

[Editorial]



Im ersten Insider 2014 möchte ich noch einen kurzen Rückblick auf das ereignisreiche Jahr 2013 machen: neben den Neuerungen, die uns vor allem Boardintern enorm forderten, wie z.B. die Bereitstellung der neuen deutschen Advanced Level Lehrpläne samt Überarbeitung der Prüfungskataloge sowie die damit verbundenen Neu-Akkreditierungen der Trainingsunterlagen, begeistert mich vor allem der riesige Erfolg unserer neu initiierten ATB Expertentreffen. Dank der tollen Organisation von Weixi, dem ich an dieser Stelle ein großes Dankeschön sagen möchte, hat sich das ATB Expertentreffen von Mal zu Mal mit tollen Rednern und Diskutanten zum Fixpunkt in der Test Community etabliert. Daher setzen wir dieses erfolgreiche Format natürlich auch 2014 fort. Doch nun zum Insider: auch an Karl, unseren Chefredakteur, ein riesiges Dankeschön für sein tolles Engagement! Er konnte in dieser Ausgabe wieder spannende Artikel zu topaktuellen Themen der Testing-Szene zusammenstellen: Haftungsfragen bei Requirements im agilen Umfeld und im Vergleich dazu eine Betrachtung zu Abnahmetests im nicht-agilen Umfeld. Weiters werden Einblicke in die Möglichkeiten des Cloud Testings eröffnet und das Thema Real Application Testing behandelt. Abschließend möchte ich mich ganz herzlich bei allen bedanken, die unserem ATB Charity-Punsch-Event zu einem tollen Ergebnis verholfen haben - Details zu allen Themen findet Ihr auf den nächsten Seiten. Einen tollen Start ins Jahr 2014 und Gut Test

◀ (Helmut Pichler)

[ATB Expertentreff]

Von Alexander Weichselberger

Auch im Jahr 2014 gibt es wieder unsere gut etablierten Expertentreffs. Geplant sind 5 Termine (März, Mai, Juli, September, November). Die genauen Termine und Themen sind noch nicht definiert. Mögliche Themen sind

- Destruktives Testing
- Softskills—Kommunikation
- Marketing für Softwaretest
- SW Testing vom Scratch weg
- g'scheites Bugreporting

Wenn Sie weitere Themenvorschläge haben oder als Referent oder Diskutant mitmachen wollen, wenden Sie sich bitte an weichselberger@austriantestingboard.at

Nutzen Sie die Möglichkeiten zum Erfahrungsaustausch im Rahmen der ATB Expertentreffs. Informationen zu den Veranstaltungen und zur Anmeldung finden Sie auf unserer Homepage unter <http://www.austriantestingboard.at/expertentreff>

Ihr
Alexander Weichselberger
ATB Expertentreffs

◀ (Alexander Weichselberger)

Einen Bericht vom letzten Expertentreff finden Sie auf Seite 16.

[ATB Charity]

Von Helmut Pichler

Am Montag, dem 16.12.2013 fand der diesjährige ATB Charity Punsch statt, bei der wieder eine namhafte Summe gesammelt werden konnte. Die Summe kommt wieder der „Stiftung Kindertraum“ zugute, die damit Kindern ihre Herzenswünsche erfüllen kann.

Einen Bericht finden Sie auf Seite 13.



◀ (Helmut Pichler)

[Inhalt]

■ Goldpartner	Seite 2 - 3
■ Abnahmetest von agil entwickelter Software	Seite 4
■ Software-Tests in der Cloud	Seite 5 - 6
■ Rechtliche Fragen bei Requirements Engineering	Seite 7 - 11
■ Umfrage / Awards	Seite 12
■ ATB Charity Event	Seite 13
■ ISTQB / Events	Seite 14
■ ASQF	Seite 15
■ ATB Expertentreff	Seite 16
■ Seminare	Seite 17 - 21
■ Konferenzen, Silberpartner, iSQI	Seite 22
■ Real Application Testing	Seite 23 - 27
■ Cartoons	Seite 28
■ Impressum	Seite 19



Service Virtualisierung

Der virtuelle Weg im Software-Test

Dienstag, 28. Jänner 2014 | 08:30 - 11:00 Uhr | IBM Forum | 1020 Wien

Steigende Anforderungen, weniger Zeit sowie immer kürzer werdende Releasezyklen gepaart mit knappen Budget und time-to-market Druck- sind auch Sie mit diesen Ansprüchen konfrontiert? Wir zeigen Ihnen einen Weg, diesen Herausforderungen effizient im Software-Test zu begegnen.

Weitere Informationen und Anmeldung auf www.anecon.com/service-virtualisierung-event



Bezahlte Einschaltung

Das neue Academy Program 2014/15 ist verfügbar!



- Neue Seminare und Trainings
- 4 Lehrgänge
- Aus- und Weiterbildungsempfehlungen für 11 verschiedene Zielgruppen

» [Programm](#)

NEU

- Moderationstechniken im Requirements Engineering » [Termine](#)
- Funktionale Sicherheit » [Termine](#)
- Continuous Integration für agile Teams » [Termine](#)

SPEZIALPREIS

ISTQB® Certified Tester Foundation Level – Englisch

- » 20. – 23.01.2014: [Wien](#) [München](#)

SPEZIALPREIS

ISTQB® Certified Tester Foundation Level

- » 20. – 23.01.2014: [Wien](#) [Linz](#) [Graz](#) » 21. – 24.01.2014: [Lustenau](#) [München](#)
- » 24. – 27.02.2014: [Wien](#) [Linz](#) [Graz](#) » 25. – 28.02.2014: [Lustenau](#) [München](#)

SPEZIALPREIS

IREB® Certified Professional for Requirements Engineering Foundation Level

- » 27. – 29.01.2014: [Wien](#) [Linz](#) [Graz](#)
- » 28. – 30.01.2014: [Lustenau](#) [München](#)
- » 11. – 13.02.2014: [Wien](#) [Linz](#) [Graz](#) [Lustenau](#) [München](#)

14. – 16.01.2014
Hotel Savoyen
Wien



Haben Sie schon Ihre Tickets gebucht?

▶ [Ticket Shop](#)

» [Programm](#) » [Keynotes](#)

Bezahlte Einschaltung

Aviso: Acceptance Café 1. April



Thema: „Agiles Testen: Vom Test First bis Test Nonstop“

Kein Scherz: Am **1. April 2014** ist es wieder soweit: Das Acceptance Café öffnet zum dritten Mal bei **freiem Eintritt** seine Pforten. Es freut uns sehr, dass wir den renommierten Experten und Fachbuchautor **Tilo Linz** als Vortragenden gewinnen konnten. Er war lange Zeit Präsident des German Testing Board und auch ISTQB-Präsident.

In seinem Vortrag geht er der Frage nach, wie Testen in **agilen Projekten** bzw. **Scrum** funktioniert, und wie die aus dem **V-Modell** bekannten Teststufen hier konkret aussehen.

Wann: 1. April 2014, Check-in ab 8:30 Uhr, Beginn: 9:00 Uhr,
Veranstaltungsende: 11:30 Uhr

Wo: Café Griensteidl, Michaelerplatz 2, 1010 Wien

Anmeldung: [hier!](#)



BDC-Trainings: Jetzt Frühbuche Preis sichern

ISTQB® Certified Tester - Foundation Level: 24. bis 27. Februar 2014 (Wien)

Software Test für Embedded Systems: 10. bis 11. April 2014 (Wien)

[Details & Anmeldung](#) zu unseren Terminen in **Wien, St. Pölten, Graz und Salzburg**

Frühbuche Preis bis vier Wochen vor Trainingstermin!

SQS Academy - Lernen mit den Experten



sqs.com

Über SQS Software Quality Systems

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.

Aktuelles Vorteils-Angebot



Buchen Sie jetzt einen Kurs für das 1. Quartal 2014, und Sie erhalten 10% Rabatt auf den Listenpreis.

Kommende Seminartermine der SQS in Wien:

- ISTQB® CT Foundation Level, 03.-06.02.
- ISTQB® CTAL Test Analyst, 24.-27.02.
- ISTQB® CTAL Test Manager, 10.-14.03.
- ISTQB® CTAL Technical Test Analyst, 24.-26.03.
- ISTQB® CT Foundation Level, 07.-10.04.
- Certified Agile Tester® (CAT), 05.-09.05.
- ISTQB® CT Foundation Level, 02.-05.06.

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

[Abnahmetest von agil entwickelter Software im nicht-agilen Umfeld]

von Ernő Büki

Über den Glaubenskrieg [1] zwischen Befürwortern von agil bzw. nicht agil (sequenziellem bzw. klassischem Phasenmodell) hat wahrscheinlich jeder in der Branche bereits gehört. Wir wissen mittlerweile auch, dass sowohl agile als auch nicht agile Softwareentwicklungsmethoden - abhängig von diversen, meistens einander widersprechenden Aspekten des Projekt-Gesamtprozesses - durchaus Legitimität haben können.

Gibt es bei dem gegebenen Projekt allerdings weitere Faktoren (z.B.: Heterogenität der Systemlandschaft, Diversität der Lieferantenkreise, Unterschiede der Releasezyklen der einzelnen Software) im Projektumfeld, die nicht unbedingt mit der Softwareentwicklung zu tun haben, beginnt man rasch darab zu zweifeln, auf welcher Seite der „Front“ man letztendlich steht.

Im Abnahmetest von sehr heterogenen Systemlandschaften, in denen Software von einer hohen Anzahl verschiedener Lieferanten stammt, findet man meistens beide Welten der Softwareentwicklung wieder. Es kann sogar den dritten Fall geben, dass man als Kunde die vom Softwarelieferanten verwendete Methode gar nicht kennt bzw. kennen darf oder gar der gleiche Lieferant verschiedene integrierte Produkte auf unterschiedliche Weise entwickelt hat. Es kann somit eine Mischform aus beiden Entwicklungsmodellen auftreten, nämlich eine Mischung aus einem agilen Software-Entwicklungsmodell und einem nicht-agilen Abnahmetest und einer nicht-agilen Inbetriebnahme.

Im Bereich der Zahlungsverkehrs- und Transaktionsverarbeitenden Systeme kommen solche heterogene Szenarien häufig vor. Auch wenn jeder Hersteller bei sich entsprechende Abnahmetests durchführt, steigt die Wichtigkeit des Abnahmetests durch den Kunden. Dieser kann entweder nur aus einem Factory Acceptance Test (FAT) oder Site Acceptance Test (SAT) oder aber aus beiden bestehen. Gewisse Tätigkeiten gewinnen an Gewicht, die ohne die Heterogenität weniger bedeutend wären:

- Eine gründliche Impact Analyse (Auswirkungsanalyse), die auch den Aspekt „agil oder nicht agil entwickelt“ der Systeme berücksichtigt
- Der User Acceptance Test (UAT) gewinnt an Bedeutung, die Erstellung der Testfälle benötigt viel Fachwissen (Fachbereichsmitarbeiter müssen sich intensiv beteiligen).
- Die Anforderung von aussagekräftigen Release Notes jeder Software ist unabdingbar.
- Die Erstellung eines detaillierten Release-Planes ist immens wichtig, da in der Regel die verschiedenen Systeme nicht zur selben Zeit zum Test freigegeben werden. Ergo müssen die Test- und Integrationsdurchführung präzise geplant und geregelt zustande kommen.
- Es werden hohe Anforderungen an das Konfigurationsmanagement sowie Traceability gestellt.

Um eine fundierte Abnahmeempfehlung liefern zu können, benötigen Tester in solchen Konstellationen Expertise. Obwohl die Impact Analyse einen immer zentraler werdenden Stellenwert hat, können bei weitem nicht alle Auswirkungen a priori ermittelt werden. Die Heterogenität und Komplexität der Systemlandschaften generiert Effekte und Schwachstellen, die „emergent properties“ (auftauchende Eigenschaften) [2] genannt werden können, und die in keinem der Subsysteme zu finden sind, da das resultierende „Zusammenspiel“ mehr als die Summe der einzelnen Systeme ist [3].

Die Abnahme von solch komplexen Systemlandschaften erfolgt meistens als „nicht-agiler“ Prozess, da die inkrementelle Abnahme pro Iteration mit prohibitiv hohen Kosten verbunden wäre. Die Hauptgründe für die hohen Kosten sind: der User Acceptance Test blockiert viele Mitarbeiter aus dem Fachbereich, die Bereitstellung der Systemlandschaft für das Testen ist mit hohem Aufwand verbunden (Konfigurationsmanagement). Dies relativiert wieder den Ausgangspunkt des genannten Glaubenskrieges. Im Fokus steht die Qualität, das darf man nie außer Acht lassen.

Zusammenfassung

Beim Abnahmetest von sehr heterogenen Systemlandschaften muss das Testmanagement situationsbedingt auf das Kosten-Nutzen-Verhältnis der möglichen Abnahmetest-Vorgehensweisen achten, und den Testprozess dementsprechend gestalten. Im Bereich Zahlungsverkehr empfiehlt sich dadurch meistens ein nicht-agiler Abnahmetest-Prozess, unabhängig davon, ob die Subsysteme agil oder nicht-agil entwickelt worden sind, hauptsächlich aus Kostengründen.

◀ (Ernő Büki)

LITERATUR:

[1] M. Ekssir : Warum Glaubenskriege?, in: SQ Magazin, Juni 2013

[2] J. Bloomberg, R. Schmelzer: The Enterprise as Complex System, in: The Agile Architecture Revolution: How Cloud Computing, Rest-Based SOA, and Mobile Computing are Changing Enterprise IT, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2013

[3] M. Winter, M. Ekssir, H.M. Sneed, R. Seidl, L. Borner: Der Integrationstest, Hanser Verlag, München, 2013

[Der Autor]

Ernő Büki studierte Informatik und Wirtschaft und ist bei der BDC IT-Engineering GmbH als Projekt- und Testmanager tätig. Er greift dabei auf seine Erfahrung als Programmierer, Requirements Engineer, Projektleiter, Tester und Testmanager zurück, und verfügt über Branchenexpertise in den Bereichen Finanzdienstleistungen, Einzel- und Großhandel und Industrie.



[Pro und Contra: Software-Tests in der Cloud]

von Sven Euteneuer

Die Nachfrage nach Cloud Testing nimmt zu. Die erhofften und versprochenen Vorteile lassen sich durch die Angebote allerdings oft nicht erreichen.

Die Versprechen sind bekannt: Es entstehen geringere Kosten als beim traditionellen Vorgehen, zudem ist es flexibler und benötigt weniger Zeit. Auch Software-Tests sind mittlerweile im Cloud-Zeitalter angekommen. Zahlreiche Anbieter von Testdienstleistungen in der Wolke treffen daher auf eine steigende Nachfrage. Ein weiterer, wichtiger Vorteil des Testens via Cloud: Echtzeitbedingungen im Web lassen sich hervorragend simulieren.

Soweit die Realität in Broschüren und in Projektplänen. Im Arbeitsalltag erreichen viele Unternehmen die gesteckten Ziele nicht. Der anvisierte Mehrwert und Effizienzgewinn durch Cloud-Testen stellt sich in diesen Fällen gar nicht oder nur teilweise ein. Was sind die wichtigsten Ursachen?

Technik ist nicht alles

In der Diskussion um Software-Tests in der Cloud geht es heute vor allem um Aspekte, die auf die Technik abzielen – zum Beispiel bezüglich der Definition von Service Level Agreements (SLAs): Wie lassen sich diese auf der technischen Ebene monitoren? Wie lässt sich die Testplattform in der Cloud gegen Einbrüche absichern? Was ist notwendig, um die Verbindung zur Cloud-Plattform garantiert unterbrechungsfrei (redundant) und ausfallsicher zu implementieren? Solche Fragen sind wichtig. Und Cloud-Dienstleister müssen sie beantworten können.

Doch Technik ist nicht alles. Und darf auch nicht am Anfang von Software-Testprojekten in der Cloud stehen. Denn letztlich folgt die Technik jenen strategischen und konzeptionellen Weichenstellungen, die Unternehmen zuvor vorgenommen haben – oder eben auch nicht. Die Versäumnisse, die an dieser Stelle oft gemacht werden, zeigen beispielhaft, dass das Thema **Industrialisierung der IT** häufig noch unterschätzt wird.

Gerade wenn Unternehmen IT-Prozesse auslagern und damit die Verantwortung dafür teilweise abgeben, sind reife und standardisierte Abläufe eine unabdingbare Voraussetzung. Der Wille einer Organisation, Cloud-Services zu nutzen, kann noch so stark sein; ist die Reife jedoch nicht gegeben, stellen sich keine signifikanten Mehrwerte ein.

Ein Symptom für mangelnde Reife ist die Tendenz, das Rad immer wieder neu zu erfinden. So kann es durchaus vorkommen, dass in einer Organisation mehrere Ansätze des internen Software-Testens nebeneinander existieren, dazu noch Lösungsansätze, die auf billige, manuell testende Offshore-Dienstleister setzen, sowie weitere Vereinbarungen mit Cloud-Anbietern für das Software-Testen. Eine solche Vielzahl unterschiedlicher Lösungen lässt sich nicht mehr zusammenführen und neutralisiert alle Kostenvorteile, die Cloud-Lösungen bieten könnten.

Ziel muss es hingegen sein, sich auf einen Satz konsensfähiger Services des Software-Testens festzulegen und diese für das ganze Unternehmen zu standardisieren. Denn nur mit geordneten Testabläufen macht das Buchen von Cloud-Diensten überhaupt Sinn, da diese ja per se standardisiert sind. Nur einheitliche und klar definierte Testprozesse lassen sich effizient in der Cloud abbilden.

Welche Tests in der Cloud funktionieren

Diese Industrialisierung der Testprozesse bedeutet dabei nicht unbedingt Testautomatisierung. Natürlich lässt sich die Cloud einsetzen, um bestimmte Aspekte des Testens zu automatisieren. Dies trifft besonders auf Last- und Performancetests zu, bei denen Lastszenarien oft wiederholt ablaufen müssen, oder bei denen eine besonders realitätsnahe Messung des Systemverhaltens – und damit ein Test der Produktionsumgebung – gewünscht ist.

Für andere Aufgaben der Automatisierung – wie den Test grafischer Benutzeroberflächen – stellen

Cloud-Lösungen heute noch Nischenprodukte dar. Dies liegt darin begründet, dass sich die Automatisierung komplex vernetzter Systeme der klassischen IT effizienter mit lokalen Werkzeugen umsetzen lässt. Betrachtet man hingegen die Automation leichtgewichtiger Systeme wie mobiler Apps ergeben sich größere Effizienzpotenziale.

Ganz anders stellt sich die Situation beim Thema Testinfrastruktur dar. Hier spielen Cloud-Lösungen in der Bereitstellung von zum Beispiel Testumgebungen und -werkzeugen ihre Stärken aus – auch für den Fall, dass ein Unternehmen die Testwerkzeuge sogar selbst besitzt. Dank der Cloud können weltweit Hunderte oder Tausende damit arbeiten und nicht nur eine Handvoll Menschen vor Ort. Auch lassen sich virtualisierte Testumgebungen grafisch konfigurieren und mit einem Mausklick aufsetzen. Rüst- und Totzeiten durch nicht oder falsch zur Verfügung stehende Testumgebungen lassen sich damit erheblich verringern.

Auch hier zahlt sich Industrialisierung in Form von Modularisierung und Standardisierung aus: Für Organisationen mit komplexen, interdependenten und stark individuellen IT-Landschaften ergeben sich naturgemäß geringere Vorteile als im Fall einfacher, modularer und aus Standardkomponenten bereitgestellter Umgebungen. Als Beispiel sei auch hier die Bereitstellung von Testumgebungen für das Testen mobiler Apps genannt. Durch die geringe Komplexität und Interdependenz solcher Apps ist es möglich, Testumgebungen mit hoher Effizienz aus einer private oder Public Cloud nach Bedarf zusammenzustellen.

Datenschutz und Sicherheit sind allerdings nicht nur im Licht der NSA-Affäre die am meisten diskutierten Fragen beim Cloud Computing. Insbesondere hier gilt: Kaum jemand macht es richtig. So erfordert eine Einbeziehung der Cloud in das Testvorgehen eine systematische Risikoeinschätzung sowohl im Hinblick auf den Datenschutz, als auch auf die IT-Sicherheit.

>



Mobile Testumgebungen aus der Cloud

Im Zweifel müssen Produktionsdaten – zum Beispiel, wenn sie personenbezogene Datensätze enthalten – erst aufwändig anonymisiert oder synthetisch erzeugt werden, bevor sie sich risikofrei in der Cloud verwenden lassen.

Angesichts all dieser potenziellen Stolpersteine und eventuell nötiger Vorabinvestitionen muss mit der Absicht, Teile des Testens in die Cloud zu verlagern, eine sehr genaue Kosten-Nutzen-Rechnung einhergehen. Diese muss auch für absehbare Reibungsverluste und Anlaufschwierigkeiten Kosten beziffern, wenn sie für eine realistische, nicht wolkige Einschätzung sorgen soll.

Drei Schritte in die Test-Cloud

Um eine solche Transparenz zu erhalten, empfiehlt sich vor dem Eintauchen in eine Test-Cloud ein Dreischritt aus einem "Cloud Readiness"-Assessment (Wo stehe ich?), Reifung (Was muss ich ändern?) und einem Implementierungsplan (Wie soll das Testen über die Cloud konkret aussehen?).

Das "Cloud Readiness"-Assessment" evaluiert systematisch, welche Teile der IT-Architektur in Bezug auf das Software-Testen in die Cloud ausgelagert werden können, und welche Reifegrade das Unternehmen beim Testen aufweist: Gibt es systematische Testprozesse? Sind diese organisationsweit ausgerollt? Oder folgt jedes Projekt seinem eigenen Vorgehen? Welche Testvorgehen und -infrastrukturen sind im Einsatz?

Die Fragen sollten sich bis hin zu den Testwerkzeugen erstrecken: Welche werden verwendet? Zu welchen Plattformen sind sie kompatibel? In einem solchen Frage- und Antwortspiel lassen sich alle relevanten Betrachtungsgegenstände und die Anforderungen an sie möglichst komplett abdecken. Die Liste der Betrachtungsgegenstände und der Anforderungen kann sich dabei auf Standards wie CMMI, SPICE für Projektorganisationen sowie auf ITIL für den IT-Betrieb stützen – je nachdem, welches Referenzmodell das jeweilige Unternehmen bereits im Einsatz hat.

Anhand der im Assessment festgestellten Erfüllungsgrade lässt sich darauf schließen, wie gut bestimmte Prozesse für das Cloud-Testen geeignet sind. Verfügt ein Unternehmen bereits über eine zentrale Testorganisation, kann diese nun definieren, welche Testaktivitäten am ehesten in Frage kommen. Existiert diese zentrale Stelle nicht, liegt es nahe, einzelne Testprojekte für die Cloud zu identifizieren. Insofern liefert ein "Cloud Readiness"-Assessment eine Art Marschroute für die einzelnen Schritte hin zur Cloud und zeigt auch notwendige Verbesserungsmaßnahmen auf, die erforderlich sind – und wie viel sie kosten.

Zugleich geben die Assessment-Ergebnisse bereits konkrete Hinweise für die Anbieterauswahl. Ein Beispiel: Zeigt die Untersuchung, dass ein Unternehmen bestimmte Werkzeugketten einsetzt, kommen Anbieter, die diese Technologien nicht unterstützen, schon

einmal nicht in Frage. Oder falls deutlich wird, dass aufgrund gesetzlicher Datenschutzaufgaben nur ein Datenhosting innerhalb der EU möglich ist, schränkt dies die möglichen kommenden Dienstleister ebenfalls ein.

Im zweiten Schritt geht es darum, die im Assessment identifizierten Lücken zu schließen, falls die betroffenen Prozesse in die Cloud transferiert werden sollen und der dafür notwendige Aufwand vertretbar ist. Diese Verbesserungsmaßnahmen finden aus Effizienzgründen zumeist in Zusammenarbeit mit dafür ausgewiesenen Spezialisten und Beratern statt. Sie stellen auf der organisatorisch-prozessualen, der testmethodischen sowie der technologischen Ebene die notwendige "Cloud Readiness" her.

Im dritten Schritt schließlich können die Verantwortlichen das System für das Cloud-Testen auf Basis des Assessments und der Verbesserungsmaßnahmen implementieren. Die zuvor systematisch definierten Anforderungen dienen nun sowohl als Vorgehensplan bei der Auswahl des Cloud-Anbieters als auch beim Aufsetzen der Testinfrastruktur.

Die Aufwände fallen dabei – je nach Testart – sehr unterschiedlich aus. Last- und Performancetests oder die Qualitätsprüfung von Software-Code liefern auch bisher schon zumeist als Managed Service. Durch diese bereits vorhandene Standardisierung ist eine Verlagerung in die Cloud schon mehr oder weniger vorgezeichnet.

◀(Sven Euteneuer)

[Der Autor]

Sven Euteneuer ist Global Head of Technical Quality bei der SQS Software Quality Systems AG.



[Rechtliche & Haftungsfragen bei Requirements-Engineering im agilen Umfeld]

von Johannes Bergsmann

In agilen Vorgehensweisen wird oft sehr locker mit ungenauen Requirements umgegangen mit dem Wissen, dass dies ja Grundgedanke der agilen Vorgehensweise ist. Doch wie sieht es in der Praxis mit verschiedenen rechtlichen Aspekten und Haftungsfragen aus, an die die Entwickler oder auch Kunden oft nicht denken? Erst im Schadensfall kommen dann die Probleme, die durch schlechtes Requirements-Engineering verursacht wurden, zu Tage. Dann ist es jedoch zu spät!

Im agilen Manifest steht: „Funktionierende Software wird höher gewertet als umfassende Dokumentation.“ In jedem agilen Methodenbuch steht, dass es besser ist, mehr zu kommunizieren, als alles im Detail niederzuschreiben. Im Idealfall eines Projekts kann dies voll unterstützt werden. Leider besteht das Leben nicht nur aus Idealfällen.

Solange alles gut und in den erwarteten Bahnen verläuft, wird kaum einer darauf drängen, dass alles aufgeschrieben werden muss, um sich abzusichern. Wenn dann aber die Sache schief läuft, dann kommt meist die Aussage: „Hätten wir das doch vorher schriftlich festgehalten oder vereinbart“.

Bei den Juristen gibt es folgendes Sprichwort: „Wer schreibt der bleibt und wer red't der geht!“. Dies kommt daher, dass in einem Streitfall - im schlimmsten Fall vor Gericht – alles **Niedergeschriebene eine um vieles höhere Beweiskraft hat, als eine mündliche Aussage**. Das ist natürlich ein konträrer Standpunkt zu den agilen Methoden und das andere Ende des Spektrums. In der Praxis kommen solche Fälle jedoch immer wieder vor, wie der Autor aus eigener Erfahrung als gerichtlich beeideter Sachverständiger berichten kann.

Ein weiteres rechtliches Problem zeigt sich bei der Vereinbarung von **Festpreisen in Projekten**. Der Kunde möchte verständlicherweise möglichst bald wissen, was sein Vorhaben kosten wird. Der viel zitierte Schätz-Unschärfetrichter ist dabei ein Faktum und kann zwar ignoriert, aber nicht eliminiert werden. Damit tritt in praktisch jedem Projekt, für das vor der Fertigstellung ein Festpreis vereinbart wird, das Problem auf, dass der geschätzte Preis und die Leistung nicht zusammen passen und bei größeren Abweichungen beginnen dann die Diskussionen, wer nun (drauf-)zahlt.

Es gibt mittlerweile auch verschiedene Literatur zum Thema „agiler Festpreis“¹. Darin wird jedoch nicht der Stein der Weisen beschrieben oder gar eine Lösung für den Unschärfetrichter gefunden, sondern meist alter Wein in neuen Schläuchen verkauft – also Methoden, die schon seit Jahrzehnten in klassischen Ausschreibungsverfahren verwendet werden (wie Bonusregelungen, Bandbreitenbegrenzungen, mehrstufige Vorgehensweisen, Risk-Sharing, etc.), werden in einem modernen agilen Kontext dargestellt. Das eigentliche Problem wird dabei aber nicht gelöst.

Ein wichtiger Punkt bezüglich Haftung ist auch das Thema **Normen und Standards**. Praktisch alle gängigen Normen im Umfeld der SW-Entwicklung verlangen gewisse Mindestdokumentationen und Spezifikationen für eine ordnungsgemäße Projektabwicklung.

Wir haben also die beiden Seiten „viel reden und weniger schreiben“ bis hin zu „möglichst alles aufschreiben, um abgesichert zu sein“.

Die nachfolgenden Abschnitte beleuchten die genannten Themen noch detaillierter.

¹ Es gibt in der Praxis nur genau einen *wirklichen* „agilen Festpreis“. Das ist dann möglich, wenn der Kunde einen festen Preis vereinbart und das Projekt dann auf jeden Fall genau mit der Erreichung dieser Kosten beendet wird, egal, was dann herausgekommen ist. Dies ist jedoch in der Praxis ein sehr unwahrscheinlicher Fall. Oft werden zusätzliche Iterationen angehängt und damit Mehrkosten in Kauf genommen.

Allgemeine rechtliche Aspekte aus Sicht des RE

Warum ist das Thema Requirements-Engineering aus rechtlicher Sicht überhaupt relevant? In der agilen Literatur (z.B. [Cohn, 2010] Seite 40 oder [Wirdemann, 2011] Seite 65) wird explizit erwähnt, dass User-Stories, die ja ebenfalls Requirements sind, kein Vertrag sind.

Dies ist nur bedingt richtig. Wenn nichts anderes vereinbart wurde, dann sind die schriftlich formulierten Anforderungen – auch wenn es sich „nur“ um ungenau formulierte User-Stories handelt – ein Bestandteil der Vereinbarung mit dem Kunden, sobald diese in den Backlog eingefügt werden und als zu realisieren gekennzeichnet werden, spätestens jedoch dann, wenn die Umsetzung dieser Requirements in einer beauftragten Iteration begonnen hat.

Es ist daher in jedem Fall sinnvoll, schon vor Projektbeginn eine schriftliche Vereinbarung mit dem Kunden zu treffen, welche Arten von Requirements vertraglich bindend sind und welche Requirements nur als Diskussionsgrundlage und nicht bindend angesehen werden.

An sich soll es nicht das Ziel sein, eine Spezifikation nur unter dem Aspekt zu schreiben, damit dann irgendwann vor Gericht gehen zu müssen. Es sollte jedoch vor allem bei externen Kunden-Lieferanten-Verhältnissen und jedenfalls bei Festpreisprojekten aus Gründen der Risikoabsicherung zumindest eine angemessene Struktur und Detailliertheit vorhanden sein, die das finanzielle und rechtliche Risiko in überschaubaren Grenzen hält.

Nun kann man natürlich auch den Standpunkt einnehmen: „Mir wird schon nichts passieren!“. Die Praxis zeigt jedoch, dass mehr als die Hälfte aller Software-Projekte (auch im agilen Umfeld) nicht so laufen wie geplant und ein Teil dieser Projekte auch komplett abgebrochen wird.

Es gibt natürlich auch viele „interne“ Situationen, bei denen das Entwicklungsteam und der interne Kunde im selben Unternehmen sind. Hier ist die Situation rechtlich meist nicht so kritisch. Und die Beteiligten, die in diesem Fall ja Arbeitskollegen sind, werden sich typischerweise nicht gegenseitig verklagen, wenn Requirements nicht korrekt umgesetzt wurden, sondern eine interne Lösung suchen. >

Requirements Engineering

Es kann jedoch auch hier eine rechtliche Relevanz geben, wenn es durch die beteiligten Personen zu fahrlässigen Handlungsweisen kommt und damit ev. eine persönliche Haftung einzelner im Schadensfall zum Tragen kommt. Fahrlässigkeit kann hier z.B. auch dadurch entstehen, dass durch unterlassene Spezifikation oder Analysen wichtige und risikoreiche Punkte übersehen wurden, die dann zu einem Schaden führen.

Vertragsbasis und Vertragserfüllungspflicht²

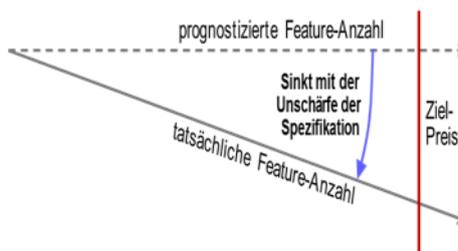
In den meisten extern beauftragten Projekten³ wird zwischen Kunde und Lieferant (zumindest grob) vereinbart werden, was bis wann zu liefern ist. Damit entsteht für den Lieferanten eine rechtliche Pflicht, diese Vereinbarung auch zu erfüllen. Je unklarer nun ein Auftrag bzw. die zugrundeliegende Spezifikation ist, desto unklarer ist natürlich auch die rechtliche Vertragsbasis.

Beispiele unklare Vertragsbasis:

Der Auftraggeber könnte z.B. als sehr unklare Ausgangsbasis im Extremfall formulieren: „Es ist ein neues Zeiterfassungssystem zu liefern.“

Hier wird jedem einleuchten, dass dies zu unklar formuliert ist. Aus Sicht der Vertragserfüllung ist dies natürlich sehr angenehm für den Lieferanten, weil dieser seine Verpflichtungen bereits damit erfüllt hat, wenn er *irgendein* neues Zeiterfassungssystem liefert, das grundsätzlich das macht, was ein Zeiterfassungssystem marktüblich kann. Der Kunde hat aufgrund der unklaren Anforderung keinen Anspruch darauf, ein bestimmtes System zu bekommen oder gar einzelne Funktionen, die ihm nicht gefallen, zu reklamieren.

Faktum ist, dass wir in einer Welt der begrenzten Ressourcen (Personen, Zeit, Geld) leben und dies auch für die agilen Vorgehensweisen gilt. Was oft nicht bedacht wird oder dem Auftraggeber verschwiegen wird ist, dass er bei Einhaltung eines definierten Zielpreises zwar ein lauffähiges Produkt bekommt, dass jedoch dann möglicherweise noch nicht alle von ihm gewünschten oder typischerweise zu erwartenden Features⁴ zu diesem Zeitpunkt realisiert wurden. Wenn der Auftraggeber nun auf seinen Feature-Wünschen bzw. Vorstellungen besteht, müssen zusätzliche Iterationen eingeschoben werden, was das Projekt nun auch wieder teurer macht. Diese Abweichung bis zur Erreichung des vereinbarten oder prognostizierten Zielpreises bzw. -Budgets wird umso größer, je ungenauer die Spezifikation am Beginn der übergreifenden Planung ist.



Besonders kritisch wird es, wenn im Rahmen der anfänglichen (groben) Projektvereinbarung auch KO-Kriterien vereinbart werden. Wenn diese Kriterien dann nicht bis zum Projektende erreicht werden, hat der Auftraggeber ev. das Recht⁵, das Projekt abzubrechen oder auch eine Rückabwicklung durchzuführen. Dies kann für einen kleinen Lieferanten möglicherweise sogar existenzbedrohend sein.

Nun sind KO-Kriterien in Projekten an sich nichts Ungewöhnliches. Problematisch wird dies beim agilen Vorgehen dadurch, dass hier bewusst nicht alle offenen Punkte vorab geklärt werden, weil man sich dann schrittweise an die Lösung annähert. Wenn nun im laufenden Projekt in einer Iteration eine technische Unklarheit auftaucht, so wird versucht, diese Unklarheit durch einen technischen Spike zu analysieren und eine Lösung dafür zu finden. Das Ergebnis der Analyse ist jedoch nicht vorhersehbar. Es kann auch sein, dass das Problem nicht lösbar ist. Wenn der Fall eintritt, dass man in einem Spike innerhalb einer Iteration nicht zu einem brauchbaren Ergebnis kommt, dann kann ev. noch ein weiterer Spike in der nächsten Iteration angehängt werden. Wenn dieser auch nicht zu einem Ergebnis kommt, ev. noch ein weiterer Spike, usw. Dadurch tritt eine Verschleppung der Klärung ein. Irgendwann, wenn dann klar ist, dass das Problem nicht oder nicht in einer akzeptablen Zeit oder mit vertretbarem Aufwand gelöst werden kann, wird der Auftraggeber das Projekt abbrechen⁶, da es sich um ein KO-Kriterium handelt und der zugrundeliegende Vertrag nicht mehr erfüllt werden kann.

Es kann daher nur dringend empfohlen werden, alle KO-Kriterien vor dem Beginn eines agilen Projekts zwischen Auftraggeber und Lieferant klar festzulegen und die Machbarkeit zu klären.

Kritisch sind auch Eigenschaften des Systems, die erst recht spät im Projekt oder überhaupt erst nach der vollständigen Umsetzung nachgewiesen werden können. Dies sind vorwiegend die nicht-funktionalen Eigenschaften wie z.B. Last-&Performance-Verhalten, Security, Usability, usw. >

² Wenn zwei Parteien einen Vertrag schließen, dann verpflichten sie sich gegenseitig, den Vertrag zu erfüllen. Durch die Vertragserfüllung erlischt dann das Schuldverhältnis gem. bürgerlichem Gesetzbuch. Dies bewirkt, dass beide Teile nicht mehr durch den Vertrag miteinander verbunden sind. Auf Vertragserfüllung kann auch geklagt werden. Auch wenn das Vertragsverhältnis erfüllt worden ist, sind die Partner trotzdem nicht rechtlos. Es knüpfen sich dann sogenannte Sekundärrechte an, beispielsweise die Gewährleistung.

³ Ausnahme von dieser Regel sind z.B. beauftragte Forschungsprojekte.

⁴ Die dem Kunden nach den gesetzlichen Regelungen zustehende „üblicherweise zu erwartende Beschaffenheit“ einer beauftragten Sache ist in vielen Projekten nachträglich ein großes Problem, da dies rechtlich relevant ist, jedoch hier meist für beide Vertragsseiten ein sehr großer Interpretationsspielraum offen bleibt.

⁵ Das Recht des Auftraggebers auf einen Projektabbruch oder Rückabwicklung hängt von den geltenden gesetzlichen und sonstigen getroffenen vertraglichen Vereinbarungen ab.

⁶ Hier zeigt sich dann, ob es wirklich ein KO-Kriterium war. Viele Auftraggeber stufen anfangs ein Kriterium als KO-Kriterium ein. Wenn dann der Projektabbruch, ein großer finanzieller oder zeitlicher Zusatzaufwand ev. auch für den Auftraggeber im Raum steht, dann wird aus dem KO-Kriterium oft ein „verzichtbares Kriterium“. Es ist daher ratsam, jedes KO-Kriterium schon am Anfang zwischen Auftraggeber und Lieferant unter dem Gesichtspunkt zu diskutieren und zu definieren, dass das Projekt bei Nichterfüllung wirklich abgebrochen wird!

Requirements Engineering

Besonders kritisch ist es dann, wenn nicht-funktionale Eigenschaften als KO-Kriterien vereinbart werden. Nicht funktionale Eigenschaften, die als KO-Kriterien vereinbart werden, müssen jedenfalls vor Projektstart detailliert (messbar) definiert und auf Machbarkeit geprüft werden!

Formulierungen wie „Die Software soll eine gute Usability haben.“, „Die Software muss ausreichend sicher sein.“ oder „Die Software muss eine hohe Performance aufweisen.“ sind zu vermeiden und führen in den meisten Fällen nachträglich zu langwierigen Diskussionen oder Streitereien, weil sowohl die Vertragsbasis als auch die typischerweise zu erwartende Beschaffenheit hier ziemlich offen sind.

Agile Spezifikation und Festpreis

Gerade bei Festpreisprojekten ist eine unklare Spezifikation ein sehr großes Problem, vor allem dann, wenn zusätzlich noch eine unkooperative Verhaltensweise des Lieferanten dazu kommt.

Bei Ausschreibungen wird meist vom Auftraggeber ein Festpreisangebot erwartet, das dann gemeinsam mit der ausgeschriebenen Spezifikation die Vertragsgrundlage für die Abwicklung des Projekts bildet.

Wenn die Spezifikation des Auftraggebers unklar formuliert ist, gewinnt bei Ausschreibungen typischerweise der „unkooperative“ Lieferant.

Warum dies so ist, zeigt der in untenstehender Abbildung dargestellte Unschärfe-Trichter:

Es ist Faktum, dass der Preis bei ungenauen Anforderungen auch mit einer gewissen Unschärfe kalkuliert werden muss.

Im Beispiel sei der erwartete Zielpreis 200.000 EUR.

Der **kooperative Lieferant** wird in diesem Fall auf einen geschätzten bzw. erwarteten Zielpreis einen Sicherheitsaufschlag dazugeben, da er ja damit rechnen muss, dass vom Auftraggeber wegen der zu diesem Zeitpunkt noch unklaren Anforderungen entsprechend viele Change-Requests geltend gemacht werden, von denen der Auftraggeber behaupten wird, dass diese aus seiner Sicht natürlich im Preis enthalten sein müssen. Der angebotene Preis dieses Lieferanten wird daher z.B. mit Sicherheitsaufschlag bei 290 T. Euro liegen, sodass er noch etwas Spielraum für Change-Requests hat, ohne gleich mit dem Auftraggeber wegen jeder Änderung den Preis neu verhandeln oder streiten zu müssen.

Der **unkooperative Lieferant**, der den Kunden bewusst „über den Tisch ziehen“ möchte, wird überlegen, dass er mit einem Sicherheitsaufschlag den Auftrag auf keinen Fall bekommen wird, da er dann im Vergleich für den Auftraggeber zu teuer ist. Er wird daher in diesem Fall jedenfalls einen niedrigeren Preis als den erwarteten Zielpreis angeben unter der Annahme einer möglichst minimalistischen Umsetzung des Systems und damit rechnen, dass vom Auftraggeber viele Change-Requests geltend gemacht werden, die er dann gegen ein entsprechend hohes Zusatzentgelt verrechnen kann und sich damit dann den gewährten Preisnachlass vom Auftraggeber im Laufe des Projekts wieder zurückholt. Der angebotene Preis dieses Lieferanten wird daher z.B. bei 160 T. Euro liegen.

Welcher Lieferant wird nun die Ausschreibung gewinnen?

Natürlich der Lieferant mit dem geringsten Preis, da ja beide Lieferanten den Preis auf Basis derselben (unklaren) Spezifikation abgegeben haben und damit scheinbar dieselbe Auftragsgrundlage besteht.

In der Praxis wird der Auftraggeber in diesem Fall bald merken, dass das Projekt zeitlich und preislich völlig aus dem Ruder läuft und im Endeffekt genau so viel oder mehr gekostet hat, wie beim Lieferanten, der von Anfang an realistisch und kooperativ geschätzt hat.

Der Auftraggeber muss nun als „Strafe“ für diese Fehlentscheidung zum erhöhten Preis auch noch Streitereien mit dem Lieferanten ertragen, hat dann meist auch wegen der zeitlicher Fehleinschätzungen einen unnötigen Projektstress und muss eventuell nachträgliche Budget-Erhöhungen gegenüber seinen Geldgebern und Entscheidungsträgern argumentieren. Im schlimmsten Fall kann es sogar zu einem Projektabbruch kommen.

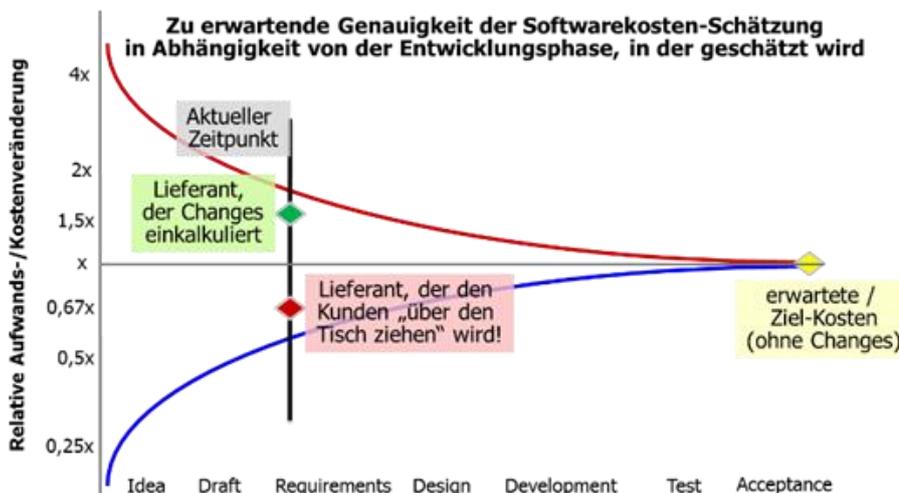
Wenn ein Festpreis vereinbart werden soll, kann daher nur dringend empfohlen werden, neben guten kaufmännischen und rechtlichen Rahmenbedingungen auch die inhaltliche Spezifikation im Vorfeld möglichst klar zu definieren, wobei eine Schätzgenauigkeit von < 20% angestrebt werden sollte.

Wenn man einen **Festpreis** für ein gesamtes Projekt **mit einer agilen Projektentwicklung** kombinieren möchte, sollten einige zusätzliche Aspekte und Hinweise beachtet werden die in [Bergsmann, 2014] nachgelesen werden können.

Öffentliche Ausschreibungen

Für öffentliche Ausschreibungen gelten meist recht enge rechtliche Vorgaben, die bei der Durchführung der Ausschreibung einzuhalten sind.

Ein Punkt, der dabei durch die ausschreibende Organisation einzuhalten ist, ist die **Neutralität und Vergleichbarkeit zwischen den Anbietern**. Es darf kein Anbieter bevorzugt werden! Wenn dies nicht gewahrt wird, kann die Ausschreibung eventuell von den Anbietern angefochten werden und dem Auftraggeber entstehen dadurch erhöhte Kosten und ev. auch ein großer Zeitverlust durch eine erneute Ausschreibung.



Requirements Engineering

Ein weiteres Thema ist die **Vergleichbarkeit** der Angebote. Bei agiler Projektvorgehensweise ist dies schwierig. Wenn nur ein grober Backlog bei der Ausschreibung vorliegt, ist hier keine objektive Vergleichbarkeit der Angebote möglich. Der Kunde kann nur die Firmen an sich vergleichen (Größe, Referenzen, Personen, Vertrauenswürdigkeit, etc.). Was er um den angebotenen Preis bekommt, ist jedoch nicht vergleichbar und kann in vielen Punkten zwischen den Anbietern sehr unterschiedlich sein.

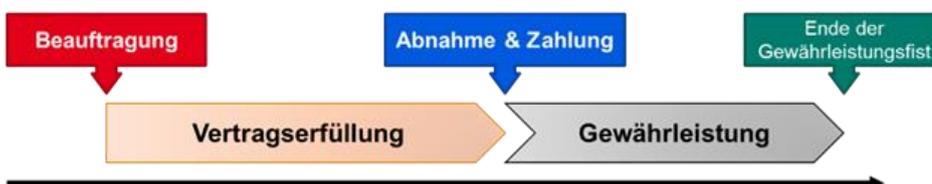
Natürlich könnte der Auftraggeber auch mit mehreren Lieferanten parallel 1-2 Iterationen als „Probeprojekt“ durchführen. Das ist jedoch aus Sicht der Neutralität und Vergleichbarkeit auch gefährlich, da man hier praktisch jedes Gespräch im Team detailliert mitprotokollieren muss und den anderen Lieferanten ebenfalls zur Verfügung stellen muss, da diese sonst behaupten können, dass nicht alle die gleichen Infos und Voraussetzungen hatten. Die meisten Firmen werden sich diesen Aufwand nicht leisten wollen.

Eine Festpreis-Ausschreibung ist daher aus den genannten Gründen bei agilen Projekten nicht sinnvoll möglich, wenn im Vorfeld keine detailliertere Spezifikation (klassisch oder agil) erstellt wird, auf deren Basis dann der Umsetzungsauftrag zum Festpreis vergeben wird und die dann die Basis für die Beurteilung der sicher kommenden Change-Requests sein kann.

Gewährleistung

Die **Gewährleistungsphase** beginnt nach der Vertragserfüllung, typischerweise nach der rechtlich relevanten Abnahme des Systems. In dieser Phase muss der Lieferant dann eventuelle Fehler im bereits abgenommenen System auf eigene Kosten so lange für den Kunden kostenlos nachbessern, bis das vereinbarte Kriterium umgesetzt wurde.

Siehe Abbildung unten.



Vertragserfüllung und Gewährleistung

Problematisch wird die Situation im agilen Bereich hier insofern, wenn durch (Zwischen-)Abnahmen bei jeder Iteration eine undurchschaubare Mischung aus Vertragserfüllung und Gewährleistungsansprüchen in einem Projekt entsteht.

Aus Sicht des Lieferanten wird eine regelmäßige Abnahme am Ende einer jeden Iteration anzustreben sein, da er dann jede Iteration gegenüber dem Kunden bezüglich Vertragserfüllung rechtlich abgesichert hat.

Für den Kunden ist eine solche Vorgehensweise fatal, weil dadurch die Leistungen einer Iteration bereits abgenommen sind und damit für diese Teile die Gewährleistung zu laufen beginnt. Der Lieferant ist jedoch schon in der nächsten Iteration bei der Implementierung und ändert hier möglicherweise wieder Teile des bereits abgenommenen Systems. Nun könnte es sein, dass daher für einen Teil des Systems Gewährleistung relevant ist und für einen anderen Teil die Vertragserfüllung. Dass so eine Situation im Schadensfall praktisch nicht mehr durchschaubar sein wird, ist sehr wahrscheinlich.



Problem der unterschiedlichen Gewährleistungsfristen bei agiler Entwicklung

Fehlersuche und Behebung verursachen auch zusätzliche (ev. vorab nicht eingeplante) Aufwände, sodass z.B. dann für den Kunden aufgrund der notwendigen Fehlerbehebung weniger Features als vorgesehen in der Iteration realisiert werden können.

In den Verträgen sollte daher unbedingt auch eine passende Formulierung für die Wirksamkeit und den Umgang mit (Zwischen-)Abnahmen und die Fehlerbehebung in nachfolgenden Iterationen enthalten sein.

Haftung des Teams

In agilen Projekten liegt die Verantwortung für die Umsetzung ja beim Team. Es stellt sich nun die Frage: **Wer ist für was verantwortlich im agilen Projekt, wenn es zu einem Schaden kommt?** Es ergeben sich im agilen Umfeld - unabhängig davon, ob nun eine gute Basis für agile Projekte vorhanden ist oder nicht - für die Haftungsfrage natürlich andere Voraussetzungen und Gesichtspunkte.

Einige Fragen in diesem Zusammenhang sind:

- Wer ist für was verantwortlich im agilen Projekt, wenn es zu einem Schaden kommt?
- Was passiert, wenn ein unerfahrener Programmierer fahrlässig einen Fehler einbaut und dadurch ein Millioenschaden oder Personenschaden entsteht?

- Haften nun alle Team-Mitglieder solidarisch für den Schaden? Wer im Team zahlt wie viel?
- In welchem Umfang haftet der Auftraggeber oder der Product-Owner selbst?
- Entsteht Fahrlässigkeit, wenn die (nachweisliche) Kommunikation zu gering ist, also wenn z.B. nicht aufgeschrieben wird, was in den Meetings besprochen wird?
- Inwieweit haftet der CIO des Auftraggebers als Projektverantwortlicher des Auftraggebers?

>

Requirements Engineering

Per Definition ist im agilen Team das Team für alles verantwortlich und der Code gehört dem gesamten Team.

Was passiert, wenn ein unerfahrener Programmierer fahrlässig einen Fehler einbaut und dadurch ein Millionschaden oder Personenschaden entsteht?

Bei Fahrlässigkeit wird den Auftraggeber und möglicherweise auch den dafür verantwortlichen Entwickler ein Haftungsanspruch treffen. **Haften nun alle Team-Mitglieder solidarisch für den Schaden des einen Entwicklers? Wer im Team zahlt wie viel?**

Der Nachweis, wer den fehlerhaften Code verursacht hat, kann heute bei Verwendung eines Code-Versionsmanagement-Systems und einer guten Zugriffsrechteverwaltung recht einfach erbracht werden.

Die Frage ist dann, ob das Team wirklich solidarisch handelt und den Schaden als Team übernimmt. Aus der Erfahrung zeigt sich, wenn es um finanzielle Schäden geht, die solidarisch zu tragen sind, dann hört sich meist jede Solidarität und Freundschaft auf und es wird um jeden Euro gestritten.

Es kann daher nur empfohlen werden, vorab in einem agilen Projekt schriftlich zu definieren, für welche Fälle überhaupt eine Haftung übernommen wird und ob bzw. in welchen Fällen das Team oder den Einzelnen ein Haftungsanspruch trifft.

Standards & Normen

In den 10-15 Jahren hat sich die SW-Branche stärker hin zu einer qualitätsorientierten und „industriellen“ Vorgehensweise bewegt. Dies wird in zahllosen Normen und Standards sichtbar, die es mittlerweile auch im SW-Bereich gibt. So strukturiert und weit wie in anderen Branchen ist man zwar in der Software-Branche noch nicht, aber die Tendenz ist klar absehbar. Daher kann in diesem Zusammenhang auch argumentiert werden, dass von sorgfältig agierenden Vertragspartnern und SW-Experten vorausgesetzt werden kann, dass sie die gängigen Standards zur Software-Entwicklung sowohl allgemein als auch in Ihrem Branchenumfeld kennen und auch anwenden.

Insofern trifft hier sowohl den Lieferanten, als auch den Auftraggeber (im Speziellen dann, wenn er einen Experten wie einen CIO oder einen Berater beschäftigt) eine Verantwortung, für eine ordnungsgemäße Projektvorgehensweise zu sorgen.

Sofern nichts anderes definiert wurde, kann davon ausgegangen werden, dass der Stand der Technik eingehalten wird. Insofern ist es sehr wichtig, die relevanten Normen zu kennen und ggf. in Projekten die Anwendung von Standards explizit anzugeben oder auch auszuschließen (in diesem Fall ist eine entsprechende Begründung anzugeben).

Wenn der Lieferant als Experte und/oder der Verantwortliche des Auftraggebers ihre Verantwortung trotz besserem Wissen, das sie in ihren Rollen haben sollten, nicht ordentlich wahrnehmen, kann dies jedenfalls als fahrlässig angesehen werden und damit durchaus auch zu persönlichen Haftungsansprüchen führen.

Leider herrscht oft noch die Meinung vor, dass verschiedene Agile Vorgehensmodelle wie z.B. SCRUM ausreichend sind für eine ordnungsgemäße SW-Entwicklung.

Tatsache ist, dass SCRUM, KANBAN, BDD, TDD, etc. allesamt einzelne Themenbereiche und Bausteine eines guten Software-Entwicklungsprozesses abdecken, jedoch keine dieser Methoden alle notwendigen Bereiche für eine ordnungsgemäße Software-Entwicklung abdeckt.

Sich auf den Einsatz EINER etablierten Vorgehensweise wie SCRUM zu berufen, ist daher nicht ausreichend, um seine Verantwortung zu erfüllen.

Wenn man nur das Beispiel SCRUM betrachtet, so fehlen hier Definitionen für viele verschiedene wichtige Themenbereiche wie z.B. Requirements-Engineering, Risikomanagement, Testvorgehen, etc., die allesamt essentiell wichtig sind, damit ein Entwicklungsprojekt erfolgreich wird. Damit hat man in einem Streitfall natürlich das Problem, dass man nicht einfach sagen kann „Wir haben ja eine agile Vorgehensweise vereinbart – dort wird eben weniger dokumentiert und mehr geredet“. Diese Aussage wird vor Gericht kaum Gewicht haben.

Auch wenn wenig explizit vorgegeben oder dokumentiert wird, so muss die Vorgehensweise in den von den agilen Methoden nicht definierten Bereichen und das „weniger Dokumentierte“ trotzdem den gängigen Standards und Normen entsprechen, damit hier keine fahrlässige Handlungsweise vorliegt.

Weitere Infos zu diesem Thema sind im neuen Buch „Requirements Engineering für die Agile Softwareentwicklung“ enthalten, das ab ca. März 2014 im dpunkt.verlag erscheint.



Literatur:

[Bergsmann, 2014]

Johannes Bergsmann, Requirements Engineering für die Agile Softwareentwicklung, dpunkt.verlag, 2014

[Cohn, 2010]

Mike Cohn, *User Stories.*, mitp, 2010

[Wirdemann, 2011]

Wirdemann, *SCRUM mit User Stories.* Carl Hanser Verlag, 2011

◀ (Johannes Bergsmann)

[Der Autor]

Johannes Bergsmann



- Eigentümer von Software Quality Lab
- Vizepräsident der Österr. Vereinigung für SW-Qualitätsmanagement
- Vizepräsident des Austrian Testing Boards
- Studium technische Informatik
- über 25 Jahre Erfahrung in der SW-Entwicklung
- Vortragender auf verschiedenen Konferenzen
- gerichtlich beeideter Sachverständiger für Informatik
- Ziviltechniker für Informatik

Der Autor sieht sich nicht als GURU oder Dogmatiker, sondern als Praktiker und Pragmatiker, der viele verschiedene Ansätze kennt und in der Praxis erlebt hat und die vielen verschiedenen Vorgehensweisen als Werkzeugkasten sieht, der je nach Situation zielgerichtet eingesetzt werden sollte.

Umfrage / Awards

[Umfrage zum Testen in ERP-Projekten]

Von Michael Felderer

Das Institut für Informatik und das Institut für Strategisches Management der Universität Innsbruck führen gerade eine umfassende Umfrage zum Testen in ERP-Projekten durch.

Die Umfrage richtet sich an Personen im deutschsprachigen Raum, die in ERP-Projekten testen bzw. das Testen in ERP-Projekten managen. Falls Sie bereits in ERP-Projekten getestet haben, würde es uns sehr freuen, wenn Sie an der Umfrage teilnehmen könnten. Weiters würden sich die Umfrageersteller sehr freuen, wenn Sie den Link an Ihnen bekannte Personen verteilen könnten, welche in das Testen von ERP-Projekten involviert sind oder waren.

Die Umfrage ist online unter <http://erp-testing.com> verfügbar.

Neben der wissenschaftlichen Aufarbeitung der Ergebnisse, werden wir auf Basis der Umfrage Guidelines und Best Practices zum Testen in ERP-Projekten zusammenstellen und den Teilnehmern an der Umfrage zur Verfügung stellen.

Für weitere Auskünfte zur Umfrage steht Ihnen Dr. Michael Felderer (michael.felderer@uibk.ac.at) jederzeit gerne zur Verfügung.

◀(Michael Felderer)

[Best Paper Award auf IWSM-MENSURA 2013]

Von Martina Höller

Rudolf Ramler und Johannes Himmelbauer wurden für ihren Beitrag "Noise in Bug Report Data and the Impact on Defect Prediction Results" mit dem Best Paper Award der Konferenz IWSM-MENSURA 2013 ausgezeichnet.

Das Paper analysiert, wie sich das "Rauschen" in Mess- und Fehlerdaten auf die Vorhersage fehlerhafter Softwaresysteme auswirkt. Zur Fehlervorhersage werden häufig Metrik- und Fehlerdaten aus früheren Versionen herangezogen. Daten aus Industrieprojekten sind jedoch häufig durch Messfehler oder ungenaue Zuordnungen verfälscht. Anhand eines realen Projekts untersucht das Paper verschiedene Ursachen für Datenfehler und zeigt, dass mit dem verwendeten Modell auch bei 20% falschen Daten noch nutzbare Vorhersagen möglich sind.

Die Arbeit wurde bei der Joint Conference of the 23rd International Workshop on Software Measurement (IWSM) and the Eighth International Conference on Software Process and Product Measurement (Mensura) vorgestellt, die im Oktober in Ankara stattfand.

◀(Martina Höller)

[Wir sind ISTQB Award 2013!]

Von Helmut Pichler



Seit 2011 verleiht das ISTQB einen Award für besondere Leistungen im Bereich Software Testing, der von einem international zusammengesetzten Komitee verliehen wird.

Nach Boris Beizer, 2011 und Dorothy Graham 2012 wurde heuer 2013 "unserem" Harry Sneed dieser Award verliehen.

Harry hat mittlerweile mehr als ein Dutzend Bücher und über hundert Fachartikel geschrieben - und hat neben seinen Projekteinsätzen einen Lehrstuhl für Software Testing an Universitäten in Österreich, Deutschland und Ungarn.

Kurz: Harry reiht sich zurecht in die Galerie der Software-Test-Technologie-Pioniere ein.

Herzlichen Glückwunsch!

◀(Helmut Pichler)

[Der Autor]

Dr. Michael Felderer ist Assistent am Institut für Informatik der Universität Innsbruck. Zu seinen Arbeitsschwerpunkten zählen Testen von Software und Requirements Engineering. Neben seiner wissenschaftlichen Tätigkeit ist Dr. Felderer auch als Berater in den Bereichen Software-Qualität und Anforderungsmanagement für die QE LaB Business Services GmbH tätig.



Harry Sneed erhält den Award von Chris Carter, dem Präsidenten des ISTQB



3. ATB Charity Punsch: Motto „Vernetzen, Spaß haben und HEL- FEN“



Montag 16.12.2013 fand der diesjährige ATB Charity Punsch statt.

In einem tollen, vom Unibräu bereitgestellten Bereich konnten wir über 40 Gäste begrüßen, die wieder leckeren Punsch & Glühwein genossen haben und sich auch sehr großzügig bei den Spenden zeigten.

Alleine die großzügigen Firmenspenden (EUR 2000,-) bilden einen beachtlichen Grundstock unserer Erlöse. Zusammen mit den Spenden beim Punschevent selbst, erzielten wir einen beeindruckenden Gesamterlös von EUR 2300.-, der vom ATB wieder verdoppelt wird.

Wir sind nun bei der Auswahl der konkreten Kinderträume - darüber berichten wir im nächsten ATB Insider.

Wesentlich am Erfolg unseres Events trugen neben den Besuchern, wie erwähnt, folgende Firmen bei:

Wir bedanken uns im Namen des ATB und stellvertretend für "Stiftung Kindertraum" und den Kindern:



Und hier noch ein paar Fotos von diesem Event:



[ISTQB Meeting in Lissabon]

Von Helmut Pichler

Das letzte ISTQB GA Meeting dieses Jahres fand in der Woche um den 29. Nov 2013 in Lissabon statt.

Rund 80 Teilnehmer, davon 29 Board-Vertreter, darunter Helmut Pichler vom Austrian Testing Board, nahmen daran teil.

Bereits Anfang der Woche starteten die Arbeitsgruppen, wo aktiv an der Ausarbeitung der neuen AddOns Agile Testing, Model Based Testing (Anm.: wo auch wir vom ATB aktiv mitarbeiten und gestalten), an den neuen Prozessen und an der Überarbeitung des Foundation Level Lehrplanes, um nur einige zu nennen, gearbeitet wurde.

Der offizielle Teil startet immer mit einem Round Table Meeting am Vortag des eigentlichen GA, wo konkret z.B. das Thema verpflichtende Trainingsdauer besprochen wurde. Hintergrund: seit Kurzem bietet ein Unternehmen FL-Trainings in 2 Tagen an; dies wäre theoretisch (noch) möglich, praktisch jedoch aufgrund der gesetzlichen Maximalarbeitsdauer von 10 Stunden unmöglich. Hier wird nun an Richtlinien gearbeitet.

Im GA Meeting selbst wurde vom erfreulichen Durchbruch der 300.000er Schallgrenze der Zertifikatsträger berichtet, Österreich liegt hier erfreulicherweise im Ranking der FL:AL-Zertifikate weltweit an sensationeller 3.Stelle. Weiters wurden die Weichen für die Überarbeitung des FL Lehrplanes gelegt und auf die mittlerweile sehr umfangreiche ISTQB Website (www.istqb.org) hingewiesen, die im Schnitt von > 3000 Besuchern pro Tag besucht wird. Daher werden die vom ISTQB angebotenen Partnerschaften für Firmen sehr interessant. Neben vielen anderen Punkten ist noch erwähnenswert, dass es nun ein weiteres Testing Board in Kenia gibt.



Helmut Pichler mit Freunden vom schwedischen Board

◀ (Helmut Pichler)

[Krampus am Campus]

Am 3.12.2013 fand die Krampusfeier der ANECON statt, zu der die Kunden geladen wurden. Unter anderem war auch der Redakteur des Insiders da.

Hier ein paar Eindrücke von dieser Veranstaltung.



◀ (Karl Kemminger)

ANECON schenkt allen ATB-Mitgliedern einen Wandkalender fürs neue Jahr! Schicken Sie einfach Ihre Kontaktdaten an marketing@anecon.com.



Teilnehmer am ISTQB GA Meeting

[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.

[Besucherrekord bei Wiener Fachgruppenabend für Softwaretest]

Motto „Let’s have a party“ lockt Studenten an.

Der letzte Fachgruppenabend des Arbeitskreises Software-Qualität und -Fortbildung e.V. (ASQF) war mit mehr als 80 Teilnehmern ein voller Erfolg. Fachleute der Testszene und zahlreiche Studenten der Fachhochschule Technikum Wien kamen zum Fachvortrag. Unter dem Motto „Und täglich grüßt der Systemtest“ führte Rudolf Grötz gemeinsam mit seinem Kollegen in die Praxis zum Thema ein.

Der ASQF ist das Kompetenznetzwerk der Softwareentwicklungs-Branche im deutschsprachigen Raum. Die über 1.100 im ASQF engagierten Global-Player-Unternehmen, leistungsstarken Mittelständler, Fachleute, Hochschulen und Forschungseinrichtungen vereint der gemeinsame Einsatz für Qualitätsstandards in der Informations- und Kommunikationstechnologie. Die ASQF-Fachgruppe Software Test Österreich wird von Dr. Mohsen Eksir, Head of Software Test & Quality Assurance bei der BDC, geleitet. „Es freut mich sehr, dass der ASQF Praxis und Lehre miteinander vereint, und das Thema Software Test und Qualitätssicherung bereits bei Studenten etabliert.“



◀ (Stefanie Höring)

Fachgruppenabend Softwaretest Österreich: Di, 4. März 2014, 18:00 bis 20:00

Fachhochschule Technikum Wien, HS_A1.04A, 1200 Wien, Höchstädtplatz 5

Thema:

Anwendbarkeit des Prozessbewertungs-Schemas TPI Next auf Security Test-Organisationen

Abstract:

Das Vertrauen in ein Testergebnis hängt stark davon ab, ob man den durchgeführten Test als „gut“ erachtet. Um die Qualität von Testprozessen einer Organisation oder eines Testteams formal und möglichst neutral bewerten zu können, hat sich das TPI-Next Modell (Test Process Improvement—Next) bewährt. Im Vortrag wird untersucht, wie gut sich TPI-Next für die Bewertung der Qualität von Organisationen oder Testteams eignet, die hauptsächlich auf dem Gebiet des Security Testings arbeiten.

Maria-Therese Teichmann:

Maria-Therese Teichmann ist seit ihrem Studienabschluss der Quantenoptik in der IT tätig. Als Senior Quality Manager im Wertpapierbereich bei Software Daten Service GmbH ist sie seit vielen Jahren mit dem Optimieren und Bewerten von Prozessqualität in Software-Entwicklung und -Wartung vertraut. Darüber hinaus ist sie als Lektorin an der FH Technikum Wien und als Managing Direktor von Certible mit Aus- und Weiterbildung im IT-Bereich tätig.



[Link zur Anmeldung](#)

[ATB Expertentreff - Wie sind die Testergebnisse?]

von Karl Kemminger



Am 8.11.2013 widmete sich der ATB Expertentreff der einfachen (?) Fragestellung „Wie sind die Testergebnisse?“. Es ging also um Reporting, Information der Stakeholder und die Frage, welche Metriken sinnvoll sind.

Zu Beginn gab es wieder die übliche Abstimmung:

- Setzen Sie im Rahmen Ihrer Berichte auf internationale Standards, z.B. IEEE 829? -> 3 Ja, 11 Teilweise, 13 Nein
- Bedienen Sie sich bei Reports Testmanagement Software Tools? -> 13Ja, 12 Teilweise, 3 Nein
- Überlegung: XLS gibt es nicht -> ein Problem? -> 2 Definitiv, 18 Naja, 4 Nein
- Besprechen Sie Ihre Reports mit den Stakeholdern? -> 17 Ja, 9 Teilweise, 1 Nein

Nach einem Vortrag von Dr. Gerald Zincke/SQS mit vielen praktischen Tipps (siehe Folien) gab es wieder eine rege Diskussion, wobei auch Dr. Robert Rippel von Axiatus moderierte. Hier ein paar Auszüge:

Es wurde das Thema des Stellenwerts des Tests diskutiert, dabei fielen Aussagen wie „Der Tester ist das Äquivalent des Straßenkehrers, der den Müll wegräumt“ und ähnliches. Zum Nachdenken regte R. Rippel an, als er fragte, wie viele erlebt haben, dass seitens des Auftraggebers Weglassen oder Kürzen der Implementierung gefordert wurde (niemand), und wie viele dasselbe beim Test (doch einige).

Soll man, wenn Zeitmangel herrscht, den Report weglassen, und die Zeit in Tests stecken? Klares Nein, zumindest ein Minimalreport sollte unbedingt erstellt werden. Z.B. gilt in der Pharmabranche ein Produkt nur dann als getestet, wenn der Test auch dokumentiert ist. Auch in anderen Branchen gibt es Dokumentationspflichten.

Reports sollten natürlich soweit wie möglich automatisiert werden, allerdings müssen die Zahlen und Grafiken meistens noch textuell ergänzt und interpretiert werden (z.B. Erklärungen für Brüche im Verlauf durch z.B. Krankheiten oder verstärkten Einsatz weiterer Tester), erst dann wird der Bericht oft verständlich und interessant.

Auch Entwicklertests sollten dokumentiert werden, allerdings geschieht dies in der Praxis nur selten.

Projektmanager sind im Allgemeinen Optimisten und nehmen daher Testberichte, die auf Probleme und Verzug hinweisen, nicht ernst.

Abschließend das Statement, dass der Testreport (und der gesamte Test an sich) vermarktet werden muss, z.B. durch interessante Aufmachung. Es ist wichtig, dass vom Management die Bedeutung des Tests erkannt wird.

Alle Unterlagen finden Sie demnächst unter http://www.austriantestingboard.at/expertentreff_rueckblicke



Helmut Pichler, Robert Rippel, Alexander Weichselberger, Gerald Zincke



Das Publikum wurde wieder bestens unterhalten.



Nach den Vorträgen und Diskussionen gab es viel Gelegenheit für Networking, ...



... wie immer mit einem ausgezeichneten Buffet.



Kurs	Termin	Ort	Anbieter
ISTQB Certified Tester Foundation Level	20.01. - 23.01.2014, Anmeldung	Graz	BDC
	20.-23.01.2014, Anmeldung	Wien	Software Quality Lab
	20.-23.01.2014, Anmeldung	Linz	
	20.-23.01.2014, Anmeldung	Graz	
	21.-24.01.2014, Anmeldung	Lustenau	
	03.02. - 06.02.2014, Anmeldung	Wien	SQS
	11.02. - 14.02.2014, Anmeldung	St. Pölten	BDC
	17.02. - 20.02.2014, Anmeldung	Wien	ANECON
	24.02. - 27.02.2014, Anmeldung	Wien	BDC
	24.-27.02.2014, Anmeldung	Wien	Software Quality Lab
	24.-27.02.2014, Anmeldung	Linz	
	24.-27.02.2014, Anmeldung	Graz	
	25.-28.02.2014, Anmeldung	Lustenau	
	10.03. - 13.03.2014, Anmeldung	Graz	BDC
	07.04. - 10.04.2014, Anmeldung	Wien	BDC
	07.04. - 10.04.2014, Anmeldung	Wien	SQS
	07.-10.04.2014, Anmeldung	Wien	Software Quality Lab
	07.-10.04.2014, Anmeldung	Linz	
	07.-10.04.2014, Anmeldung	Graz	
	07.-10.04.2014, Anmeldung	Lustenau	
	12.05. - 15.05.2014, Anmeldung	Salzburg	BDC
	12.-15.05.2014, Anmeldung	Wien	Software Quality Lab
	12.-15.05.2014, Anmeldung	Linz	
	12.-15.05.2014, Anmeldung	Graz	
	12.-15.05.2014, Anmeldung	Lustenau	
	02.06. - 05.06.2014, Anmeldung	Wien	BDC
	02.06. - 05.06.2014, Anmeldung	Wien	SQS
	23.-26.06.2014, Anmeldung	Wien	Software Quality Lab
	23.-26.06.2014, Anmeldung	Linz	
	23.-26.06.2014, Anmeldung	Graz	
	23.-26.06.2014, Anmeldung	Lustenau	
	23.06. - 26.06.2014, Anmeldung	Graz	BDC
ISTQB Certified Tester Foundation Level englischsprachig	20.-23.01.2014, Anmeldung	Wien	Software Quality Lab
	02.-05.06.2014, Anmeldung	Wien	
	02.-05.06.2014, Anmeldung	Linz	
	02.-05.06.2014, Anmeldung	Graz	
	02.-05.06.2014, Anmeldung	Lustenau	



Kurs	Termin	Ort	Anbieter
ISTQB Certified Tester Advanced Level Test Manager	03.-07.03.2014, Anmeldung	Wien	Software Quality Lab
	03.-07.03.2014, Anmeldung	Linz	
	03.-07.03.2014, Anmeldung	Graz	
	10.-14.03.2014, Anmeldung	Lustenau	
	10.03. - 14.03.2014, Anmeldung	Wien	SQS
	07.-11.07.2014, Anmeldung	Wien	Software Quality Lab
	07.-11.07.2014, Anmeldung	Linz	
	07.-11.07.2014, Anmeldung	Graz	
	07.-11.07.2014, Anmeldung	Lustenau	
ISTQB Certified Tester Advanced Level Test Analyst	24.02. - 27.02.2014, Anmeldung	Wien	SQS
	10.-13.03.2014, Anmeldung	Wien	Software Quality Lab
	10.-13.03.2014, Anmeldung	Linz	
	10.-13.03.2014, Anmeldung	Graz	
	17.-20.03.2014, Anmeldung	Lustenau	
	10.-13.06.2014, Anmeldung	Wien	Software Quality Lab
	10.-13.06.2014, Anmeldung	Linz	
	10.-13.06.2014, Anmeldung	Graz	
	10.-13.06.2014, Anmeldung	Lustenau	
ISTQB Certified Tester Advanced Level Technical Test Analyst	18.-20.03.2014, Anmeldung	Wien	Software Quality Lab
	18.-20.03.2014, Anmeldung	Linz	
	18.-20.03.2014, Anmeldung	Graz	
	18.-20.03.2014, Anmeldung	Lustenau	
	24.03. - 26.03.2014, Anmeldung	Wien	SQS
	26.-28.05.2014, Anmeldung	Wien	Software Quality Lab
	26.-28.05.2014, Anmeldung	Linz	
	26.-28.05.2014, Anmeldung	Graz	
	26.-28.05.2014, Anmeldung	Lustenau	
Certified Agile Tester® Training (Prüfung in deutsch oder englisch)	20.01. - 24.01.2014, Anmeldung	Wien	ANECON
	03.-07.03.2014, Anmeldung	Wien	Software Quality Lab
	03.-07.03.2014, Anmeldung	Linz	
	03.-07.03.2014, Anmeldung	Graz	
	03.-07.03.2014, Anmeldung	Lustenau	
	05.05. - 09.05.2014, Anmeldung	Wien	SQS
	12.-16.05.2014, Anmeldung	Wien	Software Quality Lab
	12.-16.05.2014, Anmeldung	Linz	
	12.-16.05.2014, Anmeldung	Graz	
12.-16.05.2014, Anmeldung	Lustenau		



Kurs	Termin	Ort	Anbieter
IREB Certified Professional for Requirements Engineering Foundation Level	27.-29.01.2014, Anmeldung	Wien	Software Quality Lab
	27.-29.01.2014, Anmeldung	Linz	
	27.-29.01.2014, Anmeldung	Graz	
	28.-30.01.2014, Anmeldung	Lustenau	
	11.-13.02.2014, Anmeldung	Wien	
	11.-13.02.2014, Anmeldung	Linz	
	11.-13.02.2014, Anmeldung	Graz	
	11.-13.02.2014, Anmeldung	Lustenau	
	01.-03.04.2014, Anmeldung	Wien	
	01.-03.04.2014, Anmeldung	Linz	
	01.-03.04.2014, Anmeldung	Graz	
	01.-03.04.2014, Anmeldung	Lustenau	
	20.-22.05.2014, Anmeldung	Wien	
	20.-22.05.2014, Anmeldung	Linz	
	20.-22.05.2014, Anmeldung	Graz	
	20.-22.05.2014, Anmeldung	Lustenau	
	15.-17.07.2014, Anmeldung	Wien	
	15.-17.07.2014, Anmeldung	Linz	
15.-17.07.2014, Anmeldung	Graz		
15.-17.07.2014, Anmeldung	Lustenau		

[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.



Veranstaltung	Termin	Ort	Anbieter
Tosca Certified User Foundation Level (TCUFL)	14.01.-16.01.2014, Anmeldung	Wien	Tricentis
	04.03.-06.03.2014, Anmeldung	Wien	
	15.04.-17.04.2014, Anmeldung	Wien	
Tosca Certified Quality Designer (TCQD)	21.01.-23.01.2014, Anmeldung	Wien	Tricentis
	18.03.-20.03.2014, Anmeldung	Wien	
Tosca Certified Administrator (TCA)	28.01.-29.01.2014, Anmeldung	Wien	Tricentis
Tosca Certified User Advanced Level (TCUAL)	11.03.-13.03.2014, Anmeldung	Wien	Tricentis
Tosca Technical Training (TTT)	25.03.-27.03.2014, Anmeldung	Wien	Tricentis
Tosca Reporting Training (TRT)	08.04.-09.04.2014, Anmeldung	Wien	Tricentis
360° Testautomatisierung	26.02. - 27.02.2014, Anmeldung	Wien	ANECON
Funktionale Sicherheit	04.-05.02.2014, Anmeldung	Linz	Software Quality Lab
	04.-05.02.2014, Anmeldung	Graz	
	04.-05.02.2014, Anmeldung	Lustenau	
	18.-19.02.2014, Anmeldung	Wien	
	23.-24.04.2014, Anmeldung	Wien	
	23.-24.04.2014, Anmeldung	Linz	
	23.-24.04.2014, Anmeldung	Graz	
	23.-24.04.2014, Anmeldung	Lustenau	
Scrum Master Professional	11.-12.02.2014, Anmeldung	Wien	Software Quality Lab
	11.-12.02.2014, Anmeldung	Linz	
	11.-12.02.2014, Anmeldung	Lustenau	
	19.-20.02.2014, Anmeldung	Graz	
	15.-16.04.2014, Anmeldung	Wien	
	15.-16.04.2014, Anmeldung	Linz	
	15.-16.04.2014, Anmeldung	Graz	
	15.-16.04.2014, Anmeldung	Lustenau	
Unit Testing (Testgetriebene Softwareentwicklung)	04.-06.03.2014, Anmeldung	Wien	Software Quality Lab
	04.-06.03.2014, Anmeldung	Linz	
	04.-06.03.2014, Anmeldung	Graz	
	04.-06.03.2014, Anmeldung	Lustenau	
Moderationstechniken im Requirements Engineering	05.-06.03.2014, Anmeldung	Wien	Software Quality Lab
	05.-06.03.2014, Anmeldung	Linz	
	05.-06.03.2014, Anmeldung	Graz	
	05.-06.03.2014, Anmeldung	Lustenau	



Veranstaltung	Termin	Ort	Anbieter
Software Usability - FL	11.–13.03.2014, Anmeldung	Wien	Software Quality Lab
	11.–13.03.2014, Anmeldung	Linz	
	11.–13.03.2014, Anmeldung	Graz	
	11.–13.03.2014, Anmeldung	Lustenau	
Professionelles Requirements Engineering und Management	18.–20.03.2014, Anmeldung	Wien	Software Quality Lab
	18.–20.03.2014, Anmeldung	Linz	
	18.–20.03.2014, Anmeldung	Graz	
	18.–20.03.2014, Anmeldung	Lustenau	
Aufwandsschätzung in Softwareprojekten	26.–27.03.2014, Anmeldung	Wien	Software Quality Lab
	26.–27.03.2014, Anmeldung	Linz	
	26.–27.03.2014, Anmeldung	Graz	
	26.–27.03.2014, Anmeldung	Lustenau	
Software Testen für Manager	27.03. - 28.03.2014, Anmeldung	Wien	BDC
ISAQB® Certified Professional for Software Architecture - FL	31.03.–03.04.2014, Anmeldung	Wien	Software Quality Lab
	31.03.–03.04.2014, Anmeldung	Linz	
	31.03.–03.04.2014, Anmeldung	Graz	
	31.03.–03.04.2014, Anmeldung	Lustenau	
Software Test für Embedded Systems	10.04. - 11.04.2014, Anmeldung	Wien	BDC
GUI-Testautomatisierung in Theorie und Praxis	09.–10.04.2014, Anmeldung	Wien	Software Quality Lab
	09.–10.04.2014, Anmeldung	Linz	
	09.–10.04.2014, Anmeldung	Graz	
	09.–10.04.2014, Anmeldung	Lustenau	
UML Basics für Fachbereichs-Mitarbeiter	29.–30.04.2014, Anmeldung	Wien	Software Quality Lab
	29.–30.04.2014, Anmeldung	Linz	
	29.–30.04.2014, Anmeldung	Graz	
	29.–30.04.2014, Anmeldung	Lustenau	
Continuous Integration für agile Teams	29.04.2014, Anmeldung	Wien	Software Quality Lab
	29.04.2014, Anmeldung	Linz	
	29.04.2014, Anmeldung	Graz	
	29.04.2014, Anmeldung	Lustenau	

[Konferenz Kalender]

Konferenz	Termin	Ort	Deadline Call for Paper
Software Quality Days 2014	14.-16.01.2014	Wien	
ignite	20.-22.05.2014	Düsseldorf/ Deutschland	
OA&TEST	22.-24.10.2014	Bilbao/Spanien	31.03.2014



2014
software
quality days
EXPERIENCE THE VALUE OF QUALITY

14. – 16. Jänner 2014, Wien

EINLADUNG

**Workshops, Tutorials, Vorträge, Networking
und die Tool Challenge**

Mehr als 50 Fachvorträge und Workshops
Testen und Automatisieren, Agiles, Tools, Anforderungen, Prozessverbesserung,
Embedded Systems und vieles mehr
Schwerpunkt: model driven approaches for advanced software engineering

Keynotes: Prof. Dr. Hermann Sikora, Prof. Dr. Reinhard Wilhelm, Prof. Bernhard Ludwig





➔ www.software-quality-days.com/shop



Bezahlte Einschaltung



Accelerate. Innovate. Automate.

**Accelerate
Business
Innovation**

Enterprise Software
Test Automation

www.tricentis.com



[Öffentliche ISTQB Prüfungstermine]

Der nächste öffentliche Prüfungstermin in Wien ist am Mittwoch, 29.01.2014.

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



[Der nächste Schritt zu professionellen Service- und Softwaretests

Real Application Testing]

Vorstellung neuer Testtechniken

Teil 2, von Torsten Zimmermann

Beispiel 2: Konsolidierung und Validierung neuer Hardware

Aufgabe

In vielen Rechenzentren von Unternehmen und Konzernen sieht man heute häufig folgendes Szenario: Es gibt viele – meist Hundert oder mehr – Oracle Datenbanken auf unzähligen Servern verteilt, welche die Daten für Anwendungen verwalten. Ferner kommt nicht auf allen Servern das gleiche Betriebssystem zum Einsatz noch stammen die Server aus der gleichen Hardwarefamilie. Oft laufen verschiedene Versionen der Datenbank-Management-Systeme innerhalb des Rechenzentrums, auf denen jeweils ein Teil der Datenbanken residieren. Das gesamte Umfeld ist somit schwierig zu verwalten.

So liegt es nahe, dass das IT-Management die Menge von Datenbanken in einer standardisierten Plattform konsolidieren will. Das Ziel sollte es hierbei sein, die Anzahl der Datenbanken als die hierzu benötigten Server deutlich zu reduzieren. In der Folge reduziert sich der Wartungs- und Administrationsaufwand im Rechenzentrumsbetrieb.

Methode

Das erste Ziel ist es, eine kosteneffiziente Hardware zu finden, die den Lastanforderungen aus der Produktion genügt und darüber hinaus noch Leistungsreserven bietet. Auf Basis dieser Anforderungen kann man von Hardwareherstellern problemlos Angebote zu verschiedenen Lösungsoptionen bekommen. Die Frage ist jedoch: Wie kann sinnvoll bestätigt werden, ob die Behauptungen des Herstellers tatsächlich für das konkrete Umfeld zutreffen? Und welche der angebotenen Plattformen stellt sich als die beste Alternative dar? Wäre vielleicht das kostengünstigste Angebot ausreichend, da die betreffende Hardwarelandschaft die Lastanforderungen ausreichend erfüllt? Reine Ermittlung und Vergleiche der Performancewerte einzelner Systeme helfen nicht wirklich weiter, da letztlich das konkrete IT-Umfeld und das Lastverhalten unberücksichtigt bleiben.

Real Application Testing liefert jedoch die notwendigen Aussagen, da in einer Testumgebung mithilfe echter Last und Datenvolumen aus der Produktion die Eignung der verschiedenen Hardwarealternativen überprüft werden kann.

Um Leistungseinbußen im SQL Umfeld zu vermeiden, wird SPA zur Identifikation potenzieller Leistungsverschlechterungen eingesetzt. Das heißt, hier ist der gleiche Ansatz wie in Beispiel 1 – konkret sind dies die Maßnahmen des ersten Abschnittes – auszuführen.

Nachdem die Fragen bzgl. möglicher Leistungseinbußen beantwortet sind bzw. etwaige Leistungsschwächen beseitigt wurden, kann die Arbeitsauslastung der Hardware mithilfe des Tools Database Replay geprüft werden. Alle Datenbankanalysen werden mithilfe von AWR durchgeführt. Deren Berichte liefern alle Details zu Datenbankzeiten. Daneben sollte man während der gesamten Untersuchung auch die OS Statistiken im Auge behalten:

- Die erste Replay-Iteration dient lediglich dazu zu prüfen, ob der entwickelte Testlauf keine großen Fehler beinhaltet. Diese sollten zuerst behoben werden, bevor der eigentliche Test beginnt, um belastbare Ergebnisdaten zu erhalten. Die Replay-Funktion bietet eine Reihe von Parametern, welche das Systemverhalten beeinflussen. In Bezug auf das konkrete System beziehungsweise die Datenbankanwendung können die optimalen Einstellungen variieren. Das heißt, es gibt keine „Goldene Regel“ hierzu. Bevor also die eigentliche Vorprüfung oder die nachfolgenden Tests beginnen, ist Replay über geeignete Parameterwahl an das konkrete System anzupassen. Hierbei sind vor allem die Parameter `sync`, `think_time` und `connect_time` zu berücksichtigen, worüber das Datenvolumen beim Replay pro Zeiteinheit beziehungsweise der Replay-Modus gesteuert werden kann.

- Nach erfolgreicher Vorprüfung kann der erste Test gestartet werden. Hierbei wird mit der vom Hersteller empfohlenen Hardwarekonfiguration begonnen. Vor den Tests sind nochmals die Replay-Parameter zu prüfen.

- Die aufgezeichnete Workload wird auf jeder Plattform abgespielt, um jegliche Abweichung zur Produktion festzustellen. Es werden die 10 wichtigsten Performanzaussagen, wie zum Beispiel CPU, I/O oder Zeitbedarf aus den AWR Reports, für die Analyse und Bewertung ausgewählt. Der AWR Report dokumentiert für jede Konfiguration alle relevanten Informationen beziehungsweise Messwerte zu Zeit- und Ressourcenverbrauch zu allen wichtigen Hardware- und Datenbankkomponenten.

- Aufgabe 1/Identifizierung des CPU-Engpasses: Hierbei wird das Abspielen der Workloads mehrfach wiederholt. Dabei wird die Anzahl der verfügbaren CPUs schrittweise verringert. Dabei ist es wichtig, über den AWR die Wartezeiten zu ermitteln. Mit dieser Methode kann die Minimalconfiguration eines Systems ermittelt werden, welche für einen erfolgreichen Betrieb benötigt wird.

- Aufgabe 2/Plattformskalierbarkeit: Der aufgezeichnete Workload wird hierbei mit dem `SCALEUP_MULTIPLIER` ausgeführt. Der `SCALEUP_MULTIPLIER` multipliziert die Benutzeranzahl und die Nur-Lese-Operationen von jeder Shadow Session. Daneben kann man aber auch die `connect_time` und `think_time` verringern, um eine höhere Last zu simulieren. Die Änderung der `SCALEUP_MULTIPLIER` (sinnvoll sind hier Werte von 2, 4 oder 8) hilft, die Skalierbarkeit für jede Konfiguration zu ermitteln. Diese Methode ermittelt die Plattform mit dem besten Preis-Leistungs-Verhältnis.

>

Real Application Testing

- Aufgabe 3/Verringerung der Datenbankanzahl durch Serverkonsolidierung: Hierbei ist zu prüfen, ob mehrere Anwendungen in derselben Datenbank koexistieren können. Diese Überprüfung kann mithilfe des sogenannten „Consolidated Replay“ erfolgen. Damit ist es möglich, mehrere Workloads verschiedener Anwendungen beziehungsweise Datenbanken auf der gleichen Datenbank einzuspielen. Dies kann in folgenden Schritten durchgeführt werden:

- Aufzeichnung der SQL Tuning Sets für die betreffenden Datenbanken, welche zu konsolidieren sind.
- Ausführen der SQL-Performance-Analyser-Leistungsanalyse für die betreffenden Datenbanken, welche zu konsolidieren sind, um SQL Wiederholungen zu identifizieren.
- Aufzeichnen des interessanten Zeitbereichs eines Workloads von jeder Datenbank, welche zu konsolidieren wäre.
- Zunächst wird jeder Workload isoliert abgespielt, um etwaige Leistungsprobleme oder Fehler zu beseitigen.
- Ausführung eines Consolidated Replays: Nach jeder erfolgreichen Iteration beziehungsweise Beseitigung aller Probleme und Fehler in einem aufgezeichneten Workload wird besagte Aufzeichnung an das Abspielschema angefügt. Die Leistungswerte des Servers und der Datenbank werden dabei überwacht und bewertet. Der Replay-Report zeigt neue Fehler oder Leistungsprobleme auf, welche während der Wiedergabe auftreten. Diese sind dann natürlich zu beseitigen. Ist dies unmöglich, so wird hierdurch angezeigt, dass sich die betreffende Datenbank der zuletzt hinzugefügten Workload Aufzeichnung nicht mit den anderen Datenbanken konsolidieren lässt.

Ergebnisse

Mit diesem Ansatz kann rasch ermittelt werden, welche der Plattformen das beste Preis-Leistungs-Verhältnis bietet. Darüber hinaus kann die Frage beantwortet werden, welche Risiken bei einer Konsolidierung von Anwendungen zu beachten sind. Die Tests zeigen ferner auf, welche Anwendungen auf derselben Datenbank residieren können, ohne sich gegenseitig zu stören. .

Des Weiteren liefern die Tests Informationen über den zukünftigen Ressourcenbedarf. Gerade bei Konsolidierungen, in denen Datenbanksysteme zusammengefasst werden, sind diese Informationen besonders wichtig.

Beispiel 3: Statistik Refresh

Aufgabe

Ein großer Webshop kann seine Tabellenstatistiken nicht mehr aktualisieren, seitdem das Erneuern der Statistiken zu einigen SQL Regressions und geringerer Systemperformance führte. Das System läuft also seit Langem mit veralteten Statistiken. So reduzierte sich die Systemleistung während der letzten Monate. Nun müssen neue Statistiken erzeugt werden, um die Performance zu erhalten. Bei dieser Maßnahme könnten aber neue Performanceprobleme auftreten.

Ansatz

Seit der Oracle Datenbankversion 11g R1 können Statistiken im sogenannten „Pending Mode“ gesammelt werden. Dies bedeutet, dass neue Statistiken erzeugt, aber die entsprechenden Informationen nicht im Optimizer verwendet werden: Sie werden nicht veröffentlicht.

Der SQL Performance Analyzer (SPA) kann Statistiken im Pending Mode verwenden, um die Leistung des Datenbanksystems zu überprüfen. Der Enterprise Manager 12c enthält eine Benutzeroberfläche zur Steuerung des SQL Performance Analyzers. Hierüber lassen sich neue Statistiken einfach überprüfen. Da SPA nicht die Datenbankinhalte verändert, besteht keine Gefahr, dass bei der Überprüfung der Produktionsumgebung deren Leistung reduziert wird. Die folgenden Schritte müssen durchgeführt werden:

- Das SQL Tuning Set (STS) aus der Produktion wird aufgezeichnet. Hierbei wird empfohlen, mehrere STS aus verschiedenen Geschäftszeiten aufzuzeichnen, wie z. B. normale Bürozeiten, Nachtbetrieb und Monatsende, welche unterschiedliche Lastsituationen repräsentieren.
- Die neuen Statistiken werden im Pending Mode aufgezeichnet.
- Nun wird der SPA Workflow „Optimizer Statistics“ in der Produktionsumgebung ausgeführt.
- Mithilfe des SPA Reports werden die SQL Regressions identifiziert.

- Die betreffenden SQL Anweisungen werden nun optimiert. Hierbei bietet der Tuning Advisor zusätzliche Einblicke bezüglich SQL Profile, neuer Indizes oder neu angelegter Statistiken. Wenn diese Maßnahmen keine sinnvollen Ergebnisse liefern, so kann der alte Plan in das SQL Plan Management eingefügt werden.

- Nach der Optimierung werden die SPA Tests erneut ausgeführt. Hierbei sollte stets nur ein Optimierungsschritt – und nicht mehrere Schritte – zwischen zwei Prüfungen durchgeführt werden. So lassen sich im Fehlerfall schneller die Ursachen ermitteln. Selbst ein einziger Tuningschritt bzgl. einer SQL Anweisung kann zu unbeabsichtigten, negativen Effekten bei anderen SQL Anweisungen führen. Durch die konsequente Zerlegung der Tuningaufgaben in Einzelschritte mit nachfolgender Validierung vereinfachen sich in diesen Fällen die Analyseaktivitäten.

- Die Tuningratschläge sind zu implementieren und die betreffenden SPA Tests auszuführen, bis alle SQL Wiederholungen in einem iterativen Prozess aufgelöst sind.

- Im Rahmen der Optimierung ist stets zu prüfen, ob nicht an anderer Stelle durch die Veränderungen Leistungseinbußen entstanden sind. Wenn am Ende alle Anforderungen erfüllt sind und alle Optimierungspotenziale genutzt wurden, so kann die neue Statistik veröffentlicht werden.

Ergebnisse

Durch die Maßnahmen ist es gelungen, die SQL Anweisungen zu optimieren und auf dieser Basis neue Statistiken zu etablieren. So können negative Folgen auf businesskritische Systeme aufgrund veralteter Statistiken ausgeschlossen werden.

Schlussfolgerungen

Die Gewährleistung von Qualität und Leistung auf Unternehmensanwendungen erfordert die Etablierung eines umfassenden Qualitätsmanagements. Dies bedeutet auch, umfassende Prüfungen auf allen Ebenen vor der eigentlichen Überführung in die Produktion vorzunehmen. Hierbei sind Anwendungs- und Infrastrukturtests sowohl bei Neuentwicklungen als auch bei Updates durchzuführen. Umfassende Tests auf hohem Niveau müssen hierbei auch die realen Betriebsbedingungen berücksichtigen können, damit deren Aussagen eine gute Basis für die Entscheidung

>

Real Application Testing

darstellen, ob eine Anwendung für die Produktion freigegeben werden kann. Um die Effizienz zu maximieren, ist es wichtig, auf ein effektives Testframework zurückgreifen zu können: Hierüber lassen sich Tests sinnvoll planen, verwalten und auch automatisiert ausführen.

Der Oracle Enterprise Manager bietet ein umfassendes Set von Qualitätsmanagementlösungen, wie die Application Testing Suite, das Real Application Testing und das Data Masking Pack. Diese Oracle Testtools sind vor allem auf das Oracle Datenbankumfeld abgestimmt und liefern wichtige Informationen, welche mit anderen Testwerkzeugen entweder gar nicht oder nur mit erheblich höherem Aufwand zu ermitteln wären. Ferner ermöglichen die Werkzeuge, qualitativ hochwertige und sichere Tests mit geringem Umgebungsrisiko beziehungsweise geringem Risiko ungewollter Änderungen bei den Anwendungen.

Real Application Testing hat sich schon vielfach in verschiedenen Testkonstellationen bewährt. Steht ein geplanter Wechsel oder das Einspielen neuer Patches an, wird Real Application Testing bei performance-kritischen Anwendungen eingesetzt. So können Probleme frühzeitig aufgedeckt und gelöst werden. Weitere Einsatzgebiete sind Tests beim Plattformwechsel oder eine Lasttests. Bemerkenswert ist, dass - trotz der umfangreichen und detaillierten Testergebnisse im Datenbankumfeld - keine aufwändigen und komplizierten Testvorbereitungen und Testspezifikationen durchzuführen sind.

Aus diesen Gründen sollten sich Entwickler und Testingenieure mit den Oracle Testtools vertraut machen, sofern Oracle Datenbanken in der betreffenden Umgebung eingesetzt werden.

Die Vorteile von Real Application Testing

Im Vergleich zu klassischen Softwaretests bietet Real Application Testing folgende Vorteile, wie diese in mehreren IT-Projekten bereits erkannt wurden:

1. Bei der Verwendung von Workload Daten aus der Produktion können während der Tests bereits produktionsbezogene Probleme identifiziert und korrigiert werden, welche ansonsten erst in der Produktion zu erkennen wären.
2. Mithilfe von Real Application Testing verringert sich der Anteil von „Feuerwehreinsätzen“ für Datenbankadministratoren bei Problemen in der Produktion. Damit können diese Ressourcen auf weitere proaktive und strategische Aufgaben ausgerichtet werden.
3. Die notwendige Testinfrastruktur kann mit Real Application Testing kleiner ausfallen als sonst üblich: Das Database Replay benötigt nicht die Middle und die Webserver Tier. Dies reduziert den Verwaltungs- und Pflegeaufwand für die betreffende Testumgebung und den Aufwand für die Testvorbereitung.
4. Mithilfe von Database Replay müssen zur Bewertung von Datenbank- und Umgebungsqualität weder funktionale Kenntnisse über die Anwendung erworben noch umfangreiche Testskripts entwickelt werden. In vergleichsweise kurzer Zeit ist mit wenigen Klicks die volle Produktionslast auf der Testumgebung verfügbar. Hierauf können die Tests und Analysen mithilfe der betreffenden Werkzeuge – wie dem SPA – ausgeführt werden. Durch die reduzierten Zeiten im Bereich der Testvorbereitung und Testspezifikation verkürzt sich die Dauer der Testzyklen.
5. Der neue Testansatz sorgt für einen reibungslosen „Go-live“ in die Produktion, ohne auf Überraschungen bezüglich unerkannter Schwächen in der Produktion zu stoßen.

[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



>

Real Application Testing

Technischer Vergleich SPA zu Database Replay

Die nachfolgende Tabelle erläutert die Unterschiede zwischen der SPA und Database Replay Methode:

	SPA	Database Replay
Testfokus	Einzel-Select-Analyse	Gesamte Datenbank-Workload-Analyse
Wiederholung möglich?	ja	Ja, mit entsprechender Konfiguration
Installation	Keine; unter Umständen Patches notwendig siehe auch Oracle Support Note	Keine; unter Umständen Patches notwendig siehe auch Oracle Support Note
Nutzung vor 11g	in folgenden Fällen: von 9.x nach 10.x von 9.x nach 11g von 10.x nach 10.x von 10.x nach 11g von 11g nach 11g siehe auch Oracle Support Note 560977.1	in folgenden Fällen: von 9.2.0.8 nach 11g von 10.2.0.x nach 11g siehe auch Oracle Support Note 560977.1
Art des Workload	SELECT-Statements, DML über Package-Nutzung	Gesamter Datenbank-Workload wie z.B: SELECT, PL/SQL, DML, DDL, Transaktionen etc.
Testvoraussetzungen	zuvor ist SQL Tuning Set (STS) auszuführen	keine
Testergebnisse	Einzel-Ausführungspfad-Analyse	Capture und Replay Report, Compare Period Report, AWR und ASH Reports

Nützliche Hinweise

1. Real Application Testing reduziert den Zeitbedarf für die Tests. Trotzdem ist es wichtig, einen Testprojektplan und ein Testkonzept zu erstellen, in dem Umfang, Ziele, Einschränkungen und Erfolgskriterien klar definiert sind.
2. Es sollen stets die aktuellen Patches und die aktuelle Version zu Real Application Testing in beiden Umgebungen (in welcher aufgezeichnet und in welcher abgespielt wird) verwendet werden. Die Patchinformationen finden sich im My Oracle Support Hinweis #560977.1.
3. Der Workload Analyzer ist immer zuerst – vor der eigentlichen Wiedergabe – auszuführen, um etwaige Probleme bzgl. nicht unterstützter Workloads oder anderer Probleme zu erkennen.
4. Die Prüfungen sollen stets mit dem SQL Performance Analyzer beginnen, um ein Bild über etwaige Performanceprobleme zu erhalten.
5. Zunächst sind alle bekannten Performanceprobleme zu lösen, bevor mit dem Database Replay fortgefahren wird.
6. Das Aufzeichnen der SQL Tuning Sets (STS) sollte im gleichen Zeitraum erfolgen wie das Database Replay.
7. Alle kritischen Workload Bereiche sind zu identifizieren: Für jeden Bereich sollte eine eigene Aufzeichnung erstellt werden.
8. Das Aufzeichnen der Workloads sollte nicht mit der Lastspitze beginnen, sondern bereits einige Zeit davor starten und bis über die Lastspitze hinweg aufzeichnen.
9. Wenn Batchläufe ausgeführt werden, so ist die Aufzeichnung des Workloads (ähnlich wie bei Punkt 8) bereits einige Zeit davor zu starten.
10. Zunächst sollte man mit kleineren Capture-Perioden, wie beispielsweise 30 Minuten bis 1 Stunde, beginnen. Sobald alle Probleme gelöst worden sind, kann die Aufzeichnung auf längere Zeiträume ausgedehnt werden.
11. Aufzeichnungszeiträume von mehreren Tagen sollten vermieden werden. Es wird schwierig sein, diese Aufzeichnungsprozesse vollständig zu überwachen. Ferner besteht die Gefahr, dass selbst bei einer Fehlereintrittswahrscheinlichkeit von 0,1 % aufgrund der riesigen Datenmengen der Wiedergabeprozess abbricht.
12. Die Wiedergabe von Workloads sollte nicht auf kleineren Systemen abgespielt werden, als diese aufgezeichnet wurden. Sollten die Workloads auf kleineren Systemen abgespielt werden, so sind entsprechende Workload Filter zu definieren, um die Last auf dem Zielsystem zu reduzieren.
13. Zwischen den Database Replays sollten Datenbank-Flashbacks als Restore-Methode angewendet werden.
14. Neben den Standard SPA und Database-Replay-Reports sollten auch eigene Berichte entwickelt werden, um spezielle Ergebnisinformationen aufzubereiten und die definierten Testziele besser zu unterstützen.

>

Real Application Testing

Interview mit Mughees Minhas, Vice President Product & Data Management bei Oracle

Guten Tag, Herr Minhas. Zunächst einmal vielen Dank für Ihre Zeit und die Gelegenheit, Fragen zu dem Oracle Enterprise Manager stellen zu dürfen.

Unter Berücksichtigung der neuesten Oracle Enterprise Manager (OEM) Version 12c, was sind die wichtigsten Neuerungen oder die „Big Points“, gegenüber der ehemaligen OEM-Version?

Für OEM 12c hatten wir drei wichtige Ziele definiert:

1. Die Etablierung eines Next-Generation-Management-Frameworks,
2. die Verbesserung des Application-to-Disk-Managements und
3. die Unterstützung von Cloud-Services.

Bezüglich des ersten Punktes haben wir das OEM Framework weiterentwickelt, um es robuster, skalierbarer und leistungsfähiger gegenüber seiner Vorgängerversion zu machen. Wir haben eine neue Architektur entwickelt, welche nun sehr einfach Erweiterungen integrieren kann. Die neue OEM Version stellt nun Managementfähigkeiten zur Verfügung, welche eine einfache Synchronisation mit anderen Produktversionen aus unserem Hause erlaubt. Dies haben wir erreicht, indem wir das Plug-In-Feature integrierten. So können wir zusätzliche Funktionen über Plug-Ins jederzeit freigeben, ohne auf das OEM Release Management Rücksicht zu nehmen. Hier ein Beispiel: Wenn die Oracle Datenbank 12c ausgeliefert wird, wird ein betreffendes Enterprise-Manager-Plug-In am selben Tag versendet.



Mughees Minhas, Vice President Product & Data Management bei Oracle

Die OEM Anwender können jetzt einfach das DB-Plug-In im Enterprise Manager aktualisieren, anstatt die gesamte OEM Plattform anheben zu müssen. Ferner wurde das User-Interface deutlich verbessert, welches nun eine klarere und intuitivere Benutzerführung bietet, als dies in der alten Version der Fall war.

In Bezug auf den zweiten Punkt haben wir viele neue Funktionalitäten im Application Performance Management für Oracle Datenbanken, Middleware und Anwendungen realisiert. Daneben haben wir auch neue Funktionen zum Testdatenmanagement umgesetzt, welche den Betrieb sicherer Testsysteme unterstützen. Diese Features wurden in unsere Datenbank-Testtools, wie das Real Application Testing, integriert.

Aber der „Big Point“ dürfte vielleicht die Unterstützung privater Cloud-Services in OEM 12c sein. Wir haben IaaS- (Infrastructure-as-a-Service), MWaaS- (Middleware-as-a-Service) und DBaaS- (Database-as-a-Service) Lösungen für private Clouds umgesetzt. Diese Produkte haben das Potenzial, die IT grundlegend zu verändern, indem sie agiler und effizienter wird. Ferner werden diese Cloudprodukte die Qualität der dem Unternehmen angebotenen Informationsdienstleistungen verbessern.

Was waren die größten Herausforderungen für das Entwicklungsteam während der Entwicklung bis zur erfolgreichen Freigabe von OEM 12c?

Immer wenn es darum geht, eine umfassende Entwicklung für ein Major Release zu unternehmen, welches von Zehntausenden von Kunden genutzt wird, besteht die größte Herausforderung darin, die vorhandenen Produktfunktionalitäten nicht zu beeinträchtigen. Ich denke, für OEM12c haben wir diese Herausforderung gut gemeistert. OEM12c ist meiner Meinung nach die beste Version im Vergleich zu allen Versionsvorgängern.

Bezüglich Real Application Testing – welches über OEM gesteuert werden kann – wie sehen Sie die Verbesserungspotenziale für Softwaretests mit Blick auf zukünftige Produktversionen?

Real Application Testing wurde für uns zu einem Hauptprodukt, welches für unsere Kunden die Datenbanktests deutlich weiterentwickelt hat. Real Application Testing konnte die meisten Datenbankprobleme erkennen und

verhindern, dass diese in Produktionsumgebungen von Kunden übertragen wurden. Gleichzeitig sagen uns Kunden, welche Real Application Testing verwenden, dass Produktionsausfälle nach größeren Upgrades mithilfe dieses Tools gegen Null gehen.

Auch bei Oracle testen wir mit Real Application Testing unsere Datenbank Versionen, bevor wir diese für die Produktion freigeben. Dieser Ansatz hilft uns enorm dabei, dass die Qualität jeder neuen Softwareversion höher als beim Vorgänger ist.

Können Sie uns einige Hinweise geben, welche Funktionen oder Funktionalitäten Benutzer in zukünftigen OEM Versionen wie auch bezüglich real Application Testing in 2014 oder 2015 erwarten dürfen? Könnten Sie die jeweilige Roadmap in groben Zügen erklären?

Normalerweise geben wir keine Auskünfte über zukünftige Roadmaps. Alles, was ich sagen kann, ist, dass auch weiterhin mehr Innovationen in unsere Produkte fließen werden. Der Fokus liegt hierbei auf der Frage, wie zukünftig Cloud-Services Rechenzentren verändern werden.

◀ (Torsten Zimmermann)

[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.

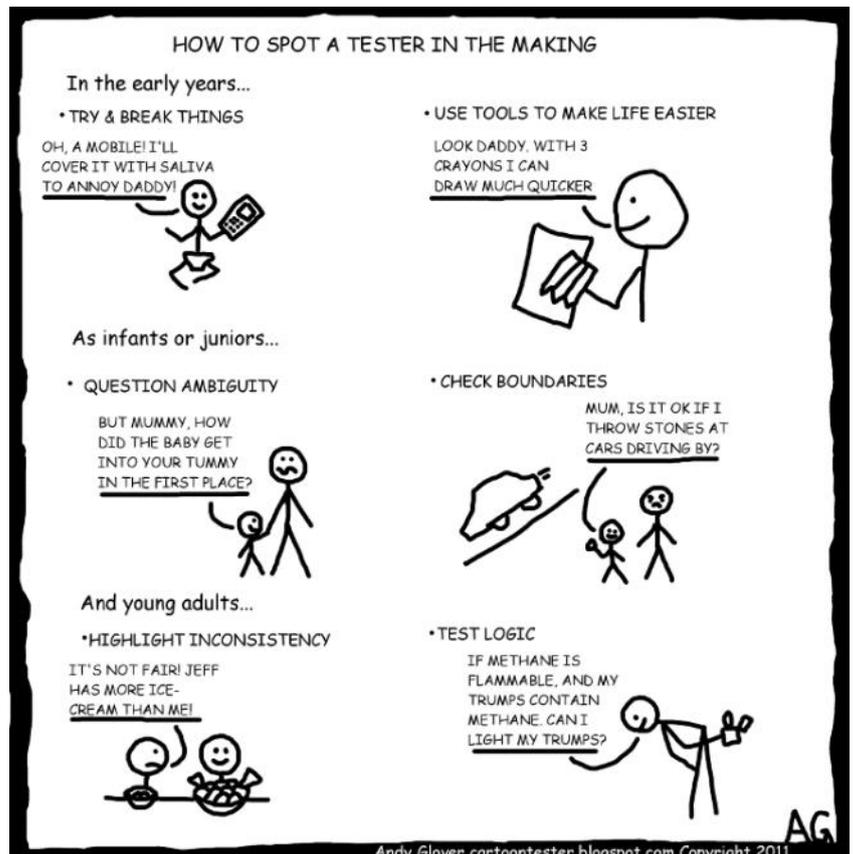
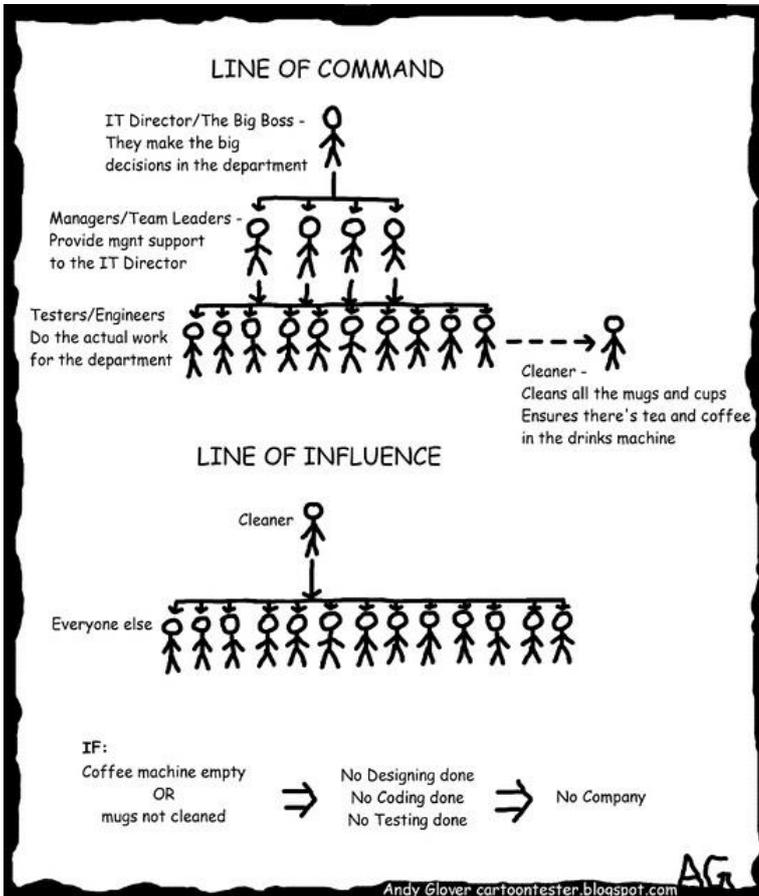
Wir suchen noch Redakteurinnen und Redakteure, die Spaß am Schreiben haben.

Einsendungen an:
newsletter@austriantestingboard.at

◀ (Karl Kemminger)

[Cartoon]

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



Andy Glover cartoontester.blogspot.com Copyright 2011