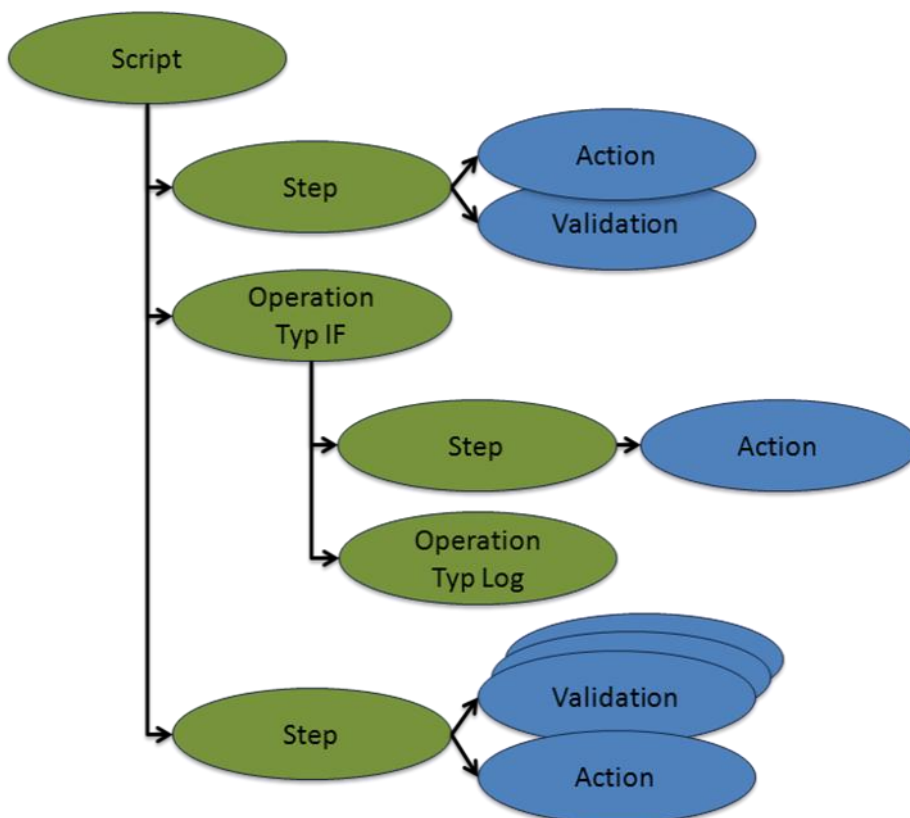


Figure 4: Merging the Test object model and the test script

[ATB Network]

Was wären Testerinnen und Tester ohne ein Netzwerk?

Am letzten Mittwoch jedes geraden Monats gibt es die Möglichkeit mit anderen Testerinnen und Testern Kontakt aufzunehmen.

Achtung: nächstes mal wegen Feiertag bereits am 19.10.2011!

Rechtzeitige Anmeldung sichert einen der begehrten Plätze.

Infos unter <http://www.austriantestingboard.org/>

Anmeldung unter: backoffice@austriantestingboard.at

Werden Sie auch Mitglied in der XING Gruppe Austrian Testing Board & Friends"

<https://www.xing.com/net/pria946f6x/atb/>

(Karl Kemminger)

Testautomatisierung

- **Test scripts are generated independently from the test tool in use**

- It is possible to use the same test scripts in different test stages (for example test scripts can be generated for different performance test tools for the performance testing stage and various scripts for automated test tools can be generated for the functional testing stage). This changes the test process significantly. Usually one team is responsible for generating and executing test scripts for each test phase. With the ATT approach, one team can be in charge for generating test scripts for the whole test process, while the test stage teams only need to be responsible for test script execution and analysis.

- In case of changing the test tool (for example a switch from one vendor to the other) all test scripts can be reused. The use of a new test tool requires the development of a new template and the regeneration of the test scripts using this new template. Regression tests can be defined independently from the test tool and can be used with any test tool.

- **Changes in the SUT do not require direct changes to the ATT test scripts.** Should changes be necessary, the test object model has to be adapted and test scripts can be regenerated. This significantly reduces the maintenance effort, as changes only have to be done at one central location (test object model).

- **Test scripts can be generated without specific knowledge of the test tool and of the programming language used.**

Manual adaptations to test scripts, which require programming skills, are usually not necessary. But as discussed earlier, ATT and its functionalities should be kept simple, as complex functionalities make it more difficult to use. This

makes manual adaptations necessary in case of functionalities that have not been implemented in ATT. For example, if ATT does not have the possibility to test AJAX functionality, the generated scripts have to be adapted manually.

- This makes it possible for “functional” test team members to generate test scripts, as it does not require programming skills. The only thing they have to do is to navigate through the test object model to generate their test scripts.

- A special training period is not necessary when switching the test tools because the test scripts are automatically generated by ATT. It is still necessary to train the responsible persons to execute the test scripts and analyze test results.

- **It makes sense to use ATT if there are steps in the test scripts that can be reused in other test scripts.** The introduction of ATT does not make sense if every script is unique.

Continued next page

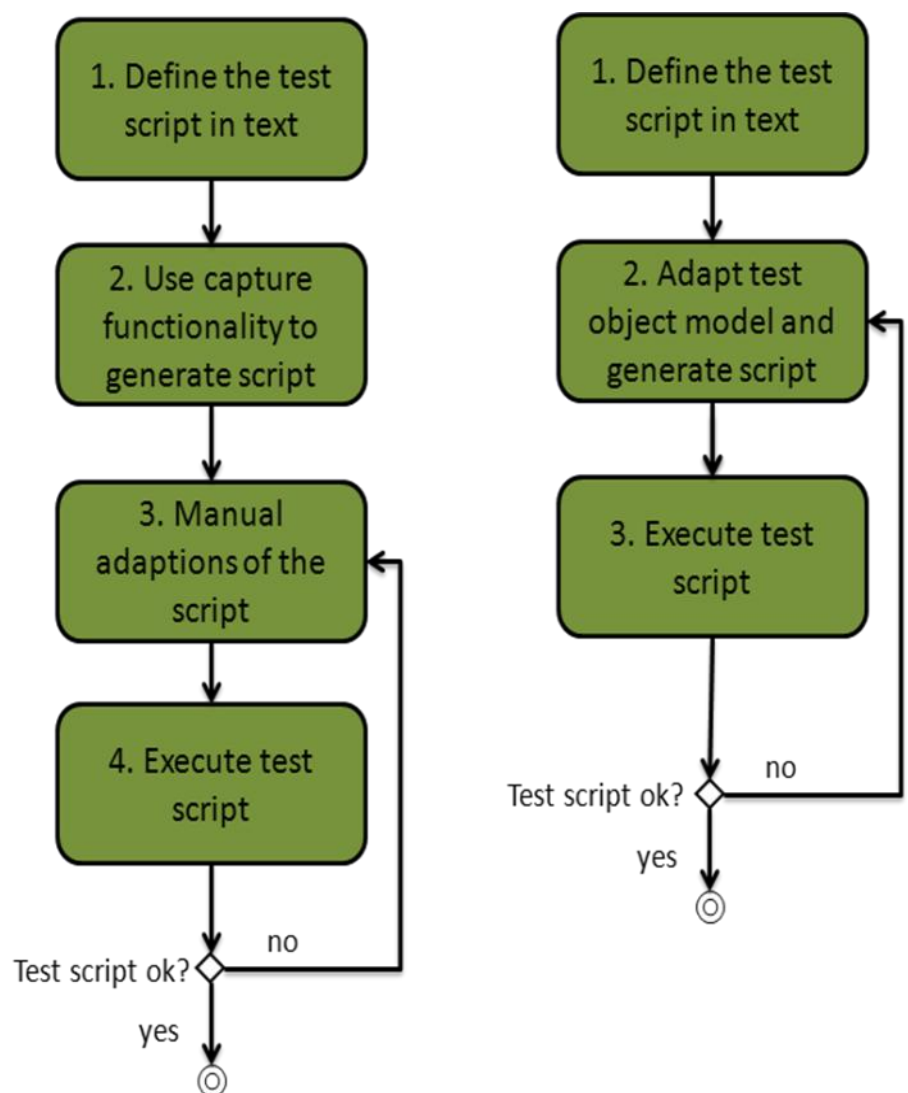


Figure 5: Changing the test process (left side: conventional flow; right side: test script generation using ATT)

Some of the **disadvantages** of ATT are:

- The test object model is a frequent source of errors. Granting write access to a high number of people should be avoided. They might change things that are still valid and should not be changed because of impacts to other test scripts.
à Only a few people with in-depth knowledge of the test object model should be allowed to make changes. Change control and review processes for each version are important as for any other test artifact.
- As ATT should be kept simple, some parts of the test object might not be testable using only ATT. For example, when testing AJAX in a web application, it will not be possible to test this if ATT does not provide the possibility to do so. The requirements for ATT need to be continuously validated and updated in the ATT where necessary
- While ATT adds test tool independence to the testing process, it generates a new dependence on ATT.

Conclusion

The described approach adds a new test script generation approach to the testing process. It has been successfully used in practice and has been proven to be effective with regards to:

- time- and cost-efficiency
- test script generation for different test tools
- re-usage of functionality across different test scripts
- usage of test scripts with a SUT, which is still under development and has frequent GUI changes.

◀ (Martin Steinegger,
Hannu-Daniel Goiss)

[The authors]

Martin Steinegger, Hannu-Daniel Goiss

Email:

Martin.Steinegger@accenture.com,
Hannu-Daniel.Goiss@accenture.com

Address: Accenture GmbH, Campus Kronberg 1, D-61476 Kronberg im Taunus

Accenture is a global management consulting, technology services and outsourcing company with more than 215,000 people in 53 countries. Accenture Test Services has been providing testing services for more than two decades, both on-site and through our Global Delivery Network.

Martin Steinegger is working at Accenture in performance- and security testing, while Hannu-Daniel Goiss is involved in performance- and integration testing. They have both been with Accenture for over 2,5 years, working for multi-national clients in various industries. They hold certifications such as ITIL Foundation Level and ISTQB Certified Tester Foundation Level.



Martin Steinegger



Hannu-Daniel Goiss

[QAMP Reloaded]

Von Katja Piroué

Liebe
Leser,



das neue QAMP ist nun 2 Monate alt. Wenn ich auf den „Zustrom“ in der Xing Gruppe sehe- hat es keine Veränderung ergeben. Ich bin also gespannt, welche mittelfristigen Auswirkungen das neue Zertifizierungsschema hat.

Allen QAMPs kann ich empfehlen, immer wieder unter http://www.qamp.org/qamp_news.html die angekündigten Benefits oder auch Zusatznutzen abzufragen. Oder haben Sie sowieso bei der Auslosung der Cebit Karten mitgemacht?

Nun aber ein Aufruf in eigener Sache: ich werde für das ATB auch die QAMP Seite <http://www.austriantestingboard.at/node/57?> aktualisieren.

Dazu möchte ich die noch nicht gelisteten QAMPs bitten, mir ein Foto zuzusenden, welches ich dann dort auch mit veröffentlichen kann. Wer kein Foto auf der ATB Seite haben möchte, kann mir dies selbstverständlich ebenfalls mitteilen. Für alle, die schon eines dort haben - hier ist die Möglichkeit, dieses Foto durch ein aktuelleres auszutauschen.

Für weitere Fragen können Sie sowohl unter <http://www.qamp.org> weiterlesen, sich als Mitglied der QAMP Gruppe in XING eintragen und mitdiskutieren: <https://www.xing.com/net/pri6243ecx/qamp> oder sich auch direkt an mich wenden.

◀ (Katja Piroué)

[Die Autorin]

Katja Piroué, M.A.

QAMP in Österreich
und Moderatorin der
Xing Gruppe.



Katja Piroué ist seit vielen Jahren als Qualitätssicherungsexperte in Deutschland und Österreich tätig, ATB Mitglied und Trainerin für ISTQB Zertifikate. Mit Sitz in Wien ist die Selbständige im Test- und Anforderungsmanagement für Ihre Kunden tätig.

www.it-qualitaetssicherung.eu