

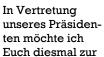
ATB-Insider

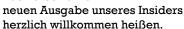
Das Magazin von Testern für Tester

Ausgabe Juli 2013

[Editorial]

Liebe Test-Community!





Die ersten beiden ATB Expertentreffs waren ein voller Erfolg (der dritte Treff fand nach Redaktionsschluss dieser Ausgabe statt). Neben einer Rückschau auf den zweiten Treff finden Sie eine Vorschau auf die nächste Veranstaltung nach der Sommerpause. Auch das ASOF veranstaltet regelmäßig Fachgruppenabende, wobei nicht nur Themen, die gerade im Mainstream liegen, behandelt werden, sondern erfreulicherweise auch wichtige Spezialgebiete wie Barrierefreiheit

Immer mehr werden in der heutigen Welt mobile Geräte und die auf diesen installierten Apps eingesetzt. Daher gewinnt auch das Thema Testen dieser Apps immer mehr an Bedeutung. Tipps zur Umsetzung finden Sie in dieser Ausgabe.

Welche Methode eignet sich am besten, um die widersprüchlichen Ziele wie hohe Qualität und rasche Realisierung bei Ihrem Projekt unter einen Hut bringen zu können, und die Teamleistungsfähigkeit zu erhalten? Die meisten Methoden unterstützen hier nur Teilaspekte - Torsten Zimmermann beschreibt einen Ansatz, wie man durch Kombination mehrerer Methoden alle Ziele abdecken kann. Im ersten Teil dieser mehrteiligen Artikelserie werden SCRUM, KAIZEN und KANBAN vorgestellt. Dazu gibt es wieder eine Buchbesprechung - vielleicht als Anregung, wenn Sie noch keine Lektüre für den Sommerurlaub

haben. Gut Test!

◀(Karl Kemminger)

[ATB Expertentreff - Requirement Spezifikation im agilen Umfeld]

von Alexander Weichselberger

Nach den ersten Veranstaltungen 2013 zu den Themen Agile vs. ISTQB, Testautomation im mobilen Umfeld und Crowdtesting widmen wir den ersten Treff nach der Sommerpause dem Requirement Engineering in agilen Projekten:

Was machen wir Tester, wenn die Software noch nicht entwickelt wurde und wir auf die ersten Lieferungen warten müssen? Richtig, wir nutzen die Zeit und starten parallel zur Entwicklung mit unseren Testvorbereitungen und entwickeln Testkonzepte, -fälle, -szenarien, usw.. Wie ist das aber im agilen Kontext?

Hier gibt's neben rudimentären Userstories, die erst im Rahmen der Sprint Plannings und im Sprint konkretisiert werden, kaum wirklich Greifbares, auf das wir setzen können – alles ist noch im Fluss und unterliegt im Regelfall Adaptionen und Konkretisierungen. Weiters setzen agile Entwicklungen auch auf iterative Zyklen -> Testautomation ist für die Regressionstests einzusetzen, stößt aber auch hier wieder auf Herausforderungen wie die oben angedeuteten "moving targets" bzw. die Nichtverfügbarkeit der Software selbst.

Hmm – viele Gründe, warum gerade wir Tester keine Fans von agilen Methoden sein müssten → ist aber nicht so, wie wir beim ersten Treff heuer im März gemeinsam festgestellt haben. Woran liegt's? Wie immer gibt es brauchbare Tipps aus der Praxis, die helfen und Lösungsansätze sind.

Im Rahmen des ATB Expertentreffs werden

diese von Kennern vorgestellt und diskutiert, in diesem Fall vorgestellt von Johannes Bergsmann, Chef des Software Quality Lab der uns über die verschiedenen Techniken der Requirement Spezifikation wie TDD, BDD, Userstories, Use-Case Description und Specification by Example informieren wird.

Falls Sie selbst zu diesem Thema Ihr Knowhow anbieten und z.B. in der Diskussionsrunde auf die Bühne wollen: Melden Sie sich bitte gerne bei mir, im Rahmen des Expertentreffs stellen wir gerne Kapazitäten aus der Community vor.

Ich freue mich bereits heute auf den Termin am 4.9., Treffpunkt ist wie immer das TechGate Vienna, und wünsche bis dahin einen erholsamen Sommer und alles Gute.

Ihr Alexander Weichselberger

ATB Expertentreffs

Nutzen Sie die Möglichkeiten zum Erfahrungsaustausch im Rahmen des ATB Expertentreffs. Informationen zu den Veranstaltungen und zur Anmeldung finden Sie auf unserer Homepage.

Wenn Sie aktuelle Themen im ATB Expertentreff behandeln lassen wollen melden Sie sich bitte bei

weichselberger@austriantestingboard.at.

Eine Rückschau auf das Experttreff vom 8.5. zum Thema "Testautomatisierung im Mobilen Umfeld" finden Sie auf Seite 8.

◀(Alexander Weichselberger)

[Inhalt]

-	Goldpartner Informationen	Seiten 2 -3
-	Testen mobiler Apps	Seiten 4-6
•	Buchbesprechung	Seite 6
•	ASQF	Seite 7
•	ATB Experttreff	Seite 8
•	Seminare	Seiten 9-10
•	Konferenzen, Silberpartner Information, iSQI	Seite 11
•	Agile Entwicklungsmethoden in der Praxis	Seiten 12-16

Neues Seminarprogramm und aktuelle Termine



Das neue Seminarprogramm

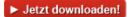
Professional Scrum Master, Aufwandsschätzung, Moderationstechniken und mehr

Im neuen Seminarprogramm finden Sie ab sofort zwei neue Kategorien für Entwickler und Tester:

- Implementierung und Integration
- Agile Softwareentwicklung

Sie beinhalten Seminare wie Professional Scrum Master, Requirements Spezifikation in agilen Teams oder Unit-Testen in der Praxis.

Werfen Sie einen Blick in das neue Programm!



Aktuelle Termine für Requirements Engineers

- IREB® CPRE Foundation Level NUR € 990,-
 - » 16.09. 18.09. in Graz, Linz, Wien
- IREB® CPRE Foundation Level mit Praxistag NUR € 1.480,-
 - » 16.09. 19.09. in Graz, Linz, Wien





Mit Frühbucherrabatt zu Konferenz und Workshops

14.01. – 16.01.2014 Wien, Austria Trend Hotel Savoyen



Very Early Bird

Acceptance Café 2.0

BDC acceptance - that's us

15. Oktober 2013, Beginn: 9:00 (ab 8:30 Check-in) - 11:30 Uhr Café Griensteidl, Wien

Referent: Univ.-Doz. Dr. Ernest Wallmüller

Thema: Softwarequalität – Schlagwort und Realität? Ist die Software-Krise überwunden oder wird sie immer wieder neu erfunden?

Software bzw. Systeme, die durch sie gesteuert werden, sind ein kritisches Element in unser aller Leben geworden. Unsere Gesellschaft und insbesondere unsere Firmen hängen von ihr ab. Software-Fehler sind nicht nur frustrierend für die Anwender, sondern sie beeinflussen auch die Profitabilität. Sie stellen Sicherheitsrisiken dar, und mit dem Internet bzw. der globalen Vernetzung sind sie Ursachen von massiver Verletzlichkeit. Gerade mit der aktuellen Thematik des Cloud-Computings, die bereits exemplarische Auswirkung auf die strategische IT-Planung von Banken und Softwarehäusern zeigt, verschärft sich die Situation nochmals.



Der Vortrag behandelt **aktuelle Ansätze** und beschreibt **Lessons Learned**, die im **Software und System Engineering** angewandt werden können, um die **Software-bzw. Systemqualität zu verbessern**.

Details & Anmeldung: www.bdc.at/kontakt/events.html

BDC-Trainings - Details und Termine: www.bdc.at/trainings

SQS Training

Professionell und individuell



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 30 Jahren Beratungsaktivität. Die Stärke der SQS-Seminare liegt in der Expertise der Trainer.

Zwölf Rollen als Zielgruppen unserer Seminare

Quality Manager, Test Project Manager, Test Manager, Requirements Engineer, Developer, Functional Tester, Agile Tester, User Acceptance Tester, Test Automation Specialist, Performance Test Specialist, Test Environments Specialist, Security Testing Specialist

Mehr Informationen finden Sie hier

Kommende Seminartermine der SQS in Wien:

- ISTQB® CT Foundation Level, 02.-05.09.
- ISTQB® CTAL Test Manager, 16.-20.09.
- Certified Agile Tester® (CAT), 23.-27.09.
- ISTQB® CTAL Test Analyst, 14.-18.10.
- ISTQB® CT Foundation Level, 04.-07.11.
- ISTQB® CTAL Technical Test Analyst, 18.-22.11.
- ISTQB® CTAL Test Manager, 02.-06.12.

Alle Seminare von A-Z finden Sie hier

Wir stellen ein: Testmanager (w/m) & Test-Consultants (w/m). Bewerbungen an Barbara Stoik: recruiting@sqs.at



Sie möchten SW-Qualität steigern und Gesamtkosten verringern? Dann wenden Sie sich an ANECON: www.anecon.com

Testen mobiler Apps

[Testen mobiler Apps: ein Paradebeispiel für agiles Vorgehen]

von Sven Euteneuer

Mobile Apps verändern

In der letzten Zeit ist ein Paradigmenwechsel am IT-Markt zu beobachten. Aktuelle Studien zeigen, dass eine Abkehr vom PC als "geschäftlichem Alleskönner" hin zu einer stärkeren Nutzung mobiler Geräte (vgl. (IDC, Inc., 2013)) erfolgt. Mobile Geräte und Anwendungen durchdringen den privaten und beruflichen Alltag und werden zum Innovationstreiber der Informationsund Kommunikationstechnologie (IKT). Eine Folge der stärkeren Verbreitung und Nutzung der Informationstechnologie ist unter anderem eine sich ändernde Nomenklatur. So wird aus Software eine App, und deren Verteilung erfolgt nicht mehr über DVDs oder FTP-Server, sondern über App-Stores. Dabei stellt Mobile Computing in Bezug auf eine Vielzahl von Aspekten eine veritable Revolution dar: Nicht nur verändern mobile, smarte Geräte die Art und Weise geschäftlichen wie privaten Lebens, sie erfordern gleichermaßen eine Anpassung der Art und Weise, wie Software für sie entwickelt wird. Indiz hierfür ist, dass mobile Anwendungen nicht als Anwendungen, Programme oder Software bezeichnet werden, sondern als Apps - ein Begriff, der bereits darauf hindeutet, dass sich mobile Software von bekannten Pfaden entfernt.

Die geänderten Rahmenbedingungen stellen auch die Softwareentwicklung und damit die Qualitätssicherung vor Herausforderungen. Konkret ergibt sich eine Reihe von Herausforderungen, die mobiles Entwickeln im Allgemeinen und mobiles Testen im Besonderen zu Anpassungen zwingen. Dies sind zunächst technische Spezifika mobiler Geräte, wie zum Beispiel begrenzte Rechen- oder Speicherkapazität, Unterschiede in der Benutzerschnittstelle oder Einflussfaktoren wie wechselnde Netzanbindung oder Unterbrechungen durch Anrufe, Erinnerungen oder andere Apps.

Zusätzlich erfordert die Heterogenität des mobilen Marktes die Anpassung und Erweiterung des Prinzips des risikobasierten Tests. Um die endlichen Testressourcen möglichst optimal zu verteilen, genügt es nun nicht mehr, nur Testobjekte anhand ihres Risikos zu klassifizieren und mehr oder weniger gründlich zu testen. So ergibt sich für mobile Apps häufig das Problem, dass mit der Anzahl der Vertriebskanäle (z.B. PC vs. Web vs. Mobile vs. TV), Plattformen (z.B. Android, iOS, Black-Berry, Windows Phone) und konkreter Geräte die Anzahl möglicher Produktvarianten und damit auch die Anzahl denkbarer Testfälle sowie die Anzahl der Testumgebungen stark steigt (vgl. (Euteneuer, 2012)).

Darüber hinaus unterscheidet sich Softwareentwicklung für mobile Geräte aber noch in einem viel wesentlicheren Aspekt von traditioneller IT: Die Innovationszyklen sind noch einmal erheblich kürzer, die Anforderungen wechseln mit noch größerer Volatilität und die Hürden für Mitbewerber mit einer guten Idee sind noch niedriger. Um in diesem Umfeld erfolgreich zu sein, ist also unter anderem eine sehr kurze "Time to Market" mit guter Qualität erforderlich.

Agilität als Problemlöser

In diesem Kontext erscheinen traditionelle, phasengetriebene Vorgehensmodelle (vgl. Abbildung 1) als Anachronismus, der mit seinen Formalismen, früh festgeschriebenen Anforderungen und erst am Projektende verfügbaren Produkten ungeeignet ist, erfolgreich Produkte zu entwickeln. Gleichzeitig klingen die oben beschriebenen Herausforderungen, als ob sie fast wörtlich aus dem Agilen Manifest entnommen wurden. Dem Dokument also, das die Werte und Prinzipien der agilen Softwareentwicklung zusammenfasst.

Insbesondere fallen die Werte zu funktionierender Software, Zusammenarbeit mit dem Kunden und dem Reagieren auf Veränderungen ins Auge (vgl. (Agile.org, 2001)).

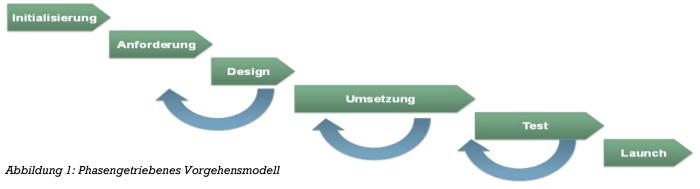
Agilität erscheint also in der Softwareentwicklung für mobile Plattformen noch mehr als die magische Lösung aller Probleme, als es in klassischen IT-Projekten ohnehin der Fall ist. Neben aller Euphorie löst die Verwendung spezifischer agiler Methoden aber in der Tat typische Probleme in der App-Entwicklung (vgl. auch (Spataru, 2010)).

Auf wechselnde Anforderungen reagieren

Im Umfeld mobiler Apps existieren gleich mehrere Treiber für sich verändernde Anforderungen. Zum einen zielt die Natur einer App häufig auf Bedarfe ab, die entweder zeitlich begrenzt sind (z.B. Apps mit Bezug auf Großereignisse oder Werbeaktionen) oder die sich im Laufe der Zeit stark verändern können.

Andererseits werden die Anforderungen an mobile Apps durch das dynamische Technologieumfeld stärker als in anderen Bereichen auch durch die Evolution der mobilen Plattformen mitbestimmt. Anders als auf dem PC erscheinen neue Geräte – und damit Hardware-Innovationen – im Rhythmus eines Quartals. Dazu werden mobile Plattformen wie Android, iOS und Windows Phone mehrmals im Jahr aktualisiert und erweitert (vgl. Abbildung 2 für Android).

Einer der Vorteile der Anwendung agiler Methoden liegt in der schnellen Reaktion auf sich ändernde Anforderungen. Der Einsatz iterativ-inkrementeller Vorgehensmodelle erlaubt es, Anforderungen in regelmäßigen Zyklen zu erheben, zu priorisieren und die zu implementierenden Anforderungen auszuwählen (vgl. Abbildung 3).



Testen mobiler Apps

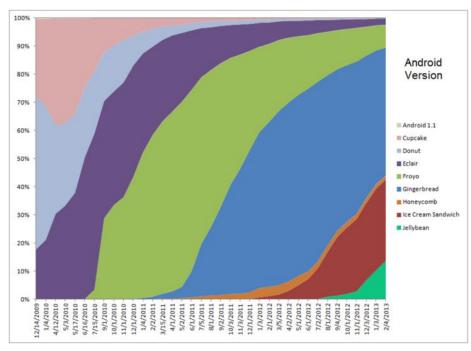


Abbildung 2: Android Releasezyklen, Quelle: (Wikipedia, 2013)

Dies erlaubt es, auch auf dynamische Änderungen schnell und gleichzeitig ohne den in phasengetriebenen Modellen einhergehenden Wertverlust großer Arbeitspakete zu reagieren. Entsprechend sind SCRUM und verwandte Vorgehensmodelle beliebte Prozesse bei der Organisation der Entwicklung mobiler Apps.

Überschaubarkeit von Aktion und Reaktion

Außerdem haben iterative Vorgehensmodelle den Vorteil, dass komplexe Aufgabenstellungen notwendigerweise in überschaubare Implementierungs-Teilpakete heruntergebrochen werden müssen. Dies schafft mehr Transparenz. Der Status des Projektes lässt sich jederzeit kleingranular berichten. Trends und Fehlentwicklungen sind schneller ersichtlich und steuerbar. Der kurze Zyklus zwischen Planen, Entwickeln, Testen und Reviewen erzeugt kontinuierlich Verbesserungen am Produkt

und am Prozess, da sich das Entwicklungsteam iterativ immer weiter optimierten Lösungen annähert.

Der wirtschaftliche Nutzen generiert sich aus dem geringeren Aufwand für Fehler- & CR-Handling sowie aus verkürzten Release-Zyklen. Das agile Vorgehen vermeidet langwierige Abstimmungs- und Freigabeprozesse in frühen Projektphasen. Dazu kommt, dass Kosten generierende Fehler in Anforderungen oder im Design deutlich früher erkannt werden.

Notwendige Anpassungen

Auch wenn viele agile Methoden scheinbar perfekt in das Umfeld mobiler App-Entwicklung passen, empfehlen sich doch Anpassungen in drei Bereichen, um Spezifika der App-Entwicklung berücksichtigen zu können. Diese sollen im Folgenden erläutert werden:

Mehr Test im agilen Mobile App Entwicklungsteam

Die Rolle des Testers gibt es beispielsweise nach der "reinen" SCRUM-Methodik gar nicht. Im SCRUM-Modus sind allein der Product Owner und das Entwicklungsteam für die Sicherung der gesamten Produktqualität verantwortlich. In der Regel beherrschen aber weder Product Owner noch Entwickler den systematischen Testansatz, noch kennen sie die Anwendung verschiedener Teststufen.

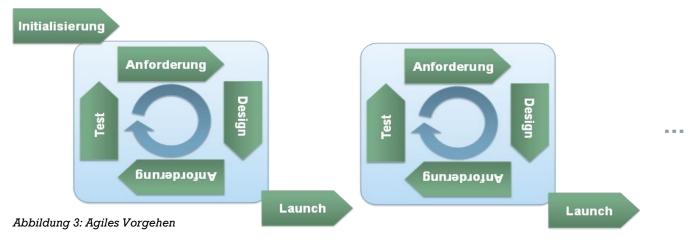
Gleichzeitig nimmt die Zahl der relevanten Qualitätseigenschaften zu (z.B. für Sicherheit, Zuverlässigkeit oder Gebrauchstauglichkeit) und die Komplexität der benötigten Infrastruktur erhöht sich. Es muss also eine Vielzahl – möglicherweise neuer – Testtechniken systematisch geplant und durchgeführt werden.

Daher ist die Etablierung von Sprint-Testern unabdingbar. Sie müssen die marktüblichen Teststandards, wie sie z.B. im ISTQB festgeschrieben sind, sicher beherrschen. Darüber hinaus müssen Tester wie Entwickler im Vorfeld den agilen Methodenkoffer vermittelt bekommen. Wünschenswert ist ebenfalls seine technische Affinität und Kenntnisse zu Entwicklungstechniken bis hin zur Automatisierung von Testfällen, um sehr eng mit den Entwicklern kooperieren zu können.

Mehr Integration in der Mobile App Entwicklung

Apps stellen zwar häufig eine eng begrenzte Funktionalität zur Verfügung, sind aber immer häufiger Teil einer komplexen Architektur, welche diverse Backendsysteme, Server, Web Services oder Cloud-Dienste nutzt und orchestriert.

Diese Integration erzwingt planerische Weitsicht, die insbesondere die Testplanung und das Release-Management betrifft.



Testen mobiler Apps / Buchbesprechung

Abhängigkeiten müssen bei der Planung genauso berücksichtigt werden wie andere Vorgehensmodelle. So werden Legacy-Dienste häufig eben nicht agil weiterentwickelt.

Gerade weil eine vielgestaltige Integration in SCRUM zunächst nicht vorgesehen ist, muss die Sprint-Planung um die o.g. Aspekte erweitert werden. Ein kompletter Entwicklungs-Lifecycle kann sich dann über ein bis zwei Jahre mit einer Reihe von Release-Zyklen hinziehen. Jedes Release setzt sich dann wiederum aus einer Anzahl von Sprints und einer komplexen Integration (oder Release-Test) zusammen, d.h. dass nicht nach jedem Sprint das Ergebnis zwingend live geht.

Mehr Tooleinsatz der agilen Entwicklung für Mobile Apps

Aufgrund der beschriebenen Vielzahl an möglichen Zielkanälen, - plattformen, und -geräten ergibt sich für den Tester eine gefährliche kombinatorische Explosion bezüglich Testfällen und Testumgebungen. Neben methodischen Erweiterungen des risikobasierten Testens bietet der Tooleinsatz die Möglichkeit, möglichst viele relevante Plattformen ausreichend mit Tests abzudecken. Dies widerspricht aber zunächst dem agilen Paradigma.

In der Praxis zeigt sich, dass sich ohne Werkzeugunterstützung im Bereich der Automatisierung der Testausführung sowie im Testumgebungsmanagement deutliche Effizienzverluste ergeben. Gerade in einem so jungen Markt wie den mobilen Apps sollten diese Tools jedoch sorgfältig ausgewählt werden, da es große Unterschiede in Bezug auf Aspekte wie Funktionalität oder Plattformunterstützung gibt.

Fazit

Agilität ist das präferierte Vorgehensmodell für die App-Entwicklung – doch erfordert die Umsetzung eine mehrschichtige Erweiterung des agilen Methodenkoffers hinsichtlich der speziellen mobilen Herausforderungen. Dies betrifft die stärkere Berücksichtigung von Testen und Testern als Rolle, der bewussten Integration mit den Entwicklungsvorgehen von Umsystemen und dem verstärkten Tooleinsatz als Mittel zur Verbesserung von Testeffektivität und -effizienz auf vielfältigen Plattformen und Geräten.

◆(Sven Euteneuer)

Literatur:

Agile.org. (2001). Manifest für Agile Softwareentwicklung. Von http://agilemanifesto.org/iso/de/abgerufen

Euteneuer, S. (19. 11 2012). SPL Testing for Efficient Mobile Testing. Von The 23rd CREST Open Workshop - Change Impact Analysis and Testing of Software Product Lines: http://crest.cs.ucl.ac.uk/cow/23/ abgerufen

IDC, Inc. (03. April 2013). PC Shipments Post the Steepest Decline Ever in a Single Quarter, According to IDC. Abgerufen am 03. April 2013 von http://www.idc.com/ getdoc.jsp? containerId=prUS24065413#.UWZrp1dtzpj

Wikipedia. (2013). Android Release Cycles. Von http://en.wikipedia.org/wiki/ Android_version_history abgerufen

Kontakt: info-austria@sqs.com

[Der Autor]

Sven Euteneuer hat Informatik studiert und ist bei der SQS AG als Senior Research Manager tätig. Dort ist er verantwortlich für das Innovieren



des Serviceportfolios in den Bereichen Mobile Computing und IT Security.

[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)

[Quergelesen]



Testen in Scrum-Projekten

Von Tilo Linz, dpunkt.verlag

Dieses empfehlenswerte Buch betrachtet agile Software-Entwicklung aus der Perspektive des Tests und des Software-Qualitäts-Managements und richtet sich sowohl an Testmanager und Tester als auch an alle anderen Rollen wie Projektmanager, Scrum Master, Product Owner, Entwickler.

Nach einer Übersicht über Scrum und Kanban und einem Vergleich mit den klassischen Vorgehensmodellen wird mit vielen praxisnahen Beispielen die Testplanung und das Vorgehen in den einzelnen Teststufen Unit Test, Integrationstest und Systemtest in agilen Projekten beschrieben. Typische agile Methoden wie Test First und Continuous Integration werden ausführlich beschrieben und ein Überblick über die Testmethoden in den einzelnen Teststufen gegeben.

Ausführlich wird auch auf die Rolle des Testmanagements in Scrum eingegangen. Es wird darauf hingewiesen, dass ein professioneller Softwaretester unbedingt Mitglied im Scrum-Team sein sollte, und auch durchaus "Testmanager" bezeichnet werden sollte.

Wichtig sind auch Hinweise zu Punkten, wo die reine agile Theorie an ihre Grenzen stößt, und wie man mit Themen wie nichtfunktionale Tests, langlaufende Tests und der Frage, wann der Systemtest eingeplant werden sollte, umgehen kann. Abgerundet wird das Thema durch Fallstudien aus der Praxis, in denen die Erfahrungen bei der Einführung von Scrum aus 5 Projekten wiedergegeben werden.

◀ (Karl Kemminger)

[ASQF-Fachgruppe Software-Test Österreich hebt ab]

von Mohsen Ekssir



Die ASQF-Fachgruppe Software-Test in Österreich wächst kontinuierlich. Der Arbeitskreis Software-Qualität und - Fortbildung e.V. (ASQF) ist das Kompetenznetzwerk für Softwarequalität, und organsiert als Fachverband im gesamten deutschsprachigen Raum kostenlose Fachveranstaltungen rund um folgende Themen:

- Software-Qualität
- Software-Test
- Requirements Engineering
- agile Methoden
- Software-Engineering
- Projektmanagement
- SOA
- Safety
- User Assistance
- und vieles mehr

Der ASQF ist ein Kompetenznetzwerk aus Global-Player-Unternehmen, leistungsstarken Mittelständlern, engagierten Fachleuten, Hochschulen und Forschungseinrichtungen, und ist in Österreich mit der Fachgruppe Software-Test stark vertreten.

Die Fachgruppe Software-Test Österreich versucht, durch die Organisation der Fachgruppenabende (vier Mal jährlich) einen kostenlosen Wissenstransfer in Bezug auf Software-Test und Qualitätssicherung zustande zu

bringen. Die Fachgruppe bietet eine Plattform für Tester, Studenten und Experten, um neue Themen, Ideen und Methoden, Praxiserfahrungen und - berichte, Diplomarbeiten usw. rund um das Thema Software-Test zu vermitteln und auszutauschen.

Der 15. Fachgruppenabend fand am 11. Juni 2013 an der Fachhochschule Technikum Wien statt. Der Fachgruppenabend rund um das Thema "Software Test für Embedded Systems" wurde von nahezu 40 Teilnehmern besucht. Der Referent, Experte Dr. Stephan Grünfelder, illustrierte den Vortrag praxisnah und ging anschließend auf die Fragen des Publikums ein.

Auch bereits die letzten Fachgruppenabende im März 2013 und November 2012 fanden regen Zulauf.

Neben aktuellen Fachvorträgen stehen auch Networking, Erfahrungsaustausch und das Zusammentreffen der Tester Community im Vordergrund.

Die Fachgruppe Software-Test erlebt in Österreich - ganz besonders in der letzten Zeit - einen Aufschwung. Die Anzahl der Personen- und Firmenmitglieder von ASQF in Österreich nimmt ständig zu und immer mehr Experten sowie Studenten nehmen an den Fachgruppenabenden teil und sorgen für qualitativ hochwertige Diskussionen.

An dieser Stelle möchte ich ganz besonders der Fachhochschule Technikum Wien, Institut Informatik, und Herrn Prof. Alexander Hofmann für die freundliche Unterstützung danken.

Falls Sie Interesse an einer Teilnahme oder einem Vortrag haben bzw. gerne nähere Informationen über die Fachgruppe und ihre Aktivitäten bekommen möchten, kontaktieren Sie mich bitte unter:

mohsen.ekssir-monfared@asqf.de.
Ich freue mich auf eine Zusammenarheit

Allgemeine Informationen und kommende Veranstaltungen sind zu entnehmen: https://www.asqf.de/

◀(Mohsen Ekssir)

Dr. Mohsen Ekssir leitet seit 2010 die ASQF-Fachgruppe Software Test Österreich. Er ist der Bereichsleiter Software-Test und Qualitätssicherung bei



der Firma BDC EDV-Consulting und unterrichtet Software-Qualitätsmanagement an der Fachhochschule Wiener Neustadt.

[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.

16.GF-Abend: Di., 24. September 2013, 18:00 bis 20:00 Fachhochschule Technikum Wien, HS1.04A, 1200 Wien, Höchstädtplatz 5



Thema: WebAccessiblity und Assistierende Technologien – Wie "sehen" und bedienen blinde und sehbehinderte Menschen Websites?

Der Vortrag zeigt den rechtlichen und organisatorischen Rahmen für das Thema Webaccessiblity in Österreich und im internationalen Umfeld auf und beantwortet die Frage, warum Accessiblity für Menschen mit Behinderung eine derart große Bedeutung zukommt. Es werden Werkzeuge und Methoden gezeigt, um Evaluierungen hinsichtlich der gültigen und vorhanden Regeln (WCAG 2.0 = ISO/IEC 40500, BITV 2.0 (Deutschland), Section 508 (USA)) durchführen zu können. Des weiteren wird ein Überblick über die vorhandenen und verwendeten AT (Assistierenden Technologien) am Markt, d.h. Screenreader, Magnifier bzw. Sprachausgabegeräte sowie Tools für Testing gegeben.

Mag. Klaus Höckner ist IT-Leiter und Accessibility Consultant bei der Hilfsgemeinschaft der Blinden und Sehschwachen Österreichs. Er hat an der Wirtschaftsuniversität Wien BWL (Schwerpunkt Wirtschaftsinformatik/Informationswirtschaft) studiert und ist Absolvent des Lehrgangs Barrierefreies Webdesign an der Kepler Universität Linz. Er beschäftigt sich seit mehr als zehn Jahren mit dem Thema Barrierefreiheit und IKT. Weitere Tätigkeitbereiche u.a. sind: Mitglied der ICT – Experts – Group des European Disability Forums EDF / Vorstandsmitglied beim Verein AccessibleMedia / Mitglied beim Austrian Standardisation Institute als Experte in Arbeitsgruppen, vor allem im Bereich "Information Transaction Machines".

Link zur Anmeldung

◀ (Mohsen Ekssir)

ATB Expertentreff

[ATB Expertentreff - Testautomatisierung im mobilen Umfeld]

von Karl Kemminger

"Test-Automatisierung im Mobilen Umfeld – Eine Bestandsaufnahme" - unter diesem Titel fand am 8. Mai die wieder sehr gut besuchte zweite Ausgabe des ATB Expertentreffs statt.

Zu Beginn gab es eine Neuigkeit - die Teilnehmer konnten selbst aktiv und mittels Punkten 4 Fragen bewerten

- Die Frage nach dem Stellenwert der Regressionstests beim Test mobiler Apps wurde überwiegend mit "Ja, sehr wichtig" beantwortet.
- Beim Einsatz der Testautomatisierung bei der Q-Sicherung mobiler Apps schieden sich die Geister: 50% setzen diese ein, und 50% halten diese für überwertet und nicht so wichtig. Niemand war allerdings der Meinung, dass das niemand braucht.
- Test von Securityaspekten wurde von praktisch allen Teilnehmern als sehr wichtig angesehen.
- Einsatz von mobile Geräten für Körper-Logging wird dagegen skeptisch betrachtet, und eher abgelehnt.

Danach gab Christoph Preschern einen Überblick über die verschiedenen Typen (native, browser-basiert und hybrid) sowie die spezifischen Herausforderungen bei Entwicklung und Test von mobilen Apps, und eine Livedemo einer Testautomatisierung mit den Tools von Ranorex.

In der anschließenden Diskussion herrschte Konsens darüber, dass die Smartphones in immer mehr Bereiche des Lebens eindringen werden, und dass Sicherheit und Datenschutz größere Bedeutung erlangen werden, und professionelles Testen in diesem Bereich essentiell ist.

Wie beim ersten Expertentreff gab es wieder viele Wortmeldungen und Fragen an die Referenten aus dem Publikum, bevor es beim abschließenden Buffet intensive Gelegenheit zum Networking gab.

Alle Unterlagen finden Sie unter http://www.austriantestingboard.at/expertentreff Test-Automatisierung im Mobilen Umfeld

Eine Rückschau auf das Expertentreff am 3.7.2013 zum Thema "Crowdsourced Software Testing" finden Sie in der nächsten Ausgabe des Insiders.

Eine Vorschau auf das Expertentreff am 4.9.2013 zum Thema "Requirement Spezifikation im agilen Umfeld" finden Sie auf Seite 1.

◀ (Karl Kemminger)

Merken Sie Sich gleich die nächsten Termine vor:

- Mittwoch, 4.9.2013
 Requirements-Spezifikation in agilen Projekten
- Mittwoch, 13.11.2013
 Thema noch offen



Das Tech-Gate war wieder gut besucht



Christoph Preschern gab einen guten Überblick über die Herausforderungen bei Entwicklung und Test mobiler Apps



Diskussionsrunde mit Thomas Nemec/Erste Bank, Gerhard Wiesinger/anovis und Christoph Preschern/Ranorex



Die Lebenshilfe spendete die Pokale, die den Referenten als Dank überreicht wurden (links Helmut Pichler, Präsident des ATB, rechts Alexander Weichselberger, der die Expertentreffs organisiert)

Trainingsprovider Ser

Kurs	Termin	Ort	Anbieter
ISTQB Certified Tester	2629.08.2013, <u>Anmeldung</u>	Wien	ANECON
Foundation Level	0205.09.2013, <u>Anmeldung</u>	Wien	sQs
	0205.09.2013, <u>Anmeldung</u>	Graz	Software Quality Lab
	0205.09.2013, <u>Anmeldung</u>	Linz	
	0205.09.2013, <u>Anmeldung</u>	Wien	
	0912.09.2013, <u>Anmeldung</u>	Wien	BDC
	2326.09.2013, <u>Anmeldung</u>	Lustenau	Software Quality Lab
	0407.11.2013, <u>Anmeldung</u>	Wien	sQs
	1114.11.2013, <u>Anmeldung</u>	Wien	ANECON
	1821.11.2013, <u>Anmeldung</u>	Wien	BDC
ISTQB Certified Tester Advanced Level	22.–26.07.2013, <u>Anmeldung</u>	Wien	Software Quality Lab
Test Manager	1117.09.2013, <u>Anmeldung</u>	Wien	ANECON
	1620.09.2013, <u>Anmeldung</u>	Wien	sQs
	0206.12.2013, <u>Anmeldung</u>	Wien	sQs
ISTQB Certified Tester Advanced Level	1619.09.2013, <u>Anmeldung</u>	Wien	Software Quality Lab
Test Analyst	1619.09.2013, <u>Anmeldung</u>	Linz	
	1418.10.2013, <u>Anmeldung</u>	Wien	sQs
ISTQB Certified Tester Advanced Level	1822.11.2013, <u>Anmeldung</u>	Wien	sǫs
Technical Test Analyst			
IREB Certified Professional for Requirements	16.–18.09.2013, <u>Anmeldung</u> mit Praxistag 16.–19.09.2013, <u>Anmeldung</u>	Graz	Software Quality Lab
Engineering Foundation Level	16.–18.09.2013, <u>Anmeldung</u> mit Praxistag 16.–19.09.2013, <u>Anmeldung</u>	Linz	
	16.–18.09.2013, <u>Anmeldung</u> mit Praxistag 16.–19.09.2013, <u>Anmeldung</u>	Wien	
	1820.09.2013, <u>Anmeldung</u>	Wien	ANECON
	2527.09.2013, <u>Anmeldung</u>	Wien	BDC
	2729.11.2013, <u>Anmeldung</u>	Wien	
	0406.12.2013, <u>Anmeldung</u>	Wien	ANECON
Certfied Agile Tester® Training (Prüfung in deutsch oder englisch)	0509.08.2013, <u>Anmeldung</u>	Wien	ANECON
, , , , , , , , , , , , , , , , , , ,	2327.09.2013, <u>Anmeldung</u>	Wien	sQs
	30.0904.10.2013, <u>Anmeldung</u>	Wien	ANECON

Partner — Veranstaltungen und Termine

Veranstaltung	Termin	Ort	Anbieter
TOSCA Certified User Foundation Level (TCUFL)	0911.07.2013, <u>Anmeldung</u>	Wien	TRICENTIS
	0305.09.2013, <u>Anmeldung</u>	Wien	
TOSCA Certified Quality Designer (TCQD)	1618.07.2013, <u>Anmeldung</u>	Wien	TRICENTIS
TOSCA Technical Training (TTT)	1012.09.2013, <u>Anmeldung</u>	Wien	TRICENTIS
Unit Testing	1719.09.2013, <u>Anmeldung</u>	Graz	Software Quality Lab
	1719.09.2013, <u>Anmeldung</u>	Linz	
	17.–19.09.2013, <u>Anmeldung</u>	Wien	
Agile Testing in a Nutshell	21.10.2013, <u>Anmeldung</u>	Wien	ANECON



Hier könnte Ihr Inserat stehen!

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

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

Konferenzen / iSQI / Silberpartner / Impressum

	[F	[Konferenz Kalender]			
Konferenz	Termin	Ort	Deadline Call for Paper		
ASOT	1920.09.2013	Graz			
Agile Testing Days	2831.10.2013	Potsdam			
OA & TEST	2931.10.2013	Bilbao			
EuroSTAR	0407.11.2013	Göteborg			
Software Quality Days 2014	1416.01.2014	Wien			

[Öffentliche ISTQB Prüfungstermine]

Die nächsten Termine der öffentlichen Prüfungen in Wien sind: 28.8., 30.10., 18.12.2013.



Für Anmeldungen und nähere Informationen wie Ort und Beginnzeiten kontaktieren Sie bitte die Homepage der iSOI.

TRICENTIS. SOLUTIONS Achieving Testing Excellence

Der 7. jährliche TRICENTIS Solutions Day bietet ein exzellentes Forum für Testexperten in allen Bereichen des Softwaretests, der Testautomatisierung und Qualitätssicherung.

Wann: Donnerstag, 7. November 2013 Austria Center Vienna, Wien 11 - 21 Uhr (inklusive Abendevent)

Mit über 250 Teilnehmern ist der Solutions Day eines der größten Events im Bereich funktionaler Softwaretest in Europa. Informieren Sie sich über aktuelle Trends und bewährte Methoden in den Vorträgen von A1 Telekom Austria. UPC Austria, Loyalty Partner Solutions, Wiener Gebietskrankenkasse und vielen

Nutzen Sie die Gelegenheit während unseres Abendevents mit Kollegen zu netzwerken und Erfahrungen auszutauschen.

5 Tracks mit 22 Sessions über:

Mobile, Agile, Test Automation, Excellence in Testing und ERP Testing

Besuchen Sie die Stände im Ausstellungsbereich und erfahren Sie von unseren Testexperten, wie Sie Softwaretests besser, schneller und einfacher machen können.

Sprechen Sie über Ihre Herausforderung im Test mit uns. Erfahren Sie, wie Sie Ihre Herausforderungen im Test meistern, Ihren Time-to-Market verbessern und gleichzeitig Zeit und Aufwand sparen können.

Begrenzte Plätze, registrieren Sie sich jetzt! Wir freuen uns Sie zu treffen.

www.solutionsday.at

Unsere Goldsponsoren









[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.

[Agile Entwicklungsmethoden in der Praxis Das Beste aus KANBAN und SCRUM

Weitere Entwicklungspotenziale nutzen durch Methodenhybriden "SCRUMBAN"]

Teil 1, von Torsten Zimmermann

Der Wind in der Softwarebranche wird rauer: Unternehmen müssen heute flexibler sein, Entwicklungsergebnisse schneller und in besserer Qualität liefern als jemals zuvor, wenn sie weiterhin erfolgreich am Markt operieren wollen. Natürlich zu geringeren Kosten. Dabei sollen die realisierten Anwendungen aber höheren Anforderungen gerecht werden. Entwicklungsteams finden sich deshalb oft in einem Spannungsfeld zwischen der Forderung nach schnellerer Zielerreichung und gleichzeitig steigendem Entwicklungsaufwand wieder: Agile, iterative Entwicklungsmethoden helfen, dieses Problem zu entschärfen, und verdrängen seit geraumer Zeit sequenzorientierte Entwicklungsansätze. Jedoch sind auch hier in der Praxis Limitierungen zu erkennen. Die Kombination aus KANBAN und SCRUM zeigt mögliche Lösungen auf und bringt Teams wie Unternehmen weiter.

Aufgrund der anspruchsvollen Vorgaben können Entwickler sich nur noch um die wichtigsten Aufgaben kümmern. In der Regel werden Vorbereitungs- und Testphasen gekürzt, um den vorgegebenen, zeitlichen Rahmen zu erfüllen. Dies kann zulasten einer stabilen Softwarearchitektur gehen: Die aktuellen Anforderungen sind zwar erfüllt, aber Erweiterungen mögen nur auf der Basis aufwendiger Änderungen an der Software-Architektur zu realisieren sein.

Enge Zeitpläne führen aber auch zu chronisch überlasteten Teams. Oft ist das den Auftraggebern nicht bewusst, da es hierzu in der Regel keine Informationen gibt. Damit sinkt langfristig die Team Performance. Viele Entwickler empfinden deshalb heute den "Burnout". Sie fühlen sich ausgelaugt und finden nicht mehr zur Ruhe. Hinzu kommt das Gefühl der Hilflosigkeit, wie steigende Anforderungen in immer kürzeren Zeiteinheiten realisiert werden sollen.

Es besteht Handlungsbedarf

Der Wind in der Softwarebranche wird rauer. Um hier weiterhin erfolgreich bestehen zu können, bedarf es eines hohen Maßes an Agilität und Ressourcenerhalt. Es gilt also, verschiedene Strategieziele nachhaltig zu realisieren:

Softwareunternehmen müssen sich rasch an Marktveränderungen anpassen können. Dabei sollte die Realisierung von Softwarelösungen in hoher Qualität in verhältnismäßig kurzer Zeit möglich sein. Beide Aussagen - Qualität und Zeit - können konträr zueinander stehen. Dies darf nicht dazu führen, dass kurze Entwicklungszeiten auf Kosten der Lösungsqualität gehen. Ein wichtiges, zentrales Thema, ist hierbei der Erhalt der Teamleistungsfähigkeit. Das Softwareunternehmen fördert durch hohe Prozesstransparenz die Eigenmotivation der Mitarbeiter. Ferner können die Teammitglieder selbst bei Engpässen rasch geeignete Gegenmaßnahmen einleiten. Die Gefahr ungeplanter Ereignisse, welche zu umfangreichen Änderungen an Projekt-/ Entwicklungsplänen führen, wird deutlich reduziert.

Diese Ziele sind teilweise gegenläufig und bedingen ein Austarieren. Das Optimum liegt also zwischen den Extremen. Hierzu bedarf es aber auch eines Methodenkonzeptes, welches in der Lage ist – durch entsprechende Anpassungen – allen Teilzielen zu entsprechen. Gerade hier scheitern viele klassische, aber auch agile Entwicklungsmethoden. Kann hier ein Hybrid, also eine Kombination verschiedener Methoden, geeigneter sein, den steigenden Anforderungen zu entsprechen als deren Einzelkomponenten?

Ausgehend von den vorgestellten Zielen, wären nachfolgende Entwicklungsmethoden geeignet:

 SCRUM: SCRUM unterstützt den agilen Prozesscharakter. Es unterstützt die Agilität und Planungssicherheit.

- KAIZEN: Mit KAIZEN werden Entscheidungsprozesse optimiert. Entscheidungen lassen sich vergleichsweise rasch treffen, da nach KAIZEN nur noch die Informationen berücksichtigt werden, welche tatsächlich entscheidungsrelevant sind.
- KANBAN: Mit KANBAN ist es möglich, mit vergleichsweise einfachen Mitteln eine hohe Prozesstransparenz zu realisieren und die Teams vor Überlastung zu schützen.

Wenn man also diese drei Ansätze geschickt zu einem Methodenhybriden kombiniert, so werden alle vier Teilziele berücksichtigt. Ich möchte die verschiedenen Methodenansätze kurz vorstellen.

Eine kleine Methoden-Übersicht

Es gibt inzwischen eine ganze Reihe von Entwicklungsmethoden, welche in der Softwareindustrie verwendet werden. Ein entscheidender Unterschied zwischen den Methoden ist der Umfang des betreffenden Regelwerks. Generell gilt: Je mächtiger das Regelwerk, desto geringer die Anpassungsfähigkeit. Oder: Methoden mit einfachen und wenigen Regeln sind in der Regel sehr flexibel auf das konkrete Umfeld anpassbar. Dieser Sachverhalt verhalf SCRUM dazu, sich schnell als Entwicklungsmethode in der Softwarebranche zu etablieren. KAN-BAN ist in diesem Punkt SCRUM ähnlich: Dessen Regelsatz ist jedoch im Vergleich geringer, und die Anpassungsfähigkeit der Methode ist im Vergleich zu SCRUM ausgeprägter.

Oft hört man die Frage, welche Methode nun die beste sei. Eigentlich eine unsinnige Frage, denn: Jede der in der Grafik dargestellten Methode hat sowohl Stärken als auch Schwächen. Wichtig ist es letztlich, die Methode zu wählen, welche am besten zu dem konkreten Umfeld passt und die vorherrschen Anforderungen optimal unterstützt. Damit eine sinnvolle Methodenselektion möglich ist, sollten hierzu nicht nur Methodenkenntnisse vorhanden sein, sondern auch konkrete Erfahrungen bezüglich der Methodenanwendung aus Entwicklungsprojekten.

SCRUM

SCRUM (engl. "Gedränge") ist ein Rahmenwerk zur Entwicklung komplexer Produkte. Der Ansatz von SCRUM ist empirisch und zeichnet sich durch eine einfache Struktur aus. Er beruht auf der Einsicht, dass die meisten modernen Entwicklungsprojekte zu komplex sind, um durchgängig planbar zu sein. SCRUM versucht, die Komplexität durch drei Prinzipien zu reduzieren:

- Transparenz: Alle relevanten Informationen eines Projektes werden täglich und für alle sichtbar festgehalten.
- Überprüfung: Regelmäßig werden neu entwickelte oder erweiterte Produktfunktionalitäten vorgestellt und beurteilt.
- Anpassung: Die Produktanforderungen werden nicht endgültig fixiert, sondern können nach jeder Lieferung neu bewertet und angepasst werden.

KAIZEN

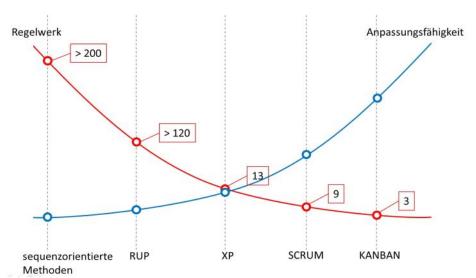
KAIZEN (japanisch KAI = Veränderung, Wandel; ZEN = zum Besseren, also "Veränderung zum Besseren") bezeichnet eine japanische Lebensund Arbeitsphilosophie, in deren Zentrum das Streben nach ständiger Verbesserung steht.

In der Wirtschaft wurde das Konzept zu einem Managementsystem weiterentwickelt. In der Praxis werden die Begriffe "Kontinuierlicher Verbesserungsprozess" (KVP) und "Continuous Improvement Process" (CIP) synonym verwendet.



Abb. 2: Das Wort KAIZEN in japanischer Schrift

Abb. 1: Methodenübersicht



KAIZEN ist jedoch mehr als ein reiner Qualitätsbegriff. In Japan steht KAIZEN für die Philosophie der "ewigen Veränderung". Mit dieser Philosophie werden wir ständig konfrontiert, indem Produkte zum Beispiel Monat für Monat mit weiteren Add-ons versorgt werden. Aber auch die Tatsache, dass viele Produkte der Elektronikindustrie stetig kürzeren Innovationszyklen unterliegen, ist ein Ergebnis von KAIZEN.

Wichtiger Baustein im Rahmen agiler Entwicklungsmetoden können jedoch die 5S-Bewegungen sein. Dabei handelt es sich um eine fünfstufige Vorgehensweise zur Neuplanung und Verbesserung sauberer, sicherer und standardisierter Arbeitsplätze.

- Seiri ⇒ Entferne Unnötiges aus deinem Arbeitsbereich!
- Seiton ⇒ Ordne die Dinge, die nach Seiri geblieben sind!
- Seiso ⇒ Halte deinen Arbeitsplatz sauber!
- Seiketsu ⇒ Mache Sauberkeit und Ordnung zu deinem persönlichen Anliegen!
- Shitsuke ⇒ Mache 5S durch Festlegen von Standards zur Gewohnheit!

Wie man gut erkennt, steht in KAI-ZEN der Mitarbeiter im Mittelpunkt. Er bestimmt letztlich den Erfolg. Auf der Basis dieser Erkenntnis entstanden auch SCRUM und viele andere agile Methoden. KAIZEN ist aus drei Gründen gut für eine Kombination mit agilen Methoden geeignet:

- KAIZEN kann gut mit agilen Methoden kombiniert werden, da es kompatibel zu agilen Ansätzen ist. Der KAIZEN-Ansatz der "ewigen Veränderung" korreliert mit der Vorstellung aus der agilen Welt, dass ein komplexes Projekt gar nicht bis ins letzte Detail geplant werden kann: Während der Projektlaufzeit reifen neue Erkenntnisse zum Beispiel durch technologische Veränderungen oder durch neu entwickelte Verfahren, welche eine neue Planung bedingen.
- KAIZEN liefert einen Rahmen in Form der 5S als Mikromanagement für das einzelne Teammitglied im Bereich der Selbstorganisation. Dieses Mikromanagement stellt eine gute Ergänzung zu agilen Methoden, wie zum Beispiel SCRUM, dar.
- KAIZEN unterstützt die Reduktion unnötiger Abläufe und Informationen, welche weder für Entscheidungsprozesse oder den Projekterfolg von Bedeutung sind. Es verschlankt so die Verwaltung und Administration, was die Zielerfüllung agiler Methoden unterstützt.

KANBAN

KANBAN ist japanisch und bedeutet so viel wie Anschlagtafel oder Schild. Es stellt eine Methode zur Produktionsablaufsteuerung dar. entwickelte Taiichi Ohno in der japanischen Toyota Motor Corporation diesen Produktionssteuerungsansatz: Die ungenügende Produktivität des Unternehmens im Vergleich zu amerikanischen Konkurrenten war damals der Auslöser für diese Entwicklungsmaßnahme. Ferner wurden die Kosten für Zwischenlager in Japan immer teurer, da auf den japanischen Inseln das Raum-Angebot limitiert ist. Es galt also, einen Steuerungsansatz zu finden, welcher

- präzise ohne komplexe Berechnungen Produktionsabläufe steuert,
- hilft, Optimierungspotenziale im Produktionsprozess ohne aufwendige Analysen zu erkennen
- 3. und geeignet ist, den Bedarf an Lagerflächen zu reduzieren.

Ohno holte sich seine Motivation für KANBAN aus dem Supermarkt-Prinzip: Der Kunde nimmt sich das gewünschte Produkt aus dem Supermarktregal. Die herausgenommene Menge stellt hierbei ein Signal dar, Ersatz zu produzieren, um die entstandene Lücke wieder zu schließen.

Zentraler Bestandteil dieses Ansatzes ist das Hol- oder Zurufprinzip, welches in der Fachwelt auch "Pull-Prinzip" genannt wird. KANBAN orientiert sich demnach ausschließlich am tatsächlichen Verbrauch von Materialien am Bereitstell- und Verbauort. Dieser Ansatz erlaubt somit eine Reduzierung von Lagerbeständen bezüglich Produkten, Bauteilen oder Rohstoffen, welche in der Produktion benötigt werden. Somit werden in der Produktion Kosten aufgrund reduzierter Kapitalbindung durch kleinere Zwischenlager optimiert.

In der IT/Softwareentwicklung wird unter dem Namen KANBAN eine bestimmte Methode verstanden, die zwar Prinzipien aus dem ursprünglichen KANBAN übernimmt, sich aber in vielen Punkten stark von diesem unterscheidet. Dies liegt eigentlich auf der Hand: In der Softwareentwicklung stehen Artefakte im Produktionsfokus, welche immateriell sind und damit nicht den gleichen Gesetzmäßigkeiten unterliegen, wie sie auf einer Produktionsstraße - wie zum Beispiel im Automobilbau - oder in deren Produktionsergebnissen vorzufinden sind. So spricht man hier auch gerne von "Software-KANBAN" oder "IT-KANBAN", um diesen Unterschied bereits über die Methoden-Bezeichnung deutlich zu machen. Es lassen sich auch nicht einzelne KANBAN-Techniken aus der Produktionswelt in die IT-Welt ohne Anpassungen und Modifizierungen übernehmen. Software-KANBAN beinhaltet auch weitere Prinzipien aus der LEAN PRODUCTION (Schlanke Produktion), dem LEAN DEVELOPMENT (Schlanker Entwicklung), der THEORY OF CONSTRAINTS (Engpass-Theorie) und dem Risiko-Management. Es mag nun die Frage aufkommen, ob die Bezeichnung KANBAN für diesen Ansatz noch gerechtfertigt sei. Ich denke schon, da trotz der vielen Anpassungen die Methode als KANBAN-Derivat zum ursprünglichen KANBAN noch gut zu erkennen ist. 2007 stellte David Anderson Software-KANBAN 2007 erstmalig vor und führte die Methode erfolgreich bei Microsoft ein. Seit dieser Zeit steigt das Interesse an diesem Ansatz. Dies mag auch an nachfolgenden Vorteilen liegen:

- Software-KANBAN schafft schnell eine hohe Transparenz über Projektfortschritt und akute Probleme.
- Der Ansatz geht über die reine Softwareentwicklung hinaus und beleuchtet auch Bereiche wie die Qualitätssicherung, Wartung und Systemadministration.
- Er ist gut geeignet für Organisationen mit starker Arbeitsteilung und Spezialisierung.

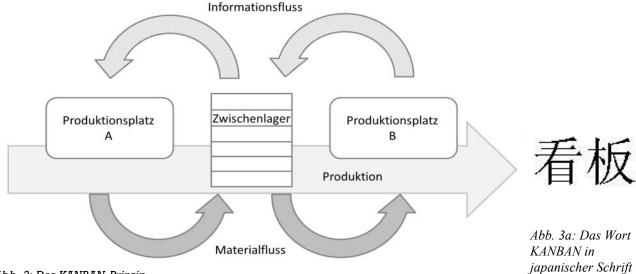


Abb. 3: Das KANBAN-Prinzip

- Der Ansatz lässt sich behutsam in bestehende, sequenzorientierte oder agile Prozesslandschaften integrieren und schrittweise anpassen
- KANBAN f\u00f6rdert eine gleichm\u00e4\u00e4sige und nachhaltige Geschwindigkeit (Sustainable Pace).

Die Grundidee von Software-KANBAN besteht in zwei Dingen:

Die Wertschöpfungskette – der KAN-BAN-Stream – mit ihren verschiedenen Prozessschritten – wie zum Beispiel Anforderungsdefinition, Programmierung, Dokumentation, Softwaretest und Inbetriebnahme – wird gut sichtbar für alle Beteiligten visualisiert. Dafür wird ein KANBAN-Board verwendet, auf dem die unterschiedlichen Stationen als Spalten dargestellt werden.

Warum KANBAN?

Welche Gründe kann es für Unternehmen geben, Software-KANBAN einzuführen? Die nachfolgende Rangliste enthält die zehn am häufigsten genannten Antworten:

- "Empowerment" der betreffenden Teams
- Verbesserung des agilen Momentums in dem betreffenden Bereich
- Reduktion der Verwaltungs- und Steuerungsaufwände
- Verbesserung der Realisierungsleistung/schnellerer "Burndown"
- Limitierung der parallel durchführbaren Arbeitspakete
- 6. Umsetzung des Pull-Ansatzes
- Rasche Problemerkennung und Lösung
- 8. Realisierung einer besseren Prozesstransparenz
- Reduzierung von Überstunden innerhalb der Teams
- Reduzierung des Bedarfes an externer Unterstützung (Insourcing)

- Die einzelnen Anforderungen (es können Tasks, Features, User Storys, Minimal Marketable Features (MMF) usw. sein) werden auf Karteikarten festgehalten und durchwandern mit der Zeit als Tickets das KANBAN-Board von links nach rechts.
- 2. Die Anzahl der Tickets, die gleichzeitig an einer Station bearbeitet werden dürfen, ist der Work in Progress (WIP), welcher für jede Station limitiert ist. Wenn beispielsweise die Anforderungsdefinition gerade drei Tickets bearbeitet und das Limit für diese Station ebenfalls drei beträgt, darf sie kein viertes Ticket annehmen, auch wenn weitere Aufgaben vorhanden sind. Hierdurch entsteht ein Pull-System, bei dem sich jede Station ihre Arbeit bei der Vorgängerstation abholt, anstatt fertige Arbeit einfach an die nächste Station zu übergeben. Dies bedeutet auch, dass die Anforderungsdefinition sich keine neue Aufgabe holen kann, wenn bereits eines der drei besagten Tickets fertig ist, die Programmierung aber nicht in der Lage ist, das Ticket abzuholen.

Die Visualisierung und die Begrenzung des WiP bieten einige Vorteile:

- 1. Hierdurch wird rasch sichtbar, wie schnell die Tickets die verschiedenen Stationen durchlaufen und wo sich Tickets stauen. Die Stationen, an deren Eingängen sich Tickets häufen, werden oft auch "Bottlenecks" genannt. Einfache Analysen des KANBAN-Boards lassen diese Bottlenecks schnell erkennen, und so können zeitnah Gegenmaßnahmen ergriffen werden, um einen möglichst gleichmäßigen Fluss (Flow) zu erreichen. Denn ein gleichmäßiger Durchfluss erzeugt einen nachhaltigen Stream, wobei das Verhältnis Ressourceneinsatz zu Durchlaufzeit der Tickets über alle Stationen in der Regel optimal im Vergleich zu einem ungleichmäßigen Fluss ist.
- 2. Die WiP-Limitierung hilft den Mitarbeitern an den Stationen, sich auf die aktuelle Arbeit zu konzentrieren.
 Ferner wird sehr schnell sichtbar, wenn das Team überlastet ist, das heißt, wenn das WiP-Limit viel zu hoch angesetzt wurde. Permanente Staus wären die Folge, sodass letztlich der gesamte KANBAN-Stream stoppen würde.

Eingang WIP: 2	Entwicklung WIP: 4		Qualitätssicherung WIP: 2		Transition WIP: 2			
	in Arbeit	erledigt	in Arbeit	erledigt	in Arbeit	erledigt	}	Produkt A
							}	Produkt B

Abb. 4: Beispiel eines KANBAN-Board

Ticketstau im KANBAN-Stream

Bei Software-KANBAN kann es zu Ticket-Staus kommen. Diese gilt es rasch zu beseitigen, um den Flow aufrecht zu erhalten oder wieder anlaufen zu lassen. Gründe für derartige Staus können zum Beispiel sein:

- Kurzfristiger Ausfall von Mitarbeitern (Krankheit)
- Unerwartete Probleme bei der Aufgabendurchführung
- Unklare Aufgabenbeschreibung
 Geeignete Gegenmaßnahmen

Es gibt jedoch eine Reihe möglicher Gegenmaßnahmen, um den Stau rasch zu beseitigen. Die hier vorgestellten Maßnahmen sind nicht gleichermaßen für alle möglichen Ursachen geeignet. Zur exakten Ursachenbestimmung und bei der Prozessoptimierung unterstützen das ISHIKAWA-Modell wie auch der KAIZEN-Ansatz. Die Gegenmaßnahmen sind der Ursache entsprechend auszuwählen, um möglichst rasch und effektiv den Stau zu beseitigen.

- Erhöhung der WIP-Limits einer Station
- Zuordnung weiterer Mitarbeiter zu einer Station (temporär oder permanent)
- Optimierung der begleitenden Dokumentation im Sinne von KAIZEN
- Optimierung der Eingangsprüfung

Nächste Ausgabe folgt der nächste Teil

"SCRUM und Software-KANBAN im Vergleich" sowie

die Vorstellung des Hybrids "SCRUMBAN"

[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

[Cartoon]

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

BLOOD PRESSURE & THE SDLC Tester's blood pressure varies through the development life cycle EARLY IN THE LIFE CYCLE: LOVELY I'VE JUST FOUND A SOMEWHERE IN THE MIDDLE: NOT ANOTHER BUG, I'M SURE THIS WAS FIXED IN THE LAST BUILD JUST BEFORE THE RELEASE DATE: OH NOI A BUGI I CAN'T STRESS BELIEVE IT! THIS IS GOING TO LOOK BAD SPIKE Andy Glover cartoontester.blogspot.com Co