

SEQIS Quality News

Requirements Engineering & Software Test



Ausgabe Q4/2015

Von Anforderung an

Anforderungsmanagement und Testing –
eine starke Beziehung

Spotlight:
Interview with 4 key
players at CETREL S.A.

Seite 5

Anforderungen gibt es
nicht:
Oder etwa doch?

Seite 10

SEQISenf:
Anforderungsmanage-
ment meets Software
Test: Erfolgsfaktoren

Seite 19

Titel: „Peppi“, Künstlerin: Lisa Kratky, Technik: Acryl auf Leinwand

LISA

aktuell. innovativ. praxisbezogen.

Lesen Sie in dieser Ausgabe:

Editorial.....3

Schwerpunkt
Test

Neulich im Netz.....4
Und wie testen Sie?

Schwerpunkt
Business Analyse

Wie viel Anforderung ist nötig?.....5
Eine Diskussion unter den SEQIS Experten

Schwerpunkt
Business Analyse

Spotlight.....7
Interview with 4 key players at CETREL S.A.: Daniel von Aesch, Heinz Saria, Alain Barthelemy und Renaud Dechambre

Anforderungen gibt es nicht.....10
Oder etwa doch?

SEQIS „10 things“.....13

Expertentreff #3/2015: Pair Analyse

SEQIS Kalender.....14

Alle Termine auf einen Blick in unserer Seminar- und Veranstaltungsübersicht

Gewinnspiel.....16

Lösen Sie unsere Gewinnfrage und gewinnen Sie einen Gutschein für 2 Personen von „Time Busters“!

Auflösung Gewinnspiel Q3....18

Finden Sie hier die Auflösung des Gewinnspiels der letzten Ausgabe

SEQISenf.....19

Anforderungsmanagement meets Software Test: Erfolgsfaktoren

Über SEQIS QualityNews:

Dieses Magazin richtet sich an Gleichgesinnte aus den Bereichen Softwaretest, Business Analyse/Requirements Engineering und Projektmanagement im IT-Umfeld.

Die Experten von SEQIS QualityNews berichten über ihre Erfahrungen zu aktuellen Themen in der Branche. Die Interessenten des Magazins gestalten die Ausgaben mit: Schreiben Sie uns Ihre Meinung im SEQIS Blog (blog.SEQIS.com) oder als Leserbrief. Wenn Sie dieses Magazin abbestellen möchten senden Sie bitte ein Mail an marketing@SEQIS.com.

Impressum:

Information und Offenlegung gem. §5 E-Commerce-Gesetz und §25 Mediengesetz

Herausgeber: SEQIS Software Testing GmbH, Neusiedler Straße 36, A-2340 Mödling

Tel: +43 2236 320 320 0

Fax: +43 2236 320 320 350

info@SEQIS.com, www.SEQIS.com

Gericht: Bezirksgericht Mödling

Firmenbuchnummer: 204918a

Umsatzsteuer-ID: ATU51140607

Geschäftsführung: Mag. Alexander Vukovic, Mag. Alexander Weichselberger, DI Reinhart Salomon

Druck: druck.at Druck- und Handelsgesellschaft mbH, 2544 Leobersdorf

Erscheinungsweise: 4x pro Jahr

Für die verwendeten Bilder und Grafiken liegen die Rechte für die Nutzung und Veröffentlichung in dieser Ausgabe vor. Die veröffentlichten Beiträge, Bilder und Grafiken sind urheberrechtlich geschützt. (Kunstwerke: Lebenshilfe Baden und Mödling, Fotos: ©fotolia.com). Sämtliche in diesem Magazin zur Verfügung gestellten Informationen und Erklärungen geben die Meinung des jeweiligen Autors wieder und sind unverbindlich.

Irrtümer oder Druckfehler sind vorbehalten.

Hinweis im Sinne des Gleichbehandlungsgesetzes: Aus Gründen der leichteren Lesbarkeit wird die geschlechtsspezifische Differenzierung nicht durchgehend berücksichtigt. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung für beide Geschlechter.

Ihre Meinung ist gefragt!

Nach den QualityNews ist bekanntlich vor den QualityNews! Schon bald arbeiten wir wieder auf Hochtouren an der nächsten spannenden Ausgabe. Lesen Sie nur das, was Sie wirklich interessiert! Sagen Sie uns, welche Themen Sie spannend finden.

Kontaktieren Sie uns: marketing@SEQIS.com
Join us: Facebook.com/SoftwareTestIsCool

Wir freuen uns auf Ihre Vorschläge und Wünsche!



Sehr geehrte Leserin,
sehr geehrter Leser,

es ist wieder soweit: Eine druckfrische Ausgabe der SEQIS QualityNews ist da! Auch dieses Mal finden Sie wieder zahlreiche interessante Artikel rund um unsere Spezialbereiche Software-test und Requirements Engineering.

In dieser Ausgabe dreht sich alles rund um die Kombination dieser beiden Disziplinen. Unsere Experten widmen sich der Frage, wie Anforderungsmanagement und Testing zu einander stehen, welche Berührungspunkte es zwischen diesen beiden Bereichen gibt und wie sie erfolgreich kombiniert werden können.

Wie gewohnt stellen wir auch dieses Mal unseren beiden Kernbereichen wieder einen nicht-technischen Aspekt zur Seite: Ein weiteres Mal spielt die Kunst in unserer Ausgabe eine entscheidende Rolle. Auf den folgenden Seiten finden Sie einige Werke der Lebenshilfe Niederösterreich der Werkstätten Baden und Mödling.

Sie fragen sich, wie diese einzigartigen Kunstwerke in ein IT-Fachmagazin

passen? Nun, unsere Spezialisten und die Klienten der Lebenshilfe verfügen über viele Gemeinsamkeiten. Beide Gruppen leben für ihre(n) Beruf(ung) und sind mit Leib und Seele dabei.

Täglich arbeiten wir mit Anforderungen – sie stellen den Ausgangspunkt für jedes unserer Projekte dar und sind für uns selbstverständlich.

Auch für die Klienten der Lebenshilfe spielen Anforderungen täglich eine entscheidende Rolle – wenn auch aus einem anderen Blickwinkel. Auch sie können ihre Tätigkeit nur dann erfolgreich ausführen, wenn bestimmte Anforderungen gegeben sind und mit ihren speziellen Bedürfnissen übereinstimmen.

Auf diesem Weg wünschen wir Ihnen alles Gute und einen erfolgreichen Start in das neue Jahr!

Viel Vergnügen beim Schmöckern in den SEQIS QualityNews! Über Feedback und Anregungen Ihrerseits freuen wir uns.

Ihre SEQIS Geschäftsleitung



Mag. Alexander Vukovic



Mag. Alexander Weichselberger



DI Reinhard Salomon



Titel: „Leopold“, Künstler: Max Minichmayr, Technik: Aquarell auf Papier

Und wie testen Sie?

von Hansjörg Münster



Hansjörg Münster ist Principal Consultant und Teamleiter bei SEQIS.

Als Allrounder deckt er ein breites Spektrum an Aufgaben ab. Die Schwerpunkte seiner Tätigkeit liegen in den Bereichen Testmanagement, Testautomation und Lasttest.

Ganz oben auf der Prioritätenliste des IT-Profis steht einen Nutzen und Mehrwert in der Qualitätssicherung seiner IT-Projekte zu generieren.

Im Frühjahr 2011 starteten drei deutsche Fachhochschulen eine Umfrage zum Thema „Softwaretest in der Praxis“ und erstellten aus den Ergebnissen eine umfangreiche Studie mit gleichnamigem Titel (siehe: www.softwaretest-umfrage.de/2011/Broschüre_Softwaretest_in_der_Praxis.pdf). Im Jahr 2013 erschien dazu passend noch eine zweite Publikation: Der „Technische Report“ zur Umfrage aus 2011 enthält noch mehr Detailinformationen und geht auf den aktuellen Stand der Qualitäts- und Testaktivitäten in der Praxis ein (siehe: www.softwaretest-umfrage.de/2011/TechnischerReport130317.pdf)

Mehr als 1.600 Personen nahmen an der Umfrage teil, allerdings beendeten diese nur etwas mehr als die Hälfte aller Teilnehmer. Dies lag wohl auch daran, dass ca. 100 Fragen zu beantworten waren. Dennoch lieferte die

Umfrage wichtige Ergebnisse, um den Handlungsbedarf im Testen für Forschung, Ausbildung und Beratung abzuleiten.

Und jetzt ist es wieder soweit: Unter der wissenschaftlichen Leitung der drei Fachhochschulen Bremen, Bremerhaven und Köln wurde eine Neuauflage dieser Studie erstellt und eine erneute Online-Umfrage zu Entwicklungen und Trends im Bereich Testen und Qualitätssicherung gestartet. Bis 15. Dezember 2015 hatten alle Interessierten die Möglichkeit, an der Umfrage teilzunehmen. Die Fragebögen und weitergehende Informationen sind auf der Webseite der Umfrage zu finden: www.softwaretest-umfrage.de

Neu hinzu kam der Fragekomplex „Forschung“, in welchem gezielt auf Ausbildungsmöglichkeiten und Forschungsschwerpunkte eingegangen wird. Darüber hinaus wurden die Projektorganisation, Informationen zum Tätigkeitsschwerpunkt des Unternehmens, der Umgang mit neuen Herausforderungen sowie Kennzahlen aus dem Bereich Qualitätsmanagement abgefragt. Allerdings wurde die Gesamtzahl der Fragen deutlich reduziert.

Es gab unterschiedliche Fragebögen für drei Zielgruppen: Einer richtet sich an das Management, einer an operativ tätige Personen (z.B. Projektleiter, Tester, Business Analysten, Entwickler, etc.) und einer sprach gezielt Forscher aus Hochschulen, öffentlichen Forschungsinstituten oder Forschungsabteilungen von Unternehmen an.

Interessant wird der Ergebnisvergleich der beiden Umfragen – daraus werden sich vermutlich viele Schlüsse ziehen lassen: Wie gestaltete

sich die Entwicklung bei unterschiedlichen Vorgehensmodellen? Wie verbreitet sind agile Methoden in der Zwischenzeit? Welche Trends lassen sich bei der Testautomation erkennen? Wie betrachtet man heute den Nutzen im Vergleich zu 2011?

Für die Forschung und Lehre ergeben sich daraus sicherlich spannende, neue Schwerpunkte für die nächsten Jahre. Denn gerade die Ausbildung unserer jungen Kollegen muss mit den Entwicklungen in der Wirtschaft und der gelebten Praxis mithalten.

Insofern ist es klar, dass eine Reihe von Institutionen diese Studie unterstützen und förderten. Auch das Austrian Testing Board (ATB) ist mit dabei und forderte seine Mitglieder (und auch Nichtmitglieder) aktiv zur Teilnahme auf.

Schon jetzt gibt es für alle Early Birds die Möglichkeit, sich zu registrieren und somit frühzeitig Zugriff auf die Ergebnisse, die im Frühjahr 2016 vorliegen sollen, zu erhalten. Die Ergebnisse werden nach der Auswertung per Email an alle Registrierten versendet.

Meine Tipp daher an alle Leser: Nehmen Sie sich ein paar Minuten Zeit, sehen Sie sich die Infos zur Umfrage an und klicken Sie im Frühjahr in die Ergebnisse rein! ■

Wie viel Anforderung ist nötig? Eine Expertendiskussion.

von Vinzenz Preiss

Fachdiskussion unter Business Analysten: Bei SEQIS beschäftigen sich täglich Experten mit der Tätigkeit der Anforderungsanalyse in Kundenprojekten. Aus der Erfahrung zeigt sich, dass die elementare Grundlage dieser Kompetenz – das Anforderungsdokument selbst – großes Diskussionspotential hinsichtlich Aussage und Wirkung auf den gesamten Entwicklungsprozess in sich birgt.

Das Schreiben einer Anforderung, Implementierung dieser in Programmcode und testen der korrekten Funktionsweise klingt wie die unausgesprochene Selbstverständlichkeit im täglichen Lauf eines Softwareentwicklungsprojekts. Die Erfahrung zeigt jedoch, dass diese vermeintliche Selbstverständlichkeit oft nicht von Beginn an gegeben ist. Speziell auf größerem Detaillevel und in komplexeren technischen und sozialen Systemen zeigt sich, dass persönliche Auffassung und Einstellung der einzelnen Rollen rund um die Softwareentwicklung erst sorgfältig aufeinander eingestellt werden müssen, ehe ein Verständnis vorliegt, das sich zumindest in mancher Hinsicht wie von selbst ergibt.

Definition ist Definitionssache

Aus Sicht des Anforderungsmanagements beruft man sich gerne auf eine sehr einfache aber auch eindeutige Faustregel um klarzustellen, was denn überhaupt durch eine Anforderung definiert werden soll.

Diese Regel besagt, dass aus der Analyse eine schriftliche Vorgabe hervorgeht und was umgesetzt werden soll – während sich die Implementierung damit beschäftigt, wie diese Anforderung umgesetzt wird.

Unsere Experten sind der Meinung, dass diese Faustregel eine anschauliche Erklärung darstellt, in der Praxis jedoch selten von brauchbarem Nutzen ist. Denn selbst festzulegen, wie sich die Frage „Was ist umzusetzen?“ überhaupt abgrenzt, ist eine nicht lösbare Herausforderung.

Stellen Sie sich vor, der Business Analyst erstellt ein Anforderungsdokument für einen Online-Shop. Bei oberflächlicher Betrachtung meint man, die Vorgabe, welche Produkte angeboten werden können und wie der Benutzer mit diesen interagieren kann, ist genau das, was zu entwickeln ist.



Vinzenz Preiss ist Test Consultant und Requirements Engineer bei SEQIS.

Mit seinen umfangreichen, breit gefächerten Kenntnissen und Kompetenzen unterstützt er Kunden in zahlreichen Projekten. Von der detaillierten technischen Schnittstellendefinition bis hin zur übergreifenden organisatorischen Prozessoptimierung kann seine Fachkenntnisse optimal einsetzen.

In seiner Rolle als Requirements Engineer findet er sich als kommunikativer, freundlicher Typ in jeder Situation schnell zurecht. Seine Spezialgebiete sind die Konzeption, Design- und Anforderungserhebungen von Teilbereichen unterschiedlicher Systeme.

Aus Sicht des Entwicklers kann Information auf solch grobgrangularer Ebene meist nicht zu einem zufriedenstellenden Ergebnis verarbeitet werden. Dieser benötigt da schon weitaus detailliertere Rahmenbedingungen, wie z.B. ob eine Einfach-Selektion (Radio-Button) oder eine Mehrfach-Selektion (Checkbox) angewandt werden soll.

Für den Entwickler gibt auch diese Information noch keine Auskunft darüber, was er überhaupt tun soll. Für manch anderen handelt es sich dabei schon um die detaillierte technische Ausarbeitung, wie die gegebene Anforderung umgesetzt wird.

Ich sehe da aber zwei Probleme: es gibt nicht nur eine "orientierende Lösung" für ein bestimmtes Problem; und da die Meinungen darüber, was eine "orientierende Lösung" genau häufig auseinander liegen, wird vielerorts sagen, das ist eine orientierende Lösung, weil sie nur erläutert und programmiert wurde - und der Analytiker könnte sagen: ja okay, es macht aber nicht das, was der Kunde braucht, ich würde also sagen, die "verpflichtende Interesse an einer orientierenden Lösung" ist eine notwendige, aber nicht hinreichende Bedingung.
vor etwa einem Monat · Gefällt mir · gefällt mir

Ich habe schon Recht, Kreka Ikonoholismus zeigt die Praxis dass oft schon bei Grundlegendem Probleme auftreten. Ich unterschreibe auch sofort, dass die "orientierende Lösung" wieder hinreichend ausreicht um direkt existieren wird. Doch genau an dieser Stelle schließt sich für mich der Kreis beim Tester: der ja durch die Anforderung einen Rahmen bekommt, den er sicherstellen muss. Die Anforderung hat dadurch natürlich den Hebel, mehr oder weniger Freiheiten zu lassen und somit das Ergebnis direkt zu beeinflussen. Die Frage ist, ob Persönlichkeiten allein die Form von Artikulation und Art und Weise damit umzugehen vermögen. Klar, Menschen sind wie sie sind, ich persönlich hätte aber auch nach einem Jahr im selben Umfeld an meiner Erfahrung fest und erkäre Leuten immer wieder mal wie meine halbwissige Steuererklärung aussieht. Auch wenn das nur selten wirklich zu Veränderung beitragen würde, versucht ich das möglichst, die Rollen durch die Formulierung mehrerer Anforderungen auf ihre zentralen Aufgaben hinzuordnen und möchte unter keinem Umstand die bereits angesprochene "absolute Wahrheit" prägen. Tatsächlich müsst du ja auch länger am Formular festhalten können, und späterstens auf der Ebene einzelner Attribute einer Schnittstelle ist die Grenze an Überblick und laufender Aktualität des Analytikers in greifbarer Nähe.
vor etwa einem Monat · Gefällt mir · gefällt mir

Eine häufig gehörige Antwort auf die von Vinzenz gestellte Frage ist: "Der Analytiker beschreibt das Wks und nicht das Wie". Was halte ihr davon?
vor etwa einem Monat · Gefällt mir · gefällt mir

Klingt gut, aber auch beim WAS gehen die Meinungen schon sehr auseinander ... läuft man dann als Analytiker nicht wieder Gefahr, diese angesprochene "absolute Wahrheit" predigen zu wollen?
vor etwa einem Monat · Gefällt mir · gefällt mir

Also ich muss sagen, dass ich im Programm ... mit dem "Der Analytiker beschreibt das was und nicht das wie" schon ein paar mal eingefahren bin. Bspw. in einer meinen ersten Stories habe ich geohrfeigt "Wähn' bähm' Ändern des Passworts ein Fehler äußertert ist wird eine Fehlermeldung angezeigt." In dem Irrglüben dass es bereits ein Leitfaden gibt, wie Fehlermeldungen darzustellen sind. Die Konsequenz war, dass jedes Frontend anders dargestellt hat und auch der Fehlermeldungs-Text unterschiedlich war. Da hier aber sehr auf Konsistenz der verschiedenen Plattformen geachtet wird, ist das beim Test dann aufgefallen und ich musste einen Change Request schreiben um es gerade zu ziehen (weil es ja zu ungern angefordert war). Anderses Beispiel: Ich habe definiert, dass ein bestimmtes Setting gespeichert werden soll. Der API-Server hat das sehr genauso als Key-Value Pair implementiert. Die Konsequenz war, dass jedes Frontend sich unterschiedliche Bezeichnungen ausgedacht hat, wodurch natürlich keine plattformübergreifende Anwendung möglich gewesen wäre. Ich habe daraus gelernt, dass man bestimmte Dinge einfach vorab klar definieren, und manchmal eben das "Wie" vorgeben muss. Dinge wie z.B. eine json-Struktur kann ich jetzt immer vorab mit den Entwicklern abstimmen und schreibe das dann auch in die Anforderung. Und ich sehe das schon als Aufgabe eines Analytikers an.
vor etwa 10 Monaten · Gefällt mir · gefällt mir

Wir leben heute nicht in einer idealen Welt. Und wenn es anderes nicht funktioniert, dann muss über der Analytiker Dinge auch genauer vorgelegt. Da kommt es dann schon auf das Schreibverständnis der Entwickler an. An- und für dich eigentlich reicht eine nicht-formelle Anforderung zu schreiben, wie "Die Fehlermeldungen sollen auf allen Plattformen gleich aussehen". Wenn das so nicht funktioniert, dann bißler eben nichts anders übrig, als es detaillierter zu beschreiben. Wenn ich etwas genauer beschreibe: Algorithmen, Layouts ... kann bezeichne ich das oft als "Vorschlag". Wenn die Entwickler eine bessere Idee haben, dann können sie das ... nach Rücksprache - auch anders machen. Aber ich habe damit zumindest nachgewiesen, dass es funktioniert. Wenn es einen anderen - besser - Weg gibt, der zum Ziel führt, auch gut. Die Unterscheidung zwischen Was und Wie gefällt mir aber aus einem anderen Grund nicht: jede Anforderung kann je nach Betrachtungsebene sowohl "Was" als auch "Wie" sein. Beispiel: Das "Was" könnte sein: Das System soll Tickets verkaufen können. Jedes Detail / welche Daten eingegeben werden müssen, z. B. - wäre in dieser Betrachtungsebene schon das "Wie" - auf der nächsten Ebene aber wieder ein "Was" - und so kommt man nie zu einer Grenze, wieweit Analyse zuständig ist.
vor etwa einem Monat · Gefällt mir · gefällt mir

Abb.: Ausschnitt der Diskussion in unserem internen Social Media Network
(Quelle: ©SEQIS Software Testing GmbH)

An diesem sehr trivialen Beispiel zeigt sich also bereits, dass Faktoren, wie besprochener Detailgrad, Rolle und persönliche Interpretation, keine ausreichende Abgrenzung zulassen, wieweit die Analysearbeit gehen und demnach die Anforderung beschreiben sein soll. Noch viel weiter entfernt ist folglich die Vorstellung, dass ohne Weiteres Selbstverständlichkeit über die Aufgaben und Ergebnisse aus Anforderungsanalyse, Entwicklung und Test herrschen kann.

Hands-On Agilität: Zuerst muss es das Team akzeptieren

Ein SEQIS Test Consultant berichtet über seine Wahrnehmung, dass in jedem Entwicklungsvorgehen die Rahmenbedingungen für effektives Arbeiten zuerst auf sozialer Ebene geschaffen werden müssen.

Genauer gesagt muss ein gewisses Verständnis darüber herrschen, was ein Kollege vom anderen benötigt und was er von ihm realistisch zu erwarten hat. Bezogen auf ein Anforderungsdokument heißt es also z.B. den darin beschriebenen Detailgrad und die damit verbundene Auswirkung auf Personen, die mit diesem Dokument arbeiten müssen, zuerst zu erläutern und zumindest eine gewisse Richtung für einen gemeinsamen Arbeitsmodus zu vereinbaren.

Entwicklung und Test müssen ein Gefühl dafür entwickeln, wann eine Anforderung unzureichend genau beschrieben ist, sodass weiter nachgefragt werden muss, und wann die nicht niedergeschriebene Semantik innerhalb eines Teams als Definition ausreicht. Jedem Mitglied eines Entwicklungsteams sei an diesem Punkt jedenfalls empfohlen, eher oft und wiederholt nachzufragen, als zu wenig. Viele Fragen schärfen im Weiteren das Verständnis und die Häufigkeit der Fragen wird sich reduzieren.

Für einen an der Diskussion beteiligten Business Analysten ist die beinahe systematische Kommunikation ohne-

hin das Mittel zum Zweck. Beispielsweise bietet das agile Projektmanagement-Modell Scrum ein Rahmenwerk, das den Kommunikationsaspekt umfassend und durchdacht aufgreift. Beginnend mit der Rolle des Product Owners (PO) soll es eine Instanz geben, die als Interessensvertretung des Auftraggebers die nicht veränderbaren Rahmenbedingungen vorgibt, also Verantwortung für die Definition eines Softwareprodukts trägt.

In diesem Interesse sollte der PO unbedingt auch laufend während des Lösungsdesigns für Detailfragen zur Verfügung stehen. Ausgehend von dieser wesentlichen Rolle liegt es in der Kompetenz von Analysten, Entwicklern und Testern, eine aus der Vorgabe folgende Lösung zu entwerfen und solange daran zu feilen, bis ein unter konsolidiertem Verständnis entstandenes (Teil-)Produkt vorliegt, an dessen Korrektheit zumindest innerhalb dieses Personenkreises kein nennenswerter Zweifel mehr besteht.

Werkzeuge wie Sprint-Plannings, Schätztechniken wie Planning-Poker im Rahmen der Estimation-Meetings, Retrospektiven aber auch das bereits angesprochene darüber hinausgehende Maß an proaktivem „Nachfragen“ sind die essentiellen Bestandteile, die benötigt werden, damit eine Anforderung den gewünschten Nutzen beschreiben kann.

Nur durch die ständige Auseinandersetzung und Reflexion der eigenen Auffassung zu einem Thema können die bereits erwähnten Risiken durch unabgestimmte Interpretation vermieden werden, was durch eine einzelne Person – unabhängig von deren Kompetenz – nicht zu bewerkstelligen wäre.

Daher unser Praxistipp:

Lassen Sie Diskussionen zu Ihren umzusetzenden Inhalten nicht nur zu, sondern forcieren Sie diese. Es

müssen nicht die in Scrum definierten Meetings sein. Denn im Kern geht es darum, den Beteiligten zur Anwendung solcher Techniken eine angemessene Freiheit einzuräumen, um das individuelle Verständnis und somit die Gesamtqualität maßgeblich zu steigern. Sehen Sie die richtigen Kommunikationskanäle auch in zeitkritischen Projektphasen nicht irrtümlich als Sparpotential an.

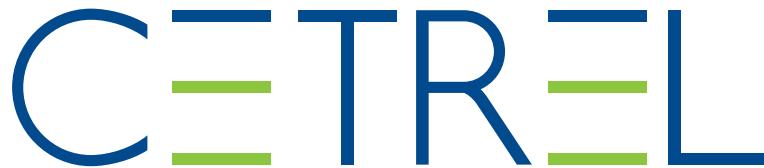
Die Kunst, Vorgaben und Freiheiten zu wählen

Die wohl am heißesten diskutierte Frage in der Expertenrunde der SEQIS Analyse-Mannschaft ist jene nach Detailgrad des Inhaltes einer Anforderung. Ausgehend von der Tatsache, dass es ganz offenbar keine allgemeingültige Formel gibt, was eine Anforderung tatsächlich definieren soll und darüber hinaus ein enormer Einfluss durch Projektumwelt und Persönlichkeit auf eine Gruppe einwirkt, stellt sich die Frage, was der Analytiker nun im konkreten in die Anforderung schreiben soll, damit Entwicklung und Test effizient und zielgerichtet arbeiten können und der Auftraggeber gleichzeitig ein zufriedenstellendes Produkt erhält.

Zwei unserer Business Analysen berichten aus ihrer täglichen Routine, Beschreibungen – in gewisser Hinsicht Anleitungen – für Programmierer und Tester zu erstellen. Die Erkenntnisse zeigen, dass in der Praxis oft mehr nötig ist, als ein fachliches Big Picture aufzuzeichnen. Die hierzu erwähnten Erfahrungen der beiden stammen u.a. aus einem System mit verteilter Architektur zwischen Backend-Anwendungen, einer verteilenden bzw. aufbereitenden Zwischenschicht und der Weiterleitung von dort zu mehreren unterschiedlichen Client-Anwendungen.

Die Fortsetzung finden Sie auf S. 24!

Interview with 4 key players at CETREL S.A.



a SIX Company

© CETREL S.A.

CETREL S.A., a 100% affiliate of SIX Payment Services, and SEQIS have a long-term history of successful cooperation. After finishing the big migration project that encompassed the migration of the Luxembourg banks' card portfolio on the new processing platform of SIX, Alexander Vukovic (SEQIS) takes the chance to interview the key players at SIX and CETREL:

Daniel von Aesch (COO), Heinz Saria (Project Lead), Alain Barthelemy (Lead System Integration Test Team) and Renaud Dechambre (Lead Business Acceptance Test Team).

VUKOVIC: Daniel, as the COO of CETREL and the main project lead for this huge migration project, what do you think, were the most thrilling challenges in this critical and successful project from a SIX point of view?

VON AESCH: Over its many years in service, our legacy environment had grown with our customers and their needs. In the new system, there had to be a corresponding working solution for all their requirements. The biggest challenge was to gather absolute clarity of the features of our 25 year old home grown solution and their behavior in the new solution. Enriching the basic functionalities of the new solution with the features and functions that capture our industry expertise and experience and that have served our customers for decades drives by the end of the day architectural

and performance challenges. Another challenge was the high number of necessary customizations, which by definition interfere with the maintainability on the long run of a standard solution.

VUKOVIC: Alain, in your SIT-Team you introduced the idea of an automatic sanity check, could you describe the main targets of the sanity check? What do you think about the SEQIS-implementation of the Sanity Check based on Jenkins, Subversion and Powershell?

BARTHELEMY: As we are working with an external provider in an intensive roadmap, we have several deliveries, which may introduce a lot of regressions in the production system. Therefore, we needed a way to quickly be able to say if a delivery of a new release was passing the minimum set of quality criteria. This was the target of the Sanity Check. It is more than useful. We are using it intensively. The choice of Jenkins and Subversion gave us the opportunity to enhance the test cases easily in a collaborative way.

We rewrote it to a light Sanity Check to have additional faster checks. We analyze also all dependencies of the delivery, i.e. we are checking that there is no orphan piece of code. Next step will be to include NeoLoad tests (Webservices testing tool) in Jenkins.

VUKOVIC: CETREL/SIX wanted to improve the Test Automation Strategy. What is your opinion about

Rubric "Spotlight"

We always focus on customer needs. No matter which industry, technique and challenge – we support with consulting, coaching and training.

In this rubric we introduce some of our customers and want to provide an insight into our different projects and services.



Mag. Alexander Vukovic is a pure agile testing wizard, founder of SEQIS and member of the SEQIS management board.

In his career he worked as java and javascript developer, functional tester, load tester, test automation specialist, test manager, project manager, trainer and coach. Since 2004 he converted to be a pure agile wizard, starting to work in and with agile teams. Today he works as specialised agile testing and quality coach.

He combines fundamental theoretical knowledge (ISTQB Full, Certified PO, SM, CAT-Trainer, TDD) with 20 years of practical experience.

the achievements SEQIS in this main project target, and what do you think, is the future of test automation at CETREL?

BARTHELEMY: I am proud of the solution which is currently in place. Of course it has to be enhanced. Test Automation is something that is living and it is definitely the future for us. It also requires analysis and methodology: The maintenance of an automated test may cost more than performing manually the test; a right balance has to be found. Next step is to have a dedicated screen showing the actual status of the Jenkins Sanity Check as well as including our own pipes.

VON AESCH: The target has been fully achieved. It is absolutely mandatory to have a sufficient test automation solution in place, if you have a standard product enriched by own developed customizations, which have to be tested on compatibility with each

and every delivery of the supplier. It is necessary to know as fast as possible, if the changed customizations and the changed standard product are still working together.

DECHAMBRE: By appointing SEQIS we wanted to address the fact that we had barely practiced test automation in the past. Thanks to your cooperation, we now have the sanity check and 50% of the functional testing is fully automated. With tools answering our needs, I am thinking of the implementation of Jenkins. It also helped us to get the appropriate management attention for benefits of and the dangers in the absence of test automation.

E.g. GUI-Testing, we realized that it was economically not a good idea. The future: There is always room to improve, so we will continue to work on test automation, and SEQIS would be our natural choice if we need external support in the test domain.

SARIA: With the help of SEQIS we successfully introduced automated sanity checks, which performed the equivalent of 1-2 days of manual testing within 1 hour. SEQIS also implemented a load test environment, which is complex in functionality but easy to use. Additionally SEQIS selected and introduced a number of additional test support tools (file comparison, DB comparison, ...) and provided training sessions for our testers.

VUKOVIC: Renaud, you and your BAT-Team are doing a very advanced parallel run, comparing real production traffic on two parallel test environments with different application versions. For that you have a lot of comparison effort, e.g. comparing result files and xmls.

So SEQIS did a lot of tool evaluations to find automatic ways to compare e.g. structured data with keys stored in flat files. What do you think



Titel: „Das große Rennen“, Künstler: René Niedereder, Technik: Acryl und Klebetechnik auf Leinwand

about the approach of the SEQIS tool evaluations and the evaluation results?

DECHAMBRE: So, first thing when you came in was explaining the selection process and criteria. It was a high sophisticated comparison approach, with clear and factual measures. This was for me already an indication that the evaluation will be really objective with transparency. Second, when you started evaluating the tools, you put yourself into the situation we are, not only reading white papers. This is the only realistic way to understand our real needs.

Finally you touched the goals by selecting tools, which were usable and providing us with the right things. This process revealed the necessity of some further in-house development to accommodate the size of the files, which was an important finding as there was no tool on the market to fulfill our needs. Very good process, very good understanding of customer needs and very objective criteria. Good results at the end. The outside view was very helpful for us to understand our real needs.

VUKOVIC: Heinz, the performance engineering of web services was one of SEQIS main tasks at CETREL. Could you outline the original performance and the final performance we achieved after the performance engineering? How was SEQIS contributing to these results?

SARIA: Integration with client bank systems via web services is a key success factor in today's payment market. At CETREL we process more web services calls than card authorizations. During our migration to a new card processing system we encountered during initial tests average response times of several seconds. This was unacceptable. Finally we achieved average response times around 200 ms.

SEQIS helped us to set up an efficient state of the art web service load test environment. Additionally SEQIS performed most of the load tests. As a result we got comprehensive and conclusive performance analysis reports.

These reports helped our vendor to find the implementation bottlenecks and helped CETREL to find the proper dimensioning of the HW platform and system setup.

VUKOVIC: How would you summarize and value the work of SEQIS in your project?

SARIA: Based on the analysis of our test environment and test methods the right methods and tools were selected and the right training was provided to make us fit for future projects. As SEQIS is no tool provider on itself the recommendations were really based on our needs. It helped a lot that SEQIS is also experienced in our business domain.

VON AESCH: Added value of SEQIS was very high. Unlike in other SIX location, test engineering knowledge was not sufficient here in Luxembourg, which we wanted to make up for by using an outside provider. SEQIS was

selected after a Proof of Concept with several suppliers and proofed the best supplier. SEQIS was known as well to be a reliable testing company with good expertise from a former project with SIX.

To setup the full testing, automation and performance, e.g. with NeoLoad, was a lot of expertise, which was brought to SIX/CETREL in Luxembourg, which was not yet established with this level here.

BARTHELEMY: Imaginative, intensive, proactive and efficient.

DECHAMBRE: SEQIS is dedicated to testing, with no influence from any solution provider, with people who are really focusing into the test domain. Due to this clear focus, SEQIS is ideally positioned to support companies implementing test attitude. They have a lot of expertise in difficult domains, where it is hard to find people with this level of expertise. For us the result is, that we have a better view of our requirements and we have the right level of test automation in place. Very friendly team and very good relationship.

VUKOVIC: Thank you very much! ■



© CETREL S.A.

f.l.: Alain Barthelemy (Head of System Integration Testing), Alexander Vukovic, Renaud Dechambre (Head of Business Acceptance Testing), Daniel von Aesch (COO CETREL S.A.), Heinz Sarria (Project Lead Sunshine/Moon)

Anforderungen gibt es nicht

von Josef Falk



Mag. Josef Falk
ist Senior Test Consultant und Business Analyst bei SEQIS.

Eine fundierte betriebswirtschaftliche Ausbildung verbunden mit langjähriger Entwicklungserfahrung in zahlreichen erfolgreichen Projekten machen ihn zum Spezialisten auf seinem Gebiet.

Diese Kombination ermöglicht es ihm, betriebswirtschaftliche Probleme auf den Punkt zu bringen, die daraus entstehenden Anforderungen zu analysieren und so zu beschreiben, dass eine zu implementierende Lösung exakt die Anforderungen erfüllt.

Wie jetzt? Sind nicht gerade Anforderungen das Wichtigste im Leben eines IT-Analytikers? Wird nicht die gesamte Profession sehr häufig „Requirements Engineering“ oder „Anforderungsmanagement“ genannt? Was bedeutet dann diese Überschrift? Das kann doch nicht des Autors Ernst sein!

Ist es auch nicht – zumindest nicht so ganz. Der Titel dieses Aufsatzes ist inspiriert durch den Artikel des Bloggers Tony Heap: „There's No Such Thing as a Requirement“ (www.its-all-design.com/theres-no-such-thing-as-a-requirement/). Tony Heap schreibt in diesem Artikel, dass es letztlich immer um Design gehe. Auch wenn Stakeholder ihre Vorstellungen und Wünsche an das System bekannt geben,

dann sind das Design-Ideen, die gut sein können oder auch nicht. Es gibt immer eine Option – man könnte immer auch eine andere Lösung finden. Deshalb: Es gibt keine Anforderungen.

Um Tony Heap's Aussagen genauer beurteilen zu können, wollen wir uns zunächst den Begriff der „Anforderung“ näher ansehen. Laut Duden ist eine „Anforderung“ ein „Anspruch an jemandes Leistung“. Wenden wir diese Bedeutung auf die IT-Analyse an. Wer ist es, der einen Anspruch stellt – und an wen wird dieser Anspruch gestellt?

Da gibt es zunächst einen Auftraggeber, der eine bestimmte Aufgabe stellt. In der Regel ist es die Entwicklung einer Software, durch die Wert für die jeweilige Organisation geschaffen werden soll.

Daneben gibt es zumeist eine Reihe weiterer Ansprüche (= Anforderungen), die an die Software gestellt werden. Diese Ansprüche werden einerseits vom Auftraggeber selbst formuliert, andererseits aber auch von Personengruppen, die – explizit oder implizit – von diesem dazu ermächtigt wurden.

Üblicherweise fassen wir diese unter dem Begriff „Stakeholder“ zusammen. Diesen Begriff wollen wir auch in weiterer Folge verwenden. Es ist die Aufgabe des Analytikers, die Anforderungen der Stakeholder zu sammeln. Aber auch bei größter Anstrengung werden die so ermittelten Anforderungen die erforderliche Lösung nicht vollständig beschreiben.

Vergleichen wir diese Situation mit einem geplanten Autokauf. Auch hier werden wir als Käufer verschiedene Ansprüche haben, die individuell unterschiedlich sind: Einmal wird es

das Platzangebot sein, das besonders wichtig ist. Das andere Mal die PS-Zahl, die Umweltfreundlichkeit, die Geländegängigkeit, der Preis oder andere Faktoren. Aber niemals werden die „Anforderungen“, die wir als Käufer haben, das Auto vollständig beschreiben. Es wird viele Elemente geben, die wir als gegeben voraussetzen und die wir niemals als Anforderungen formulieren würden.

Daneben gibt es immer auch Leistungselemente, von denen wir uns gerne überraschen lassen – die wir nach dem Kauf am Produkt schätzen, die wir vor dem Kauf aber ebenfalls nicht als Anforderungen festgehalten hätten.

Ebenso wie einem Autokäufer geht es einem Stakeholder in einem Software-Projekt. Er weiß, welche Aufgabe das Produkt erfüllen soll und er hat die eine oder andere Idee (bzw. den Anspruch), auf welche Weise das passieren soll. Niemals aber werden diese Ideen des Stakeholders – die „Anforderungen“ – das Produkt vollständig beschreiben können.

Innerhalb eines Software-Projekts ist es die Aufgabe der IT-Analyse, die geplante Lösung so zu beschreiben, dass sie im Anschluss (oder auch teilweise parallel) von der Entwicklung implementiert werden kann. Diese Lösung muss allerdings den genannten Ansprüchen (den „Anforderungen“) des Auftraggebers bzw. der übrigen Stakeholder genügen. Deshalb ist es ein wesentlicher Teil der Aufgabe eines IT-Analytikers, diese Anforderungen zu sammeln, sie zu ordnen und in richtigem Umfang zu dokumentieren.

Wir sehen also, dass die Analyse die „Anforderungen“ des Auftraggebers entgegennimmt und daraus ein

System-Design erstellt, das in weiterer Folge von der Entwicklung in eine Software-Lösung umgesetzt wird.

Wenn aber, wie wir gesehen haben, die Anforderungen der Stakeholder die Lösung nicht vollständig beschreiben, was sind dann die restlichen Bestandteile, die das Design der Lösung ausmachen?

Folgende Punkte sind in diesem Zusammenhang zu nennen:

1. Wissen über das Fachgebiet:

Jedes Fachgebiet funktioniert nach einem gewissen Regelwerk. Das kann einerseits in Gesetzen niedergeschrieben sein, es kann sich aus einem allgemein anerkannten „State of the Art“ ergeben oder es kann ganz einfach in der Natur des Fachgebietes begründet sein. Es ist die Aufgabe des IT-Analytikers, sich dieses Wissen anzueignen und, darauf aufbauend, das Design der Lösung zu bauen.

2. Kreativität und Ideen: Je nach Art der Aufgabenstellung fließen immer auch – in geringerem oder größerem Ausmaß – eigene Ideen der Analytiker, wie man die Aufgabenstellung besser bewältigen könnte, in das Lösungsdesign ein.

3. Kombination von „Anforderungen“, „Wissen“ und „Ideen“ zu einem System. Dieses Zusammenfügen der drei Grundelemente ist die Haupttätigkeit des IT-Analytikers.

Drei Dinge sind es also, die in das Lösungsdesign einfließen: „**Anforderungen**“, „**Wissen**“, „**Ideen**“. Nach der Kombination dieser drei Elemente ist die Lösung vollständig beschrieben. Sehr häufig werden diese drei Elemente unter dem Begriff der „**Anforderungen**“ zusammengefasst.

Es handelt sich jedoch um drei Dinge, die unterschiedliche Quellen haben, die auf unterschiedliche Art und

Weise ermittelt werden und die auch unterschiedlich behandelt werden müssen. Deshalb ist es zweckmäßig, diese drei Dinge auch unterschiedlich zu benennen.

Ein „Alles ist Anforderung“ ist ebenso wenig hilfreich wie ein „Anforderungen gibt es nicht“.

Soweit zunächst zu den Anforderungen aus der Sicht der IT-Analyse. Wie sieht es aber mit der Entwicklung aus. Gibt es auch hier „Anforderungen“?

Wiederholen wir an dieser Stelle noch einmal, was „Anforderung“ bedeutet: Eine Anforderung ist ein „Anspruch an jemandes Leistung“. Wo ist nun der Anspruch an die Leistung der Entwicklung festgeschrieben? Es ist der Output der IT-Analyse, also das Design der Lösung, das die geforderte Leistung der Entwicklung beschreibt.

Wenn man so möchte, könnte man also auch das Design der Lösung als „Anforderungen“ bezeichnen. Tatsächlich schlägt das auch der Business Analysis Body of Knowledge“

(BABOK-Guide) so vor: Der Begriff „Design“ wird nur für „technisches Design“ verwendet. Das fachliche Design wird als „Anforderungen“ bezeichnet.

Wir plädieren hier jedoch für eine klare Trennung der Begriffe: „Anforderungen“ sind die Ansprüche des Auftraggebers bzw. der Stakeholder an die Software. Diese Anforderungen werden von der IT-Analyse gemeinsam mit Wissen über das Fachgebiet und eigenen Ideen zum „fachlichen Design“ kombiniert, das als fachliche Vorgabe für die Entwicklung dient.

Fazit

Also noch einmal: Gibt es so etwas wie „Anforderungen“? Ja, es gibt sie. Stakeholder haben bestimmte Ansprüche an das System, das wir als IT-Analytiker entwerfen. Diese Ansprüche nennen wir „Anforderungen“. Diese Anforderungen beschreiben aber niemals das gesamte System.

Es ist die Aufgabe des IT-Analytikers, diese „Anforderungen“ als Rohmaterial zu verwenden und daraus ein System zu bauen. Neben den vom

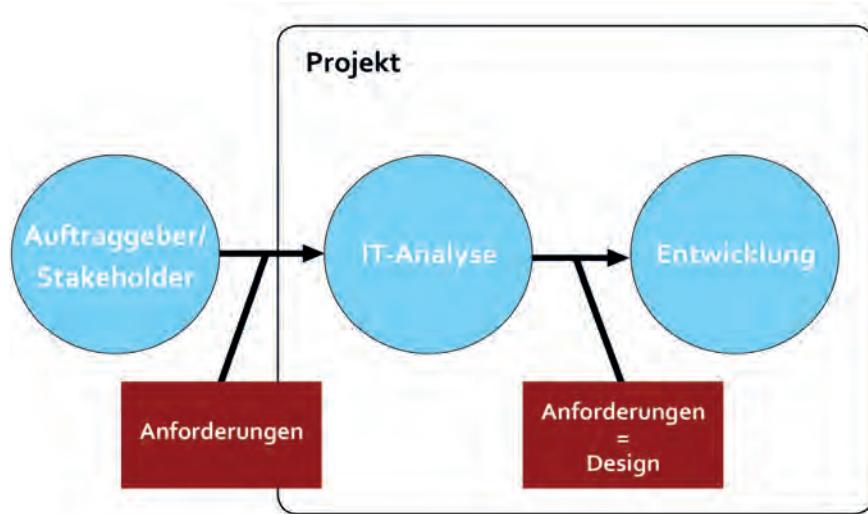


Abb.: Anforderungen in Analyse und Entwicklung
(Quelle: ©SEQIS Software Testing GmbH)

Auftraggeber und seinen Vertretern genannten Anforderungen sind dafür zwei Gruppen weiterer Ingredienzen erforderlich:

- Wissen über das zu bearbeitende Fachgebiet sowie
- Kreativität und viele gute Ideen

Die Kombination dieser drei Bestandteile ergibt schließlich ein Systemdesign, das Wert für die Organisation schafft.

Und warum ist das wichtig?

Aber ist das alles nicht bloß eine akademische Frage? Geht es hier lediglich um Wortklauberei – ohne jede praktische Relevanz?

Diese Frage ist deshalb von Bedeutung, weil sie die Arbeitsweise und das Selbstverständnis des Analytikers beschreibt. In der IT-Analyse geht es nicht (nur) darum, Anforderungen zu sammeln, zu ordnen und zu dokumentieren. Gut dokumentierte Anforderungen sind NICHT das Endprodukt der IT-Analyse.

Nein, das Ziel und das Produkt eines Analytikers ist das Design eines Systems, der Entwurf einer Lösung, die von der Entwicklung umgesetzt werden kann. Anforderungen sind ein wertvoller Rohstoff dafür.

Nicht mehr und nicht weniger! ■

Sie möchten mehr zu den Themen Business Analyse, Requirements Engineering, Anforderungsmanagement, Systemanalyse 3.0 erfahren?

Besuchen Sie unseren Blog und beteiligen Sie sich an spannenden Diskussionen unter Experten: blog.SEQIS.com

Folgen Sie uns auf Twitter: twitter.com/Systemanalyse30



Titel: „Frühling“, Künstlerin: Katarina Tadic, Technik: Aquarell auf Papier

SEQIS „10 things“-Expertentreff #3/2015: Pair Analyse

von Julia Kremsl

Pairing + Business Analyse = Pair Analyse

Diese „10 things“-Veranstaltung drehte sich ganz um das Thema „Pair Analyse: Qualität von Anfang an“. Wieder einmal konnte eine unserer Expertinnen ihren jahrelangen Erfahrungsschatz an die begeisterten Teilnehmer weitergeben: Analystin Veronika Rumpler verriet ihre innovativen 10 Tipps und Tricks zur Pair Analyse.

Doch was steckt hinter dem Begriff Pair Analyse? Pair Programming ist eine Arbeitstechnik, bei welcher zwei Programmierer gleichzeitig an einem Rechner arbeiten. Einer programmiert, während der andere über die Problemstellung nachdenkt und Bugs sofort anspricht. Dies reduziert die Fehlerhäufigkeit und erhöht die Softwarequalität.

Doch wie passen Pairing und Business Analyse zusammen? Die Antwort darauf liefern folgende 10 Tipps und Tricks zum effizienten Einsatz der Pair Analyse:

Tipp 1: Lösungen finden – „Denken in vielen Richtungen“

Ein wesentliches Skill in der Business Analyse ist das schnelle Finden von mehreren Lösungen, um anschließend das zielführendste Resultat für die gegebenen Anforderungen zu ermitteln. Die Herausforderung dabei besteht darin, in möglichst kurzer Zeit zu möglichst vielen Lösungsvorschlägen zu kommen.

Damit dies gelingt, muss unbedingt in viele unterschiedliche Richtungen gedacht werden. Alleine ist das jedoch kaum möglich. Holen Sie sich Kollegen mit unterschiedlichen Backgrounds mit an Board. Dies minimiert das

Risiko Schwachstellen zu übersehen und erhöht die Qualität der Lösung. Wenden Sie dazu unterschiedliche Kreativitätstechniken an.

Tipp 2: Meetings effektiv gestalten – Workshops immer zu zweit leiten

Der nächste Schritt beim Finden der zielführendsten Lösung sind diverse Meetings und Workshops mit Stakeholdern, Mitarbeitern des Fachbereichs, etc. Um diese Meetings möglichst effektiv zu gestalten empfiehlt sich das Hinzuziehen eines zweiten Analysten. Dieser ist für die Einhaltung der Agenda und des Zeitmanagements während des Meetings zuständig. Vor allem aber dokumentiert er Entscheidungen und Erkenntnisse.

Dadurch können Sie sich ausschließlich auf die Diskussionen und die Lösungsfindung konzentrieren und alle organisatorischen Belange abgeben.

Tipp 3: Lösung in Frage stellen – bewusst anderer Meinung sein

Häufig hilft das „Verteidigen“ der eigenen Lösung, um die Vor- und Nachteile zu verdeutlichen und die Situation aus einem anderen Blickwinkel zu betrachten. Hier können unterschiedliche Methoden eingesetzt werden:

Probieren Sie doch einmal die sogenannte „Walt Disney Methode“ aus und schlüpfen Sie in die Rollen des Träumers, Realisten und Kritikers oder setzen Sie sich nach der „6 Thinking Hats“-Methode unterschiedliche Hüte auf. Beziehen Sie auch Ihre Kollegen mit ein.

Tipp 4: Lösung untermauern – Vor- und Nachteile ermitteln im Team

Die Auflistung der Vor- und Nachteile aller Lösungen im Team hilft, alle positiven wie negativen Aspekte



Julia Kremsl,
MA ist Marketing Managerin bei SEQIS. Sie ist erste Ansprechpartnerin für alle Presse- und Marketing-informationen.

Von klassischer Pressearbeit, der Gestaltung von Drucksorten, der Wartung der Homepage bis hin zur Organisation von internen und externen Veranstaltungen übernimmt sie alle Marketing- und Kommunikationsagenden. Ihrer Kreativität kann sie dabei freien Lauf lassen.

Besonders am Herzen liegen ihr die interne Kommunikation und das Schaffen neuer Ideen und Konzepte.

aufzulisten. Eigene Lösungsfavoriten können dadurch noch einmal überdacht oder sogar verworfen werden. „Bauchentscheidungen“ sollten dabei unbedingt rational untermauert werden. Dokumentieren Sie auch die Nicht-Lösungen und halten Sie die Gründe, die gegen diese Vorschläge sprechen, fest.

Tipp 5: Wissensübergabe – auch im Team

Gerade in der Business Analyse ist die fachliche Wissensübergabe unerlässlich. Der gemeinsame Austausch zum aktuellen Problem mit Kollegen trägt nicht nur zur Vermittlung des Fachwissens sondern auch zur Verbesserung der Analyse-Skills und der Toolkenntnisse bei.

Die Fortsetzung finden Sie auf S. 17!

Terminvorschau 2016:

iSQI® Certified Agile Test Driven Development*:

07.03. - 09.03.2016,
Mödling

ISTQB® Certified Tester, Foundation Level Extension, Agile Tester*:

10.03. - 11.03.2016,
Mödling

iSQI® Certified Agile Business Analysis*:

21.03. - 22.03.2016,
Mödling

ISTQB® Certified Tester, Advanced Level, Test Analyst:

04.04. - 07.04.2016,
Mödling

CMAP® Mobile App Testing, Foundation Level*:

18.04. - 19.04.2016,
Mödling

Die Anmeldung ist ab sofort möglich!

Kontakt:
sales@SEQIS.com

Weitere Infos:
www.SEQIS.com/ausbildung/alle-kurse-im-ueberblick

| Dezember | | | Januar | | |
|----------|----|------------------|--------|----|---------------------|
| 1 | Di | | 1 | Fr | Neujahr |
| 2 | Mi | | 2 | Sa | |
| 3 | Do | | 3 | So | |
| 4 | Fr | | 4 | Mo | |
| 5 | Sa | | 5 | Di | |
| 6 | So | | 6 | Mi | Heilige Drei Könige |
| 7 | Mo | | 7 | Do | |
| 8 | Di | Mariä Empfängnis | 8 | Fr | |
| 9 | Mi | | 9 | Sa | |
| 10 | Do | | 10 | So | |
| 11 | Fr | | 11 | Mo | |
| 12 | Sa | | 12 | Di | |
| 13 | So | | 13 | Mi | |
| 14 | Mo | | 14 | Do | |
| 15 | Di | | 15 | Fr | |
| 16 | Mi | | 16 | Sa | |
| 17 | Do | | 17 | So | |
| 18 | Fr | | 18 | Mo | |
| 19 | Sa | | 19 | Di | |
| 20 | So | | 20 | Mi | |
| 21 | Mo | | 21 | Do | |
| 22 | Di | | 22 | Fr | |
| 23 | Mi | | 23 | Sa | |
| 24 | Do | Heiligabend | 24 | So | |
| 25 | Fr | Christtag | 25 | Mo | |
| 26 | Sa | Stefanitag | 26 | Di | |
| 27 | So | | 27 | Mi | |
| 28 | Mo | | 28 | Do | |
| 29 | Di | | 29 | Fr | |
| 30 | Mi | | 30 | Sa | |
| 31 | Do | Silvester | 31 | So | |

* Termingarantie! Diese Kurse finden zum angegebenen Termin fix statt.

| Februar | |
|----------------|----|
| 1 | Mo |
| 2 | Di |
| 3 | Mi |
| 4 | Do |
| 5 | Fr |
| 6 | Sa |
| 7 | So |
| 8 | Mo |
| 9 | Di |
| 10 | Mi |
| 11 | Do |
| 12 | Fr |
| 13 | Sa |
| 14 | So |
| 15 | Mo |
| 16 | Di |
| 17 | Mi |
| 18 | Do |
| 19 | Fr |
| 20 | Sa |
| 21 | So |
| 22 | Mo |
| 23 | Di |
| 24 | Mi |
| 25 | Do |
| 26 | Fr |
| 27 | Sa |
| 28 | So |
| 29 | Mo |

Über die „10 things I wished they'd told me!“-Expertentreffs

An Informationen mangelt es meist nicht – im Gegenteil, derer gibt es oft mehr als genug. Wichtiger denn je ist es, an die entscheidenden Informationen zu gelangen. Als Dienstleistungsunternehmen sind wir uns unserer Rolle als Informant bewusst und sprechen die an Softwaretest und Requirements Engineering Interessierten mit der Reihe „10 things I wished they'd told me!“ konkret an.



Für all jene die Software entwickeln, nutzen, beschaffen, in einem Betrieb für die Software-Qualitäts-sicherung zuständig oder als Requirements Engineer/ Business Analyst tätig sind, haben wir eine passende Plattform geschaffen!

Bei unseren Expertentreffs erhalten Sie die Möglichkeit branchen-bezogene Erfahrungen auszutauschen und wertvolle Tipps von den Profis abzustauben. Die Vortragenden bringen aktuelle Test- und Business Analyse-Themen auf jeweils 10 knackige Punkte und teilen mit Ihnen ihre Erfahrungen aus zahlreichen großen und komplexen IT-Projekten.

Weitere Informationen über unsere „10 things“-Veranstaltungsreihe finden Sie unter www.SEQIS.com.

Ihre Meinung ist uns wichtig!

Auch im kommenden Jahr haben Sie wieder die Möglichkeit, sich bei unseren kostenlosen Expertentreffs persönlich weiterzubilden und Ihren Arbeitsalltag mit praxisnahen Tipps zu aktuellen Trendthemen zu bereichern.

Sagen Sie uns, was Sie interessiert! Welche Aufgabenstellungen sind für Sie gerade aktuell? Zu welchen Themen möchten Sie hilfreiche Tipps für Ihren Arbeitsalltag erhalten?

Wir freuen uns über Ihre Vorschläge, Wünsche und Anregungen: www.SEQIS.com/unternehmen/events oder gerne auch per Mail an marketing@SEQIS.com

Gewinnspiel: Machen Sie mit und gewinnen Sie einen Gutschein für 2 Personen von „Time Busters“!

Auch in dieser Ausgabe haben wir für Sie wieder ein Gewinnspiel vorbereitet. Dieses Mal muss folgende Rätselfrage richtig beantwortet werden:

Drei Damen treffen sich zu einer Sitzung: Frau Rot, Frau Weiß und Frau Grün. Eine der Damen stellt fest: „Das ist aber merkwürdig, eine von uns trägt eine rote, eine andere eine weiße und die dritte eine grüne Bluse“. „Das ist wirklich erstaunlich“, meint die Dame mit der roten Bluse, „denn keine trägt die Bluse, welche ihrem Namen entspricht“. „Das stimmt“, ergänzt Frau Weiß.

Welche Dame trägt welche Bluse?

A: Frau Weiß trägt rot, Frau Rot trägt grün und Frau Grün trägt weiß.

B: Frau Weiß trägt grün, Frau Rot trägt weiß und Frau Grün trägt rot.

C: Frau Weiß trägt weiß, Frau Rot trägt rot und Frau Grün trägt grün.

Und so nehmen Sie teil:

Senden Sie uns die richtige Antwort und Ihre Kontaktdata per Mail an marketing@SEQIS.com unter dem Kennwort „QN Gewinnspiel“.

Einsendeschluss ist der 31.01.2016.

Unter allen Einsendungen verlosen wir 1 Gutschein für 2 Personen von „Time Busters“ für ein spannendes Room Escape Abenteuer.

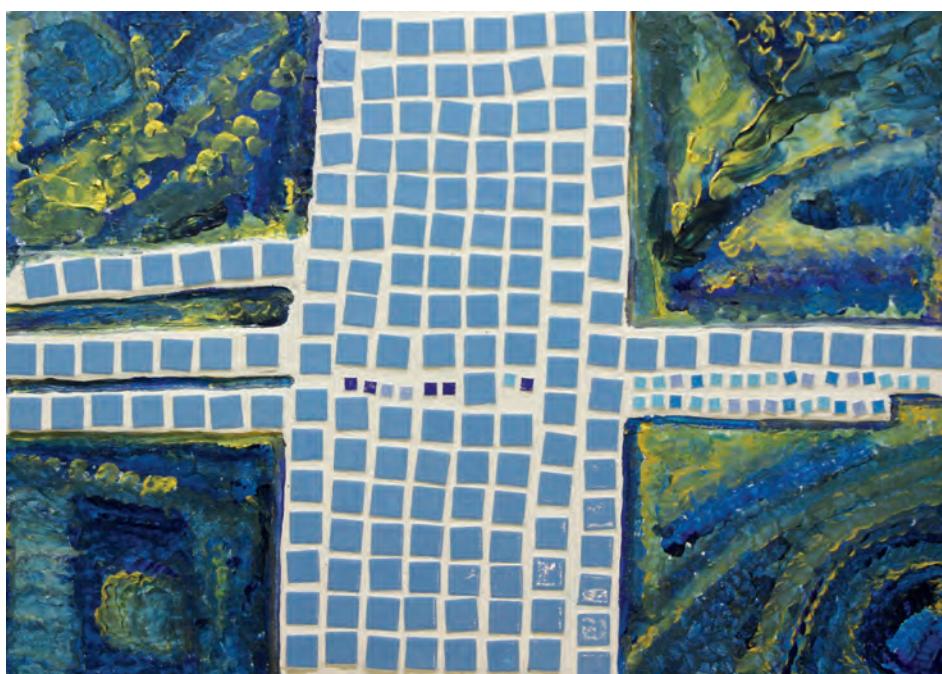
Beobachtungs-gabe, Kombinieren, Querdenken und Teamwork sind gefragt, um in der vorgegebenen Zeit die Mission zu erfüllen! Mehr Infos unter: www.time-busters.at

Die Auflösung des Rätsels finden Sie in der nächsten Ausgabe.

Wir freuen uns auf Ihre Einsendungen und wünschen viel Glück!

Teilnahmebedingungen:

Mit der Teilnahme am Gewinnspiel akzeptieren Sie die folgenden Teilnahmebedingungen. Teilnahmeberechtigt sind Personen ab 18 Jahren. Die Gewinnermittlung erfolgt nach dem Zufallsprinzip und findet nicht öffentlich statt. Der Gewinner wird per Email verständigt. Der Gewinn kann nicht in bar abgelöst werden und ist ausschließlich in der dafür vorgesehenen Form zu konsumieren. Die Gewinnverständigung gilt mit Absendung als erfolgt. Das Zugangsrisiko trägt der Gewinner. Der Teilnehmer stimmt zu, dass die Gewinneinlösung dokumentiert wird und Bildmaterial für Werbung der SEQIS Software Testing GmbH in jeder Form verwendet werden darf. Der Gewinn beschränkt sich ausschließlich auf den Erhalt eines Gutscheins für 2 Personen für „Time Busters“. Pro Person und Gewinn wird 1 Gutschein ausgegeben. Der Gewinn ist übertragbar. Der Rechtsweg ist ausgeschlossen. Die Mitarbeiter der SEQIS Software Testing GmbH sind von der Teilnahme am Gewinnspiel ausgeschlossen. Das Recht, Teilnehmer, die gegen diese Teilnahmebedingungen verstößen, mit sofortiger Wirkung von der Teilnahme auszuschließen, wird vorbehalten. Eine allenfalls bereits erworbene Berechtigung zum Gewinnbezug erlischt in diesem Fall. SEQIS Software Testing GmbH behält sich das Recht vor, die Teilnahmebedingungen für Gewinnspiele jederzeit ohne Ankündigung und Angaben von Gründen nach eigenem Ermessen zu ändern.



Titel: „Raumschiff“, Künstler: Kurt Jagodich, Technik: Acryl und Mosaik auf Karton

Fortsetzung von S. 13.

Tipp 6: Reviews, Reviews, Reviews – Pairing während der Umsetzungsphase

Auch während der Umsetzungsphase ist Pairing in der Business Analyse sinnvoll und sollte deshalb nicht unterschätzt werden. Setzen Sie auch hier auf unterschiedliche Ansichtsweisen und holen Sie auch das Review von Entwicklern und Testern ein. Fehler, die möglicherweise bislang übersehen wurden, können dadurch noch rechtzeitig entdeckt und behoben werden.

Tipp 7: Kein Pairing Partner – nutzen Sie die Gummi-Ente

Pairing kann somit in allen Prozessschritten der Business Analyse eingesetzt werden und bringt zahlreiche Vorteile mit sich. Doch was tun, wenn einmal kein Pairing Partner zur Stelle ist?

Setzen Sie doch einfach auf eine Gummi-Ente. Stellen Sie sich diese auf den Schreibtisch und erzählen Sie ihr das Problem. Mit dieser Methode können Probleme strukturiert beschrieben und neue Wege der Lösungsfindung erschlossen werden. Denn oftmals reicht es schon, wenn es jemanden gibt, der einfach nur zuhört.



© SEQIS Software Testing GmbH

Tipp 8: Pairing nur wenn's passt – nicht immer ist gemeinsam besser

Ist Pairing wirklich immer sinnvoll? Nein, denn es gibt Situationen, in denen Pairing nicht zielführend ist. Ist eine Lösung bereits vordefiniert oder benötigt man selbst erst einmal Zeit um nachzudenken und das Problem zu strukturieren, macht Pairing zu

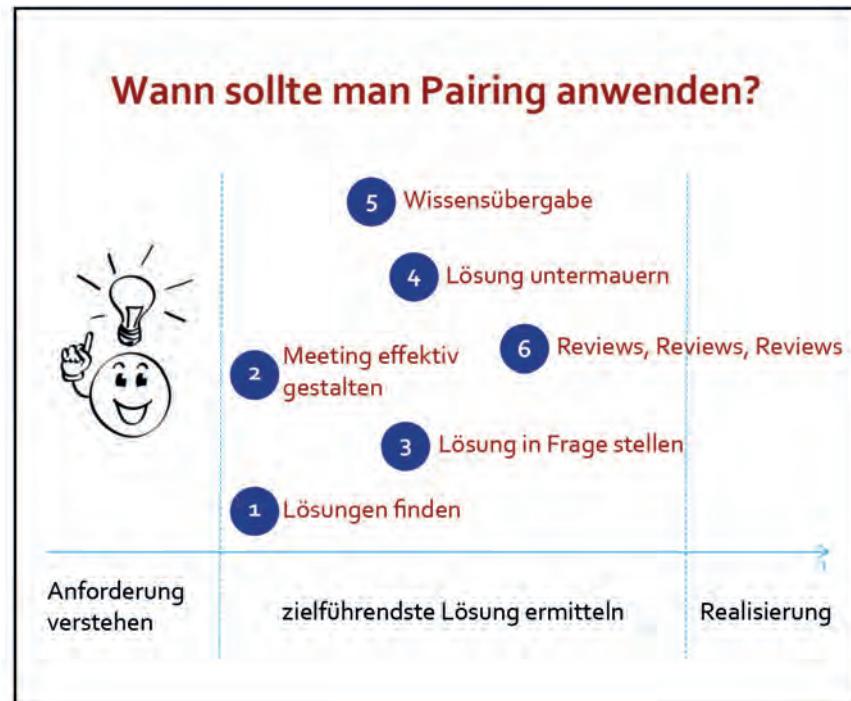


Abb.: Gründe für die Anwendung von Pairing
(Quelle: ©SEQIS Software Testing GmbH)

diesem Zeitpunkt keinen Sinn. Ein weiterer Grund, warum Pairing nicht universal in jeder Situation eingesetzt werden kann: Es gibt Kollegen, mit denen man aufgrund zu unterschiedlicher Arbeitsweisen nicht zusammenarbeiten kann und keinen gemeinsamen Lösungsweg findet. In diesem Fall sollte ein neuer Pairing Partner gesucht bzw. andernfalls auf Pairing verzichtet werden.

Tipp 9: Mit wem arbeitet man im Team – suchen Sie sich den „richtigen“ Partner!

Die Wahl der „richtigen“ Partner ist entscheidend, um mit Pairing erfolgreich zu sein. Vorteilhafte Pairing Partner zeichnen sich durch unterschiedliche Skills und Backgrounds aus und sind an der Lösung interessiert und (in)direkt betroffen. Ferner müssen sie ausreichend Zeit für die Arbeit als Pairing Partner mitbringen.

Tipp 10: Arbeiten im Team macht Spaß

Gemeinsames Ideensammeln steigert nicht nur die Motivation und das Commitment, es führt darüber hinaus auch zu einer höheren Akzeptanz des Fachbereichs, wenn zwei oder mehr Personen hinter einer Lösung stehen.

Zu guter Letzt sei an eine wichtige und nicht zu unterschätzende Tatsache erinnert: Arbeiten im Team macht Spaß! ■

Auflösung des Gewinnspiels der letzten QualityNews-Ausgabe:

Beim letzten Gewinnspiel musste folgende Gewinnfrage richtig beantwortet werden:

In einem Hafen haben vier Schiffe festgemacht. Am nächsten Tag verlassen sie gleichzeitig den Hafen.

Es ist bekannt, dass das erste Schiff alle 4 Wochen in diesen Hafen zurückkehrt, das zweite Schiff alle 8 Wochen, das dritte alle 12 Wochen und das vierte alle 16 Wochen.

Wann treffen alle Schiffe das erste Mal wieder in diesem Hafen zusammen?

- A: Nach 16 Wochen
- B: Nach 24 Wochen
- C: Nach 48 Wochen

Die richtige Antwort lautet:
C: Nach 48 Wochen

Der Gewinner ist...

Der Gewinner eines Exemplars des Buchs „More Agile Testing“ von Lisa Crispin & Janet Gregory ist

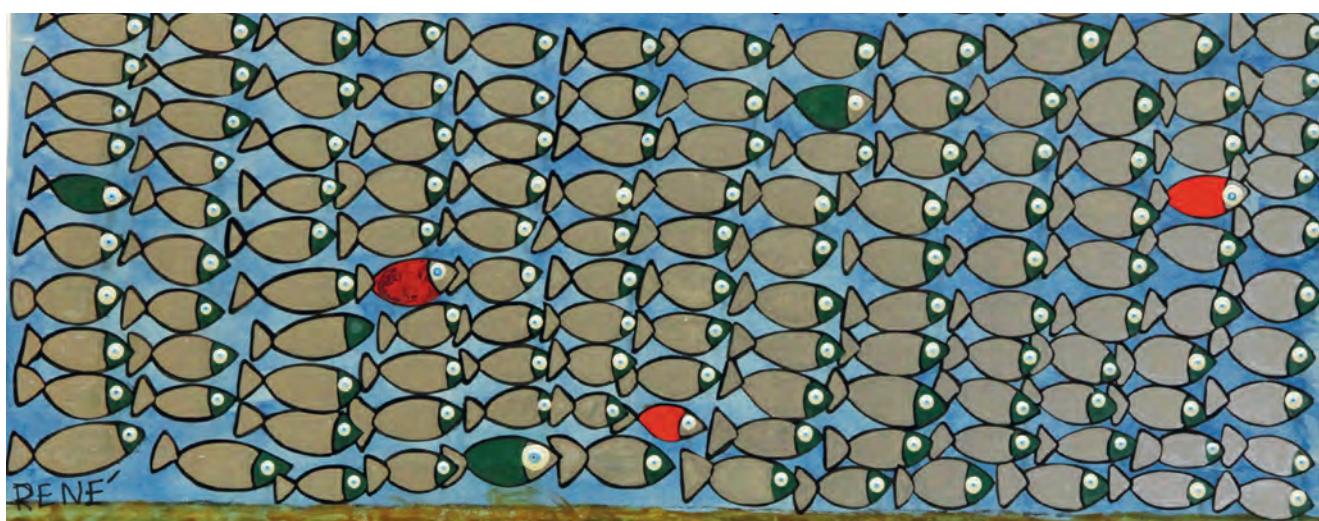
Reinhard Höfler, VB Leasing Finanzierungsgesellschaft m.b.H.

Das QualityNews-Team gratuliert herzlich!

Nicht vergessen: Auch dieses Mal gibt es wieder einen tollen Preis zu gewinnen!

Nutzen Sie Ihre Chance und gewinnen Sie einen Gutschein der „Time Busters“ für 2 Personen!

Das Gewinnspiel dieser Ausgabe finden Sie auf Seite 16.



Titel: „Die Fische im Meer“, Künstler: Rene Niedereder, Technik: Gouache mit Lack auf Leinwand



Anforderungsmanagement meets Software Test: Erfolgsfaktoren

von Alexander Weichselberger

Rubrik „SEQISenf“

Die persönliche SEQIS Kolumne mit Praxisnahem, Ernsthaftem ... und nicht immer zu 100% Ernstzunehmendem :-).

... idealer geht's kaum: Man sitzt mal da (Systemanalyse¹) und mal dort (Software Test) – und jetzt gilt es einen Artikel zu verfassen, der beide Disziplinen gegenüberstellt. Der geneigte Leser erwartet sich selbstredend – wie es der Artikeltitel ankündigt – *Erfolgsfaktoren*.



Quelle:
©SEQIS Software Testing GmbH

Vordergründig keine große Sache, **das kann man ja im Schlaf!** Okay, wie das tun? Also, einfach ausreichend Melatonin-Konzentration im Blut aufbauen (durch Licht aus). Damit geht die Herzfrequenz runter, Temperatur und Blutdruck sinken ebenfalls. Jetzt Augen zu und ab in die erste Tiefschlafphase.



Quelle:
©SEQIS Software Testing GmbH

Wir wissen, alle Sinne bleiben im Schlaf aktiv und somit sollten wir in diesem Zustand wiedergeben können was ohnedies jeder weiß:

1. Ausreichend richtige Dokumentation
2. Verständnis für ein „Miteinander“ i.S. von ungeteilter Verantwortung für den Erfolg und Prinzip des Helfens
3. Kontinuierliche Abstimmung im Rahmen der Entwicklung der Lösung

Nun, und wie machen wir das?
Richtige Dokumentation: Zumindest just enough, aber abgesichert; fragen Sie sich und andere im Team, was dokumentiert werden muss (i.S. von Menge, Art, ...) und vor allem: Für wen und welchen Zweck! Vermeiden Sie insbesondere Dokumentation, die „wir immer schon gemacht haben, aber nicht mehr wissen, wofür“. Ich selbst habe in meiner ersten Zeit als Datenbankadministrator im wahrsten Sinne des Wortes tonnenweise Papier bedru-



Mag. Alexander Weichselberger hat seine Einsatzschwerpunkte in den Bereichen Systemanalyse, Softwaretest, Koordination und Management von exponierten Großprojekten. Dieses Wissen gibt er auch gerne in Form von Coaching, Methodentraining und Fachvorträgen weiter.

Zahlreiche, erfolgreich abgeschlossene IT-Projekte führten ihn in verschiedenste Branchen und Länder, wo er auch mit multikulturellen Teams zusammenarbeitete. Er ist Mitglied der SEQIS Geschäftsleitung sowie Vorstand im Austrian Testing Board.

cken lassen – mit Listen und Reports, die keiner wirklich mehr gelesen hat. Bei Beschreibungen von Applikationen fokussieren Sie sich bitte eher darauf, was in der Applikation selbst nicht zu sehen ist (z.B. Feldfunktionen und Geschäftsregeln), weniger bzw. gar nicht, was ohnedies jeder sieht. Hier ist die Applikation selbst die beste „Dokumentation“.

Verständnis für einander, Prinzip des Helfens: Versuchen Sie ernsthaft eine gemeinsame „Haftung“ für den Erfolg zu vereinbaren. Wenn Analys-

¹ Systemanalyse, genau gesagt Systemanalyse 3.0 beinhaltet für mich Business Analyse (-> <http://www.iiba.org/babok-guide.aspx>), Requirements Engineering (-> <https://www.ireb.org/de>) und noch ein bisschen mehr

ten, Programmierer und Tester („The Three Amigos“²) gemeinsam für das Endergebnis gerade stehen müssen, müssen Sie verstehen, was der jeweils andere macht, wann er was wie braucht – und Sie selbst werden sich auch fragen, wie Sie die anderen am besten unterstützen können, insbesondere, wenn's bei denen „zwickt“. Eine Abgrenzungspolitik à la „ich bin fertig, jetzt sind die anderen dran“ hilft erfahrungsgemäß nicht.

Kontinuierliche Abstimmung: Anforderungen sind am Start nicht zu 100% formulierbar und fixierbar. Eine User Story z.B. ist eine Einladung zum Gespräch, nicht mehr und nicht weniger. Wir wissen zum „Gesprächsbeginn“ worum es geht – aber haben noch viel Klärungsarbeit vor uns (z.B. Definition der Akzeptanzkriterien), werden die jeweils andere Sicht nutzen um ein noch besseres, griffigeres und abgestimmteres Zielbild zu entwickeln. Am Ende der Abstimmung steht die analysierte, programmierte, getestete und abgenommene Funktion. D.h. planen Sie viele Treffpunkte zum Gespräch ein und nutzen Sie diese Termine für eine Präsentation des jeweils aktuellen Standes. Damit vermeiden Sie Überraschungen und kommen pünktlich und abgestimmt ins Ziel.

Aber betrachten wir mal die Kontrahenten **Software Tester** und **Systemanalysten** in diesem Spiel jeweils für sich; was würde man der jeweiligen „Seite“ an Tipps in der Zusammenarbeit mit den anderen mitgeben? Ein paar Überlegungen dazu folgend.

Software Tester

Seit Jahren wird das jeweilige „Inseln der Seligen“-Konstrukt, d.h. der Schaffensbereich der Systemanalysten einerseits und Tester andererseits, aufgebrochen. Es gibt kaum noch

Projekte, wo jeder für sich allein arbeitet. Selbst „unabhängige“ traditionelle Testteams können sich nicht mehr allein auf ihre ISTQB-Theorie und ihre Praxis im Software Test allein stützen. Durch die stärkere Verzahnung mit den Three Amigos sind wechselseitige Hilfe und damit Verständnis für die jeweils „anderen“ Disziplinen gefordert.

Ein Software Tester heute muss neben ISTQB³ und Softwaretest-Praxis viele weitere Kenntnisse mitbringen: Branchen-Know-how, Systemanalyse-Fähigkeiten und für z.B. Testautomation und Continuous Delivery (zumindest Einstiegs-) Programmierkenntnisse.

Konkret sind für die Software Tester durch die Einführung von „User Story's“ die Anforderungen an das System nicht mehr so umfangreich beschrieben, wie Vergleichsweise in Projekten mit Lasten- und Pflichtenheften. Dennoch waren und sind auch diese nicht wirklich ausreichend spezifizierend! Viele Aspekte müssen Software Tester im Kontext einfach schon wissen oder weiter erfragen. Dieses Wissen wird in Form von Testkonzepten, -plänen, -fällen und -durchführungsprotokollen dokumentiert. Leider sehr oft nicht mit dem Verständnis und Rückschluss, dass damit die zugrunde liegenden Pflichtenhefte erweitert bzw. detailliert werden müssen!

Würde ich einem Software Tester beschreiben müssen, was im Kontext mit Anforderungen auf Basis von User Story's wichtig ist, so würde ich das – in Ergänzung zu den oben stehenden Empfehlungen – auf eine Handvoll Tipps zusammenfassen:

- Ein „Als <Rolle> möchte ich <Ziel/Wunsch>, um <Nutzen>“ ist

³ Bitte nageln Sie mich nicht auf ISTQB fest – es gibt auch viele andere gute Wege, Software Test Theorie aufzubauen und sich prüfen zu lassen; aber wie auch immer Sie dazu stehen, ISTQB ist auch ein Weg ...

viel mehr als nur ein Satz. Der **Kontext** der User Story, wie die Verknüpfung zum Anwendungsfall, Informationen zur Schätzung, Priorität, ... bilden den Rahmen, der **Life-Cycle-Status** der User Story bestimmt den notwendigen Detaillierungsgrad. Blicke über den Tellerrand und kümmere dich einmal mehr darum, was nicht geschrieben steht!

- ... fundamental und wichtig: **Akzeptanzkriterien!** Am besten auch in Form von Templates, wie z.B. BDD⁴. Daraus leitet man ab, was die User Story „done“ macht, was die „Nicht-Ziele“ sind und was zu testen ist. Halte an dieser Stelle unbedingt auch nach den nicht funktionalen Anforderungen Ausschau!

... richtig erkannt! Alle Tipps gelten auch für was-auch-immer-eine-Anforderungsbeschreibung: Einfach „User Story“ durch „Anforderung“ ersetzen!
:-)

Systemanalysten

Der Zusammenhang zwischen den Arbeitsergebnissen von Systemanalysten und Software Testern erschließt sich meiner Meinung nach im Kontext der Qualitätssicherung von Anforderungen und bei der Verwendung von Anforderungen für den Soll-Ist-Vergleich, also der Ableitung von Testfällen aus den Anforderungen.

Anforderungen, Ziele und Aufgabenstellungen kommen in der Wertschöpfungskette „vorne“ rein, werden Schritt-für-Schritt realisiert und dabei konkretisiert, ggf. auch verändert und müssen am Schluss eine Aussage zu Soll und Ist ermöglichen. Richtig und vollkommen klar ist somit die Feststellung, dass eine Anforderung eigentlich

⁴ BDD – Behavior Driven Development; Akzeptanzkriterien werden dabei in Form von Sätzen wie „Angenommen <Vorbedingungen>, wenn <Aktion>, dann <Ergebnis>“ formuliert

² The Three Amigos – „Business Analyst, Developer and QA“; nach: <https://www.scrumalliance.org/community/articles/2013/2013-april/introducing-the-three-amigos>

frühestens dann „done“ ist, wenn die Software live gegangen ist.

Ob agil oder nicht – die Nachverfolgbarkeit von Anforderungen ist aus Qualitätssicherungssicht fundamental. Und natürlich brauchen wir auch testbare Anforderungen. Okay, also was bedeutet das im Detail?

Nachverfolgbarkeit von Anforderungen

... nochmals zu den **User Story's**, die eigentlich nur einen sehr groben Rahmen und allgemein gesprochen einen Platzhalter für Anforderungen bieten: Sie erheben keinen Anspruch auf Vollständigkeit. Erst durch die Akzeptanzkriterien werden die User Storys konkretisiert bzw. das Verhalten der User Story aus Sicht des zugrundliegenden Geschäftsmodells klar („Confirmation“, siehe nachstehende Grundregeln zur Erstellung einer User Story).

Grundregel zur Erstellung einer User Story – die 3 c's (Ron Jeffries):

Card: Eine User Story sollte so kurz und prägnant sein, so dass sie auf einer Karteikarte Platz hat

Conversation: Es sollte mehrfacher persönlicher Austausch über eine User Story zwischen Kunde/Product Owner und Entwicklerteam stattfinden. Die Kommentare werden meist auf der Rückseite der Karte erfasst

Confirmation: Durch die Festlegung von Akzeptanzkriterien kann die erfolgreiche Umsetzung einer User Story vom Kunden abgenommen werden (inkl. der anderen DoD Kriterien)

*Quelle: frei nach
<http://ronjeffries.com/xprog/articles/expcardconversationconfirmation>*

Akzeptanzkriterien unterstützen bei der Abgrenzung (In/Out of Scope?), bei der Konkretisierung bzw. Präzisierung (Fachbereich bzw. Projektmanager/PO machen ihre/seine

Erwartungshaltungen klar bzw. helfen sie dem Entwicklungsteam genau zu verstehen, was Inhalt der User Story ist) und sind Bindeglied zwischen User Story und Testfällen.

Akzeptanzkriterien sind so lange zu erheben bis alle Fragezeichen zu einer spezifischen User Story aufgelöst sind. Werden Fragezeichen nicht aufgelöst, exponiert man sich durch mangelhafte Qualität -> Risiko Kundenzufriedenheit (Termintreue und Qualität). By-the-way: Akzeptanzkriterien können sich auch auf andere Elemente der Wertschöpfung beziehen: Epics und andere Iterationsfeatures werden oft mit Akzeptanzkriterien versehen. In der Literatur und gängigen Praxis bedeutet „Nachverfolgbarkeit“ (oder „Traceability“) von Anforderungen:

Die Darstellungen der Beziehungen von Anforderungen

- ⇒ zu anderen Anforderungen (welche Anforderung trägt zur Erfüllung welcher anderen Anforderung bei),
- ⇒ zu den Quellen der Anforderungen (also wo kommen sie her: Aus Dokumenten, Workshops, Gesprächen...) und
- ⇒ zu den Entwicklungs-Artefakten (also Testfälle, Programmcode)

Demzufolge spricht man von „Pre-Requirement Specification Traceability“, „Traceability zwischen Anforderungen“ sowie „Post-Requirement Specification Traceability“. Die Nachverfolgbarkeit kann man mit verschiedenen Mitteln sicherstellen. Denk daran, dass eine entsprechende Toolchain dabei unterstützt!

Voraussetzungen für die Nachverfolgbarkeit sind auch,

- dass bei jeder Anforderung die Quelle festgehalten wird,
- dass die Anforderungen zueinander in Beziehung gesetzt werden
- und dass auch die Testfälle und Programming-Tickets in Beziehung zu den Anforderungen gesetzt werden.

Egal ob das Projekt traditionell oder iterativ entwickelt wird: Anforderungen sind generell aktiv zu managen!

Anforderungen ändern sich über die Projektlaufzeit, Anforderungen kommen hinzu oder werden ausgeschieden – in Wirklichkeit immer und zu jeder Zeit.⁵

Im Laufe des Projekts wächst im Regelfall der Zeitdruck und „Abkürzungen“ in den vorgegebenen Prozessen verlocken „ein Detail mal schnell abzuklären“.

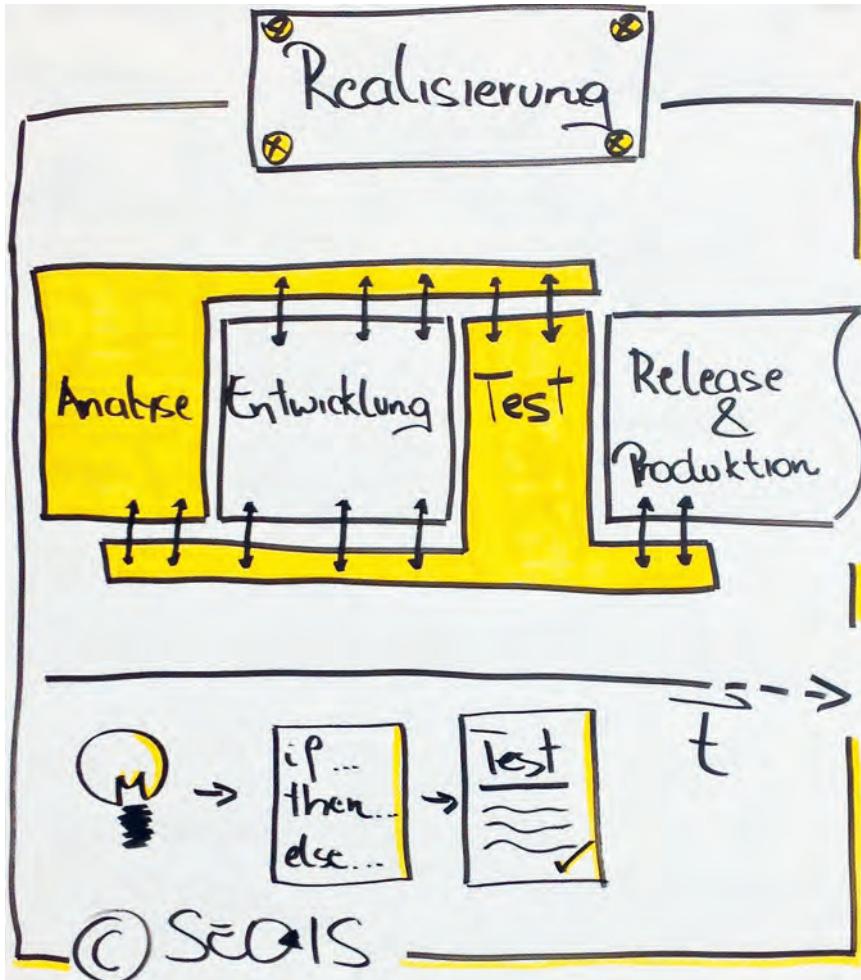
Typische Konsequenzen aus „Abkürzungen“ beim Change

- ! Prozesse aufgebrochen
- ! Dokumentation vernachlässigt
- ! Entwicklung auf Zuruf
- ! Unklarheit und Unsicherheit

Vor diesem Hintergrund empfehle ich die Nominierung eines Prozessverantwortlichen: Er/sie kann projektphasenbezogen die Standardprozesse hinterfragen (Mehrwert gegeben?) und ggf. anpassen (verschlanken?). Dennoch ist von grundlegender Bedeutung, dass Disziplin und ganzheitliches Denken eingehalten werden.

Ohne strukturierte Erfassung und Verwaltung von Anforderungen entsteht zwangsläufig Chaos! In iterativen Projekten verschärft sich diese Problematik noch weiter. Allgemein geht man davon aus, dass Anforderungen nicht wie in traditionellen Modellen, wie z.B. beim V-Modell XP, zu Beginn des Projekts umfassend erhoben und festgelegt werden.

⁵ Dies trifft auch auf klassisch geführte Projekte zu, die auf Lasten- und Pflichtenheften aufgebaut sind – nach deren Implementierung sind die Systeme im Regelfall durch Changerequests zu aktualisieren



Quelle: ©SEQIS Software Testing GmbH

Agilen Methoden folgend werden Anforderungen zu Beginn nur grob erhoben -> ein Merker für ein ausstehendes Gespräch (= User Story) als ein Dokument, das alle Details enthält. Anforderungen werden erst dann detailliert, wenn sie kurz vor der Implementierung stehen („just in time“). Damit wird versucht, „waste“ im Sinne von noch nicht notwendiger Anforderungen zu vermeiden, die durch Änderungen wieder zu aktualisieren wären (-> agile Manifesto „respond zu change“).

User Stories erfordern ein Umdenken. Details werden nicht mehr weit im Vorfeld der Story ausgearbeitet und aufgeschrieben, sondern erst, wenn die User Story konkret wird.

Weitere Aspekte, die bei agilen Realisierungen berücksichtigt werden müssen:

- Kürzere Analyzyklen -> höherer subjektiver Zeitdruck
- Kleinere Stücke, task-orientiert -> mehr Artefakte
- Weniger Dokumentation -> Notwendige Klärung der Stakeholder Anforderungen, damit keine Dokumente später nachträglich erstellt werden müssen
- Mehr Kommunikation -> Know-how, wie Gespräche in Anforderungen gewandelt werden
- Auflösung der klassischen Rollenbilder (Analytiker – Entwickler – Tester) -> alle „dürfen“ Anforderungen erheben

Zusammengefasst ist von vielen kleinen jederzeit änderbaren Reali-

sierungseinheiten auszugehen, die zusammenzuhalten sind.

Testbare Anforderungen

Wir haben jetzt kurz über Nachverfolgbarkeit von Anforderungen gesprochen – aber wie schaut's mit der Testbarkeit von Anforderungen aus? Allgemein gesprochen sind Anforderungen nur dann sinnvoll testbar, wenn sie folgenden Kriterien entsprechen:

- Konsistent
- Vollständig
- Eindeutig (nur eine Interpretation der Anforderung möglich)
- Quantitativ formuliert (vgl.: „schnelle Antwortzeit“ kann nicht verifiziert werden)
- Praktisch verifizierbar (d.h. mit begrenzten Aufwand durchführbar)

Wie kann somit die Testbarkeit der Anforderung bei der Erhebung und Dokumentation gewährleistet werden? Konzepte, Konzeptmodelle, Geschäftsanwendungsfälle, funktionale und nicht funktionale Anforderungen werden das Ergebnis der Analysetätigkeit sein. Die Systembeschreibung wird aus allgemeinen Einleitungen, Use Cases und Features sowie Subfeatures (und Sub-Subfeatures) bestehen.

Werden die oben stehenden Kriterien (Konsistenz, Vollständigkeit, ...) eingehalten, ist Testbarkeit der Anforderung gegeben -> diese Kriterien müssen bei der Erhebung und Dokumentation berücksichtigt werden. Für die User Stories, die ja nur einen groben Rahmen bieten, sind Akzeptanzkriterien zur Konkretisierung festzulegen. Diese werden im Rahmen von Gesprächen mit den Stakeholdern definiert. Üblicherweise haben Projektleiter/POs bzw. die Featureteams viele Fragen, die zu einer ausreichenden Abgrenzung anhand von Akzeptanzkriterien führen.

Methodisch kann man auch die Ableitung von Fragen anhand der Analyse

| | | |
|---|---|--|
| User Story: „Als registrierter Benutzer möchte ich meine Profildaten speichern können, damit ich meine Daten nicht immer wieder neu eingeben muss.“ | | |
| Schritt 1: Schlüsselwörter identifizieren | Vollverb: speichern | Substantiv: Profil |
| Schritt 2: Schlüsselwörter in Fragenkatalog einsetzen | V1: Wer muss speichern? V2: Wann speichern ... | S1: Welche Inhalte kommen vor? S2: Welche KANN bzw. MUSS Kriterien gelten? ... |
| Schritt 3: Antworten finden, prüfen und diskutieren | V1: Nur der Nutzer, kein automatisches Zwischenspeichern V2: Nur, wenn alle Pflichtfelder ausgefüllt wurden... | S1: Name, Adress- und Kontaktdaten S2: Optional: Telefonnummern, alle anderen Pflichtfelder ... |
| Schritt 4: Akzeptanzkriterien definieren | A1: Die Profildaten müssen aus den Feldern Vorname, Familienname, Straßename, Hausnummer, Wohnort, Postleitzahl, Handynummer, Festnetznummer und Email-Adresse bestehen A2: Die Felder Handynummer und Festnetznummer sind optional auszufüllen ... | |
| Schritt 5: Testfälle spezifizieren | TF1: Ungültige Email-Adresse testen <ul style="list-style-type: none"> • Voraussetzung: Benutzer ist registriert, eingeloggt und die Seite der Profildaten aufgerufen • Testschritte: <ul style="list-style-type: none"> • 1. Werte in Pflichtfelder eingeben • 2. Ungültigen Wert in Email-Adresse eingeben • 3. Speichern Button drücken • Erwartetes Ergebnis: Speichervorgang nicht durchgeführt, DB ist leer, Fehlermeldung („falsch eingegebene Email-Adresse“) • Verwendete Testdaten: ... | |


© SEQIS Software Testing GmbH, 2014

Abb.: Methodische Ableitung von Testfällen aus User Story's anhand von Schlüsselwörtern und W-Fragen
(Quelle: ©SEQIS Software Testing GmbH)

der Schlüsselwörter mit Fragenstellungen („W“-Fragen) absichern. Wie in obiger Abbildung exemplarisch ausgeführt, basiert die Methode darauf Schlüsselwörter aus den User Storys abzuleiten und mit einem vorbereiteten Fragenkatalog⁶ zu hinterfragen. Abhängig von den Antworten werden Akzeptanzkriterien erstellt, wobei mehrere Antworten zu einem Akzeptanzkriterium zusammengefasst werden können. Abschließend sind nur noch korrespondierende Testfälle zu formulieren.

Empfehlung: Da die User Story's mit ihrer Erfüllung („Done“) am Ende ihres Lifecycles sind, ist zu bewerten: In welchem Umfang sollten diese künftig im Rahmen des Regressionssets weiterverwendet werden, wobei davon auszugehen ist, dass die Testfälle bei diesem Schritt auch nochmals verändert werden.

⁶ Anwendung des Fragenkatalogs solange Fragestellungen im Kontext sinnvoll sind, bzw. wenn diese die User Story sinnvoll ergänzen bzw. konkretisieren

Durch diesen Re-Use werden explizit die systembeschreibenden Artefakte (Use Case und Features) zyklisch getestet.



Quelle:
©SEQIS Software Testing GmbH

Wrap up
Ja, zwischen Analysten und Testern bestehen viele Potentiale für Erfolgs-

faktoren; viele Artefakte aus der Analyse werden 1:1 im Test verwendet, viele Ergebnisse aus dem Test erlauben eine Prüfung der Analyseüberlegung auf Tauglichkeit und Erfüllung. Man braucht sich de facto wechselseitig.

Eine gemeinsame Verantwortung aller Akteure für den Erfolg, Helfer-Prinzip, Blick über den Tellerrand bei Kenntnissen der jeweils anderen Domäne und abgestimmte Artefakte bringen es – und führen immer zum Erfolg! Garantiert! Natürlich nur dann, wenn man auch die Entwickler und den Projektauftraggeber und, und, und, ... einbindet. Aber das ist jetzt eine andere Story :-)! ■

Ihre Meinung ist uns wichtig!
Benötigt ein Software Tester auch Kenntnisse aus dem Bereich Anforderungsanalyse? Und muss ein Business Analyst auch Software testen können?
Sagen Sie uns Ihre Meinung:
blog.SEQIS.com
facebook.com/SoftwareTestIsCool



Titel: „Ein schöner Tag“, Künstler: Rene Niedereder, Technik: Acryl- und Papier-Collage auf Leinwand

Fortsetzung von S. 6.

Wie sich vermuten lässt und auch eine unserer vorhergehenden QualityNews Ausgaben (www.SEQIS.com/unternehmen/news) bereits veranschaulicht hat, spielen Schnittstellen zwischen diesen Komponenten eine sehr zentrale Rolle.

Bei Erstellen einer neuen Anforderung stellt sich nun die Frage: Ist es Aufgabe der Analyse-Rolle Know-how und Dokumentationsbasis über die doch sehr detaillierten technischen Details aufzubauen? Oder ist es eher die Rolle des Entwicklers, der, wie bereits angesprochen, ja die Anforderung (was zu erledigen ist) umsetzt (wie es erledigt)? Was fängt dann jedoch im Folgenden der Tester mit einem programmiernahen Modell an, wenn durch zu viel Details der inhaltliche Konsens auf fachlicher Ebene fehlt?

Für unseren Analysten ist klar, dass der Blick über den Tellerrand keine Fleißaufgabe sondern eine Notwendigkeit darstellt. Zu wissen, welche

Parameter einer Schnittstelle entscheidend für das Ergebnis eines Algorithmus sind, einzelne Fehlercodes zu kennen und hin und wieder auch über die Entwickler-Schulter selbst in den Programmcode zu blicken, ist gut um zu verstehen und um auf denselben Wellenlänge kommunizieren zu können.

Weiters verbirgt sich oft noch auf solch einer Detailtiefe Potential zu grundlegenden Design- oder Interpretationsfehlern, deren Behebungskosten ja bekanntlich früher im Entwicklungsprozess weitaus kostengünstiger sind.

Ein weiterer Test Consultant und Requirements Engineer bereichert die Runde durch seine Erfahrungen aus Sicht des Entwicklers: „Als Entwickler will ich nicht, dass mir jemand Algorithmen oder Datenmodelle vorgibt, jedoch möchte ich gerne Schnittstellendefinitionen, Mock-Ups oder auch Use-Cases erhalten.“ Eine klare Ansage also, die im Grundsatz dem zuvor beschriebenen Ansatz folgt und auch

von unserem Systemanalyse 3.o-Fachmann unterstützt wird. Er hält die Sache so, dass Analyseergebnisse, die doch stark auf die technische Kompetenz hin zeigen, ganz klar als Vorschläge deklariert sind.

Dadurch ist einer der möglichen Lösungswege auf jeden Fall schon mal vorhanden, Verbesserungsvorschläge haben aber immer Platz und sind auch erwünscht. Hier schließt sich nun auch wieder der Kreis zum notwendigen Maß an Kommunikation und Abstimmung, denn nur das gemeinsame Bild, resultierend aus unterschiedlichen Know-how Gebieten und Blickwinkeln, kann einer qualifizierten und konsolidierten Aussage des gesuchten „Bestmöglichen“ entsprechen.

Unser Praxistipp:

Gehen Sie stufenweise vor und brechen Sie vom Groben ins Feine. Das zusammenhängende Gesamtbild ist der Ursprung und spätestens in einem systemübergrei-

fenden Test ist das ursprünglich vorgestellte Bild mit dem tatsächlichen Outcome gegenüberzustellen und zu beurteilen.

Verlaufen Sie sich als Analytiker nicht oder zumindest nicht zu früh in Detail-Sackgassen. Aber: Seien Sie sich bewusst, dass Sie ein sehr wichtiger Know-how Träger sind und mit diesem Wissen sehr viel Gutes tun können. Helfen Sie, wenn möglich und nötig, auch bis in die kleinsten Aufgaben der Programmierarbeit mit. Persönlich zu konsultieren und schriftlich zu definieren sind immer noch getrennt behandelbar. Machen Sie diese Trennung auch im Anforderungsdokument ersichtlich, damit jede Rolle den jeweils nötigen Detail-level und die damit einhergehende Verbindlichkeit dem geschriebenen Wort ablesen kann.

Anforderer als stille Drahtzieher Ihres Projektes

Bei SEQIS agieren Business Analysten laufend im Projektalltag und sind somit unmittelbar und maßgeblich am Geschehen der Softwareentwicklung beteiligt. Unsere Erfahrungen von serverseitigen Anwendungen bis zur Mobile App zeigen, dass speziell im agilen Projektvorgehen das bloße Vorlegen von Anforderungsdokumenten ohne ein Einbinden in einen sinnvoll abgestimmten Prozess in vielerlei Hinsicht keine wünschenswerten Ergebnisse nach sich zieht.



Einer unserer Analysten musste feststellen, dass jede Definition nur so gut ist wie das Verständnis darüber, wie mit dieser umzugehen ist. Ist dies nicht gegeben, folgen nicht selten weitaus teurere Nacharbeiten aufgrund von unterschiedlicher Auffassung und zu geringer, manchmal aber auch zu detaillierter Tiefe der Anforderungsbeschreibung.

Letztlich muss vor allem der Tester ein klares Bild des Soll-Verhaltens haben, um überhaupt ein aussagekräftiges Urteil über den Qualitätsstand abgeben zu können. Inhaltlich korrekte aber unvorteilhaft beschriebene Anforderungen sowie der fehlende „rote Faden“, die dieses Verständnis trüben und dadurch die Unabhängigkeit des Testers angreifen, können im Extremfall erheblichen Schaden in einer späteren Phase des Produktlebenszyklus verursachen, deren Behebungskosten wiederum vielfach höher sind, als sie es zum Zeitpunkt der Anforderungsdokumentation gewesen wären.

Unser Systemanalyse 3.0-Experte steuert der vorangehenden Diskussion einen weiteren ernstzunehmenden Faktor bei: Analyse ist per Definition auch ein Reifeprozess, daher ist eine der größten Gefahren, zu spät und an der falschen Stelle mit der Arbeit zu beginnen.

Soll in einem Team etwas entwickelt werden, muss schon zuvor eine gut gehärtete Konzept-Basis vorliegen, die dann in einem finalen Schritt verfeinert und umgesetzt werden kann. Alternativ ist es aber möglich, die Analyse schon im Grobstadium an ein Entwicklungsteam zu übergeben. Dies hat den Vorteil, dass die letztendlich Tätigen der Umsetzung schon früh einbezogen werden, um relevanten Input zu geben und so das Risikopotential frühzeitig einzudämmen. Bei diesem Vorgehen muss aber jedenfalls eine sehr gute übergreifende Überwachungsinstanz agieren, um auf der Metaebene weiterhin das richtige Ziel zu verfolgen.

Eine Erfahrung jedenfalls können viele der Anforderer von SEQIS bestätigen: Durch die so elementare Stellung eines Anforderungsdokuments hat dieses einen enorm großen Hebel auf Prozessparameter wie Zeit und Qualität.

Auch der Ersteller der Anforderung hat durch seine Eingliederung in der Prozesskette eine zentrale Stellung und damit ein breites Beobachtungsspektrum, das weit über die fachliche Systembeschreibung hinausgeht.

Die Anforderungen in Ihrem Projekt richtig umzusetzen (hier sei bewusst unterschieden zu „die richtige Anforderung umsetzen“) und die umgebenden Einflüsse und Abläufe individuell einzustellen und laufend zu reflektieren erfordert viel Feingefühl, Vorausblick und einen ausgeprägten Werkzeugkasten an Soft-Skills.

Jedoch kommt nach Forming, Storming und Norming ja bekanntlich das, was dem großen Bestreben entspricht: Performing.

Daher noch ein Praxistipp:

Investieren Sie gleichermaßen in ein qualitätsgesichertes Vorgehen wie in das daraus entstehende Produkt. Gerade in agilen Projekten ist oft ein hohes Maß an Energie anzutreffen, das erst einmal gebündelt und auf ein gemeinsames Ziel gerichtet werden muss.

Helfen Sie mit, dass das Team so geschützt und individuell wie möglich sein Fachwissen entfalten kann und geben Sie ausreichend Raum, die richtigen Schritte aus den Erfordernissen heraus entstehen zu lassen.

Reflektieren Sie Ihr Vorgehen und die Zufriedenheit der Ergebnisse aber auf jeden Fall laufend, um gegebenenfalls rechtzeitig korrigierend eingreifen zu können. ■

lebenshilfe

Niederösterreich



www.noe.lebenshilfe.at

Lebenshilfe Niederösterreich Werkstätten Mödling, Baden 1 und Baden 2

Sie sind auf der Suche nach außergewöhnlichen Geschenken und handgefertigten Einzelstücken?

Die Werkstätten Baden und Mödling der Lebenshilfe Niederösterreich bieten neben der Ausstellung auch den Verkauf von handgefertigten Einzelstücken wie Bilder, Grußkarten, Keramiken, Kerzen, Kunsthandwerk aus Holz, Metall & Glas, Mosaiken, Schmuck, Strickwaren, Taschen, uvm. Das gesamte Jahr über arbeiten die KlientInnen eifrig mit viel Leidenschaft und Begeisterung an den Kunststücken.

Das vorrangige Ziel der Werkstätten der Lebenshilfe Niederösterreich ist, den KlientInnen die Teilhabe am Arbeitsprozess zu ermöglichen. Der Schwerpunkt der Werkstätten liegt primär auf handwerklichen und künstlerisch-kreativen Tätigkeiten.

Kontakt:

Werkstätte Mödling:

Leitung: Bernd Hatter
Untere Bachgasse 7
2340 Mödling

Werkstätte Baden 1:

Leitung: Monika Waidhofer
Mariengasse 1
2500 Baden

Werkstätte Baden 2:

Leitung: Ursula Habres
Gutenbrunnerstraße 12
2500 Baden

Neu ab 2016: Termingarantie für ausgewählte Kurse!

Ab 2016 gilt für ausgewählte Kurse unseres Education-Portfolios eine Termingarantie!

Diese Kurse finden fix zum angegebenen Termin statt, unabhängig der Teilnehmerzahl – Sie können sich somit schon fix für einen (oder mehrere) dieser Kurse anmelden. Die Termine finden garantiert statt.

Die Termingarantie gilt für folgende Kurse:

- iSQI® CAT Certified Agile Tester
- ISTQB® Certified Tester, Foundation Level Extension, Agile Tester
- iSQI® Certified Agile Test Driven Development
- CMAP® Mobile App Testing, Foundation Level
- iSQI® Certified Agile Business Analysis

SEQIS Education...

- ✓ Termingarantie
- ✓ Frühbucherrabatt
- ✓ Internationale Zertifikate
- ✓ Topmodernes Equipment
- ✓ Erfahrene Trainer



Nächste Kurse mit Termingarantie:

- iSQI® CAT Certified Agile Tester:
25.04. - 29.04.2016, Mödling
12.09. - 16.09.2016, Mödling
- ISTQB® Certified Tester, Foundation Level Extension, Agile Tester:
10.03. - 11.03.2016, Mödling
17.11. - 18.11.2016, Mödling
- iSQI® Certified Agile Test Driven Development:
07.03. - 09.03.2016, Mödling
17.10. - 19.10.2016, Mödling
- CMAP® Mobile App Testing, Foundation Level:
18.04. - 19.04.2016, Mödling
07.11. - 08.11.2016, Mödling
- iSQI® Certified Agile Business Analysis:
21.03. - 22.03.2016, Mödling
10.10. - 11.10.2016, Mödling

Infos & Anmeldung
www.SEQIS.com

* Titel: „Profeulez“, Künstler: Kim Novak, Stefan Riesner, Katharina Stockreiter, Technik: Acryl auf Leinwand

SEQIS Portfolioübersicht

R Requirements

- Business Analyse & Requirements Management
- Systemanalyse 3.0 Coaching



C Coding & Tools



- Test Driven Development
- Continuous Integration
- Unit Tests für Legacy Code
- API Test Automation
- Tool Evaluierung

T Traditionelles Testing

- Functional Testing
- Functional Test Automation
- Test Management
- Quality Coach
- Non-functional Testing



A Agile Testing



- Agile Testing
- Session Based Testing
- Agile Quality Coaching

M Mobile Testing

- Mobile Testing
- Mobile Testing Coaching
- Mobile Testlab



L Load & Performance Testing



- Load & Performance Testing Experte
- Load & Performance Testlab

A Ausbildung/Kurse/Trainings

Zu all unseren Leistungen bieten wir international anerkannte Zertifizierungsschulungen und Trainings an.

