

Der frühe Vogel fängt den Wurm

Frühes Testen erhöht nicht nur die Qualität und vermeidet technische Schulden, sondern spart dabei auch Zeit und Kosten

von Marcel Langer und Metin Savignano

Bei der Softwareentwicklung unterscheidet man in der Regel vier Teststufen: Unit-, Integrations-, System- und Abnahme-Test. Als frühest mögliche Teststufe bringen Unit-Tests schon von Anfang an hohe Qualität in den Softwareentwicklungsprozess und verringern dadurch erheblich die Kosten für die Fehlerbehebung. Der Vorteil der Testmethode: Einzelne Softwarebausteine können isoliert getestet werden, ermittelte Fehler lassen sich somit nicht nur viel früher finden, sondern ihre Ursache auch sehr viel leichter einer Komponente zuordnen. Aber wie setzt man sie auf dem Mainframe optimal ein?

Das Testen von Software ist prinzipiell zeitaufwändig. Das trifft natürlich genauso auf Mainframes zu. Leider werden notwendige Funktionstests aus Zeitmangel oftmals nach hinten verschoben. Zugrunde liegt die irriige Annahme, der Gesamtaufwand sei geringer, wenn nicht schon während der Entwicklung systematisch getestet werden muss, sondern erst nach der Fertigstellung der Software. Dass dies ein Trugschluss ist, zeigt sich erst, wenn im Abnahmetest zahlreiche Fehler entdeckt werden – vor allem dann, wenn es sich um Fehler im Design oder gar in den Anforderungen handelt. Im Gegensatz zu simplen Programmierfehlern erfordern solche Probleme für eine zukunftsorientierte Lösung fast immer eine umfassendere Analyse und Korrektur der Konzeption. Meist reicht aber die Zeit nicht mehr für grundlegende Korrekturen, und selbst neue Anwendungen gehen dann schon mit „technischen Schulden“ in Produktion. Für diese Schulden müssen dann teuer Zins und Zinseszins in Form von erhöhten Wartungs- und Weiterentwicklungskosten bezahlt werden.

Aber selbst wenn es zeitlich noch möglich ist, Anforderungs- und Designfehler zu korrigieren, ist dies signifikant teurer, als wenn dies schon früh während der Implementierung geschieht. Grundsätzlich gilt: Je später die Anforderungen oder das Design einer Software korrigiert werden, desto weiter muss man in der Konzeption zurückgehen und bereits Erreichtes wieder verwerfen. Diese Mehrfach-Arbeit verschlingt Budget, das andernfalls für mehr Funktionalität zur Verfügung stehen könnte.



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haaßengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de

Auch sollte hierbei nicht die psychologische Auswirkung unterschätzt werden, wenn beispielsweise Projektmitarbeiter erkennen, dass ihre Arbeit zum Teil verworfen werden muss. Das erzeugt nicht nur Frust, sondern führt mitunter dazu, dass die erforderlichen grundlegenden Anpassungen lieber vermieden werden. Die Problemlösung findet dann stattdessen über zusätzliche Sondervarianten statt. Das schlägt sich dann wiederum auf die Kosten für Wartung und Weiterentwicklung der Software nieder.

Für das Risikomanagement eines Projektes sind frühe Tests ein Vorteil, den man gar nicht hoch genug einschätzen kann. Fehler, die erst kurz vor der geplanten Produktivsetzung entdeckt werden, können insbesondere bei Fixterminen wie zum Beispiel gesetzlichen Änderungen ein hohes Risiko darstellen. Um diese zu vermeiden, sollte die Früherkennung eine strategisch bedeutsame Rolle spielen.

Was also tun?

Es ist illusorisch, diese Probleme dadurch in den Griff bekommen zu wollen, dass man mehr Zeit in detaillierte Voruntersuchungen steckt oder die Anforderungen vor dem Start der Softwareentwicklung noch umfassender und detaillierter zu spezifizieren versucht. Nicht nur liegt es daran, dass es in der menschlichen Natur liegt, sich zu irren. Es ist häufig auch so, dass die Gesamtheit der Anforderungen komplexe Wechselwirkungen beinhalten, deren Effekte erst erkannt werden, wenn diese umgesetzt werden.

Um diese Problematik in den Griff zu bekommen, gibt es nur einen sinnvollen Weg: Mit den Tests so früh wie möglich beginnen, damit Probleme rechtzeitig identifiziert werden können. Nur auf diese Weise lassen sich Fehler in der Software möglichst zeit- und kostensparend beheben. Im Gegensatz zu den in Abnahmetests üblichen Systemtests, bei denen die Anwendung in ihrer Gesamtheit geprüft wird, erlauben Unit-Tests, einen extrem frühen Testlauf. Darüber hinaus sichern sie durch ihre Wiederverwendung als Regressionstests über den gesamten Projektverlauf, dass es bei der Weiterentwicklung für bestehende Funktionalität nicht zu ungewollten Seiteneffekten kommt und sich dort Fehler einschleichen. Diese Verwendungsmöglichkeit beschränkt sich nicht nur auf die eigentliche Projektdauer, sondern gilt auch dauerhaft für die anschließende Wartung und Weiterentwicklung. Unit Tests bieten somit die einzigartige Möglichkeit, sie einerseits schon besonders früh, andererseits aber auch besonders nachhaltig einzusetzen.



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haasengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de

Unit-Tests im Einsatz

Schon in den 1970er-Jahren zeigten Analysen, dass die Kosten für die Behebung von Fehlern in der Softwareentwicklung davon abhängen, in welcher Phase sie entdeckt werden. In einer entsprechenden Erhebung wurde nachgewiesen, dass für Fehler in den Anforderungen, die erst beim Abnahmetest entdeckt werden, durchschnittlich der fünffache Zeit- und Kostenaufwand erforderlich ist, wie für solche, die schon während der Implementierung entdeckt werden. Bei größeren Projekten war der Effekt sogar noch stärker.

Im Vergleich zu Systemtests ermöglichen Unit-Tests, die Kosten für die Fehlerbehebung um bis zu 80 Prozent zu senken. Diese Kostenreduzierung beruht hauptsächlich auf der Tatsache, dass Fehler schon sehr viel früher gefunden werden. Zudem führt dies dazu, dass die Software sehr viel stabiler in den Abnahmetest geht und dort signifikant weniger Aufwand verursacht.

Daran, dass es ökonomisch absolut sinnvoll ist, Anforderungen so früh wie möglich zu überprüfen, zweifelt wohl niemand. Um der Problematik mit den hohen Kosten für spät entdeckte Anforderungs- und Designfehler beizukommen, setzen sich inzwischen auch im Mainframe-Umfeld immer mehr dieser Vorgehensmodelle durch, die durch einen inkrementell-iterativen Ansatz möglichst früh das Erkennen und Beheben von Fehlern ermöglichen wollen. In der agilen Softwareentwicklung ist es ein Grundprinzip, möglichst schnell und flexibel auf Probleme einzugehen und Änderungen willkommen zu heißen, da dies bedeutet, Software zu entwickeln, die dem Kunden einen noch höheren Nutzen bringt. Hierfür ist es es aber wichtig, in jeder Iteration Tests auszuführen, um auf diese Weise sicherzustellen, dass sowohl die in der aktuellen Iteration implementierte Funktionalität (das Inkrement) als auch die bestehende Funktionalität korrekt sind. Es stellt sich die Frage, wie man in der agilen Softwareentwicklung frühes Testen ermöglicht. Ein Ansatz ist, in jeder Iteration eine vollständige Anwendung zu bauen und einen Systemtest durchzuführen. In großen Projekten, wie sie in Mainframe-Umgebungen typisch sind, stellt sich das jedoch oft als unrealisierbar heraus. Hier sind Tests auf Komponentenebene sinnvoll. Dabei sind Unit-Tests sowohl für die Entwicklertests als auch für Komponenten- und Integrationstests ideal geeignet.

Eine bekannte Bank etwa hat das Thema schon vor Jahren erkannt, als dort begonnen wurde, agile Softwareentwicklung einzusetzen. Den Schlüssel zur agilen Softwareentwicklung fand



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haaßengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de

man in den Anforderungen an die Testbarkeit. Um diese zu erfüllen, sollten Unit Tests eingesetzt werden. Um den Entwicklern das Testen mit der erforderlichen Effizienz zu ermöglichen, sollten hierzu keine handgeschriebenen Tests, sondern ein Tool verwendet werden. Eine Marktanalyse ergab zum damaligen Zeitpunkt, dass kein Standardtool verfügbar ist. Bei der Bank wurde daher ein eigenes Tool entwickelt und aufgrund seines erwiesenermaßen hohen Nutzens auch bis heute weiterentwickelt.

Dass die Kostenvorteile von Unit Tests nicht nur theoretisches Zahlenspiel sind, zeigt auch die Analyse eines Projekts bei einem großen Finanzdienstleister. In einem iterativ-inkrementell umgesetzten Projekt, bei dem konsequent und regelmäßig Unit Tests mit Hilfe eines Standardtools durchgeführt wurden, ging der Abnahmetest ungewöhnlich problemlos von der Hand. Um diese Tatsache sowie auch die positive Einschätzung der Projektmitarbeiter zu untersuchen, wurden die Aufwände für den Fachtest in Verhältnis zum Gesamtaufwand gesetzt und mit anderen Projekten vergleichbarer Größe und Komplexität verglichen, in denen Unit Tests jedoch nicht so konsequent zum Einsatz kamen. Das Ergebnis war überzeugend: Der Anteil für die Fach- und Abnahmetest konnte um 75% Prozent reduziert werden. Dies ist besonders bedeutsam unter dem Aspekt, dass Fachexperten aufgrund ihrer starken Auslastung praktisch immer sehr wertvolle Ressourcen darstellen. Zudem haben die Projektmitarbeiter den Abnahmetest einvernehmlich als besonders reibungsarm empfunden, weil die Qualität der Software von Anfang an schon sehr hoch war.

Im besten Fall laufen Unit-Tests, sowie Regressions- und Integrationstests automatisiert im Rahmen eines Standbaus ab, wobei die Entwickler per E-Mail informiert werden, wenn ein Test nicht erfolgreich verlaufen ist. Dass Effizienz und Kosteneinsparungen aber auch ohne eine Vollautomatisierung möglich sind, zeigt obiges Beispiel, bei dem die Ausführung der Unit-Tests und die Benachrichtigung der Entwickler nach dem Standbau noch manuell angestoßen wurde. Es ist äußerst ratsam, den Einsatz regelmäßiger Unit-Tests nicht von einer Vollautomatisierung abhängig zu machen, sondern möglichst frühzeitig damit zu beginnen. Denn auch dann wird die Softwarequalität aufgrund der ständigen Überprüfung erheblich besser ausfallen und die frühe Fehlererkennung und -behebung wird zu reduzierten Kosten führen. Insbesondere wenn Kernanwendungen betroffen sind oder eine Terminverschiebung undenkbar ist, sind Unit-Tests ein wichtiger Aspekt zur Risikominimierung. Denn nur durch eine frühe Fehlererkennung erhält man den für ihre Behebung notwendigen zeitlichen Spielraum.



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haasengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de

Besseres Projekt-Controlling

Unit-Tests erlauben zudem eine detailliertere und objektive Sicht auf den Projektfortschritt. Um den Projektfortschritt zu messen, wird meist der geleistete Aufwand im Verhältnis zum Gesamtaufwand mit dem Umsetzungsanteil verglichen. Daran lässt sich aber lediglich das bislang aufgewendete Budget ablesen. Um zu ermitteln, wie weit die Umsetzung ist, können nur die Projektmitarbeiter befragt werden. Diese schätzen jedoch aus verschiedenen Gründen den Fortschritt ihrer Arbeit sehr unterschiedlich ein – manche sind zu vorsichtig und wollen nicht zu viel versprechen, weil zum Beispiel noch unvorhergesehene Änderungen erforderlich sein könnten; andere sind grundsätzlich optimistisch und meinen stets fast fertig zu sein. Dieser subjektive Anteil in der Schätzung muss erkannt und ausgeglichen werden, da sonst keine realistische Aussage über den Projektfortschritt getroffen werden kann. Unit-Tests hingegen erlauben eine wesentlich objektivere Messung der umgesetzten Funktionalität. Zudem reduzieren sie aufgrund der frühen Fehlerkennung das Risiko, zu einem späten Zeitpunkt auf Probleme zu stoßen, deren Behebung einen hohen – und ungeplanten – Aufwand erfordert. Mit Unit-Tests kann man also den inhaltlichen Fortschritt wesentlich genauer bestimmen und erreicht sozusagen ein Funktionalitätsmonitoring über das gesamte Entwicklungsprojekt.

Fazit: Mit dem Einsatz von Unit Tests kann eigentlich nichts falsch gemacht werden – aber vieles richtig!

Vorgehensbeispiel zur Auswahl eines geeigneten Testtools

Mit der Frage, wie man all die vorgenannten Vorteile von Unit Tests auf einem z/OS Mainframe für die Cobol-Entwicklung effizient nutzen kann, hat sich Marcel Langer im Rahmen seiner Bachelor-Thesis beschäftigt. Er hat dabei insbesondere untersucht, ob es für IT-Anwendungen auf Mainframes im Bankenumfeld bereits unterstützende Tools gibt und wenn ja, welche davon am besten geeignet sind.

Nachfolgend wird beschrieben, wie die Auswahl eines geeigneten Unit-Test-Tools in der Praxis durchgeführt werden kann. Diese Vorgehensweise ist exemplarisch zu verstehen. In unterschiedlichen Projekten und Unternehmen werden die Anforderungen und Priorisierungen im Detail abweichen.

Ausgangssituation

Im ersten Schritt zur Auswahl eines geeigneten Testtools sollte



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haasengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de

zunächst die Ausgangssituation näher beleuchtet werden. So wurden in einem Fall im Anschluss an die Implementierung eines Cobol-Moduls automatisierte Unit-Tests erstellt. Dazu wurde zu jedem Modul ein Testtreiber entworfen. Die Form dieser Testtreiber war nicht weiter definiert und wurde daher von den Entwicklern auf unterschiedliche Weise umgesetzt. In einem Projektbereich wurden die Testtreiber beispielsweise jeweils in einem zusätzlichen Modul implementiert. In diesem Modul wurden Testfälle mit konkreten Testdaten erfasst, um die Aufrufchnittstelle des Testobjektes zu füllen. Nach der Ausführung des zu testenden Moduls wurden die gelieferten Ergebnisse automatisiert gegen die im Testtreiber definierten Sollwerte geprüft und die Ergebnisse der erfolgreichen und der fehlerhaften Testfälle in einer Liste dokumentiert. Eine Variation dieser Vorgehensweise bestand darin, die Daten und Prüfwerte zu den Testfällen in einer Datei zu erfassen und von dort einzulesen. Bei der Erstellung der Testtreiber wurden keine unterstützenden Werkzeuge eingesetzt. In einem Interview mit einigen Entwicklern wurden die nachfolgend aufgeführten Probleme identifiziert, die in dem derzeitigen Prozess der Erstellung von Unit-Tests aufgetreten sind.

Probleme

Die manuelle Erfassung der Testdaten direkt im Programmcode ist sehr unübersichtlich und beansprucht viel Zeit. Bei einer großen Menge an Daten schleichen sich dadurch schnell Fehler ein. Das führt dazu, dass bei den ersten Ausführungen der Tests im Fehlerfall unklar ist, ob sich das Testobjekt falsch verhält, oder die Testdaten nicht korrekt sind.

Ein weiterer Nachteil bei dieser Art von Testtreibern ist, dass die Testfallerfassung sehr technisch basiert erfolgt. Aus diesem Grund müssen Programmierkenntnisse in Cobol vorhanden sein, um Tests zu erstellen oder zu ändern. Dieser Aspekt führt zu Abstimmungsschwierigkeiten zwischen den Entwicklern und den Fachspezialisten, die teilweise die Testdaten vorgeben. Außerdem führen fehlende Initialisierungen zwischen den verschiedenen Testfällen zu Fehlern. Diese sind, falls sie nicht zu einem Abbruch führen, oft nur aufwendig lokalisierbar. Die Änderung der Testobjekte zieht eine aufwendige Anpassung der Testtreiber nach sich, besonders wenn es sich um eine Schnittstellenänderung handelt. Insgesamt benötigt die Erstellung von Testtreibern in dieser Art oft mehr Zeit, als die Entwicklung des zu testenden Programms.



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haasengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de

Das ideale Tool

Anhand der identifizierten Probleme wurde eine Testaufgabe definiert, die ein ideales Tool erfüllen müsste.

Ein ideales Werkzeug unterstützt den Entwickler bei der Erstellung der Testfälle durch eine vereinfachte Testdatenerfassung sowie bei der Definition eines Ausgangszustandes, um eine Reproduzierbarkeit und Wiederholbarkeit der Tests zu erreichen. Es erleichtert außerdem den Vergleich zwischen Soll- und Ist-Werten durch geeignete Funktionen. Außerdem kann eine automatische Testausführung auf Knopfdruck angestoßen werden. Das ideale Werkzeug kann Testfälle in Testsuiten zusammenfassen. Dadurch können neben der Ausführung von einzelnen und allen Tests auch nur bestimmte Gruppen geprüft werden. Im Anschluss an die Durchführung stellt das Werkzeug die Ergebnisse der Tests übersichtlich dar, damit auf einen Blick erkennbar ist, welche Testfälle korrekt und welche fehlerhaft sind.

Anforderungen erheben

Um in einem Vergleich von Tools zu einem aussagekräftigen Ergebnis zu kommen, ist die Spezifikation von genauen Anforderungen notwendig.

Hierzu wurden vom Autor zunächst auf Basis der Testaufgaben einige Anforderungskriterien festgelegt. Anschließend fand auf Basis von Interviews mit allen identifizierten Stakeholdern eine Anpassung und Ergänzung des Anforderungskatalogs statt. Als Stakeholder wurden hierbei sechs Entwickler als Zielgruppe eines solchen Werkzeugs identifiziert. Um auch eine umfassende Sichtweise des Entwicklungsprozesses zu berücksichtigen, wurden zudem zwei Führungskräfte einbezogen. Im Rahmen des Interviews wurden in einer offenen Frage zusätzlich weitere Anforderungskriterien ermittelt. (siehe Tab. 1)



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haassengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de

Tab. 1:

| Kriterium | Priorität (1=sehr wichtig 2=wichtig 3=weniger wichtig) | K.-O.? (ja/nein) |
|--|---|---------------------|
| Integrierbarkeit in die Host-Umgebung | | |
| Einmalige Anbindung des Testobjekts | | |
| Wiederholbare Anbindung des Testobjekts | | |
| Hilfeleistung bei der Testfallerstellung | | |
| Unterstützung durch Vergleichsarbeiten oder Assistenten | | |
| Erleichterung von Testfällen in Excel | | |
| Nachvollziehbarkeit der Testfälle durch einen funktionierenden Parser | | |
| Modularität von Komponenten des Programms für die Erstellung der Testfälle | | |
| Unterstützung der Erstellung eines Testmodells | | |
| Erleichterung der Testfallerstellung | | |
| Verwaltbarkeit der Testfälle | | |
| Einmalige Testfallerstellung | | |
| Wiederholbare Testfallerstellung | | |
| Erleichterung der Verwaltung der Testfälle | | |
| Erleichterung der Verwaltung der Testfälle bei Änderungen der Anwendung | | |
| Übersicht der Testergebnisse | | |
| Einmalige Übersicht über Teststatus über den gesamten Umfang der Testobjekte | | |
| Wiederholbare Übersicht über Teststatus über den gesamten Umfang der Testobjekte | | |
| Ausführung der Tests | | |
| Einmalige Ausführung der Tests über den gesamten Umfang der Testobjekte | | |
| Wiederholbare Ausführung der Tests über den gesamten Umfang der Testobjekte | | |
| Erleichterung der Ausführung der Tests über den gesamten Umfang der Testobjekte | | |
| Werkzeugeigenschaften | | |
| Anzahl der Möglichkeiten in der Testfallerstellung | | |
| Aufwand der Testfallerstellung | | |
| Flexibilität der Testfallerstellung in der Anwendung | | |
| Anzahl der Möglichkeiten der Testfallerstellung | | |
| Erleichterung der Testfallerstellung | | |
| Erleichterung der Testfallerstellung in der Anwendung | | |
| Möglichkeit der Anwendung der Testfallerstellung | | |
| Erleichterung der Testfallerstellung | | |

Bilddownload:

<http://www.pr-x.de/fileadmin/download/documents/savignano/Tabelle-1.jpg>

Gewichtung

Im nächsten Schritt wurden die einzelnen Anforderungen von Projektmitarbeitern nach ihrer Bedeutung in drei Stufen priorisiert, wobei sehr wichtige Anforderungen mit 3, wichtige mit 2 und weniger wichtige mit 1 bewertet wurden.

Um hieraus wurde die Gewichtung der jeweiligen Anforderungen abzuleiten, wurde die Gesamtsumme der vergebenen Prioritätspunkte aller Bewerter mit 100% gleichgesetzt und die jede einzelnen Anforderung erhielt den entsprechenden Prozentsatz ihrer Prioritätspunkte. Siehe Tab. 2

Das Ergebnis dieser Untersuchung ist ein Anforderungskatalog, in dem die Prioritäten als Prozentzahl zu jedem Kriterium angegeben sind. (s. Tab. 3 rechter Teil)



Pressekontakt:

savignano software solutions
 Metin Savignano
 Königsallee 43
 71638 Ludwigsburg
 Tel. +49 (0) 7141 / 13345 11
 ms@savignano.net
 www.savignano.net

PRX
 Agentur für Public Relations GmbH
 Ralf M. Haassengier
 Kalkhofstraße 5
 70567 Stuttgart
 Telefon: 0711/71899-03/04
 Telefax: 0711/71899-05
 ralf.haassengier@pr-x.de
 www.pr-x.de

Tab. 3:

| Nr | Kriterium | Tool A | Tool B | Gewichtung | Tool A | Tool B |
|---|---|--------|--------|------------|--------|--------|
| Integrierbarkeit in die Host-Umgebung | | | | | | |
| IN.1 | geringer einmaliger Aufwand (Installation) | 3 | 3 | 3,84% | 3,84% | 3,84% |
| IN.2 | geringer ständiger Aufwand (bei der Ausführung) | 0 | 3 | 4,06% | 0,00% | 4,06% |
| Hilfeleistung bei der Testfallerstellung | | | | | | |
| TE.1 | Unterstützung durch Vergleichsmethoden (Assertionen) | 0 | 3 | 3,84% | 0,00% | 3,84% |
| TE.2 | Erfassung von Testdaten in Excel | 0 | 3 | 4,06% | 0,00% | 4,06% |
| TE.3 | Nachvollziehbarkeit der Testskripte/-daten für zweite Person | 1 | 3 | 3,84% | 1,28% | 3,84% |
| TE.4 | keine Notwendigkeit von Kenntnissen einer Programmier- oder Skriptsprache | 0 | 3 | 2,93% | 0,00% | 2,93% |
| TE.5 | Unterstützung bzgl. der Schnittstellen eines Moduls | 0 | 3 | 3,61% | 0,00% | 3,61% |
| TE.6 | Hilfeleistung bei Initialisierung | 3 | 3 | 4,06% | 4,06% | 4,06% |

Bilddownload:

<http://www.pr-x.de/fileadmin/download/documents/savignano/Tabelle-3.jpg>

Anforderungen, die bei der Priorisierung von mindestens drei Personen als K.o.-Kriterien festgelegt wurden, führen bei der Analyse der Werkzeuge bei einer negativen Bewertung zum Ausschluss. (s. Tab. 4)

Tab. 4:

| K.o.-Kriterien | |
|-----------------------|--|
| VE.1 | zentrale Verfügbarkeit der Testfälle |
| ÜB.1 | auf einen Blick ersichtlicher Teststatus (Überblick) |
| AU.1 | automatische Ausführung aller/gruppierter/einzelner Testfälle |
| WE.3 | Patches und Upgrades durch Anbieter (Wartung) |

Bilddownload:

<http://www.pr-x.de/fileadmin/download/documents/savignano/Tabelle-4.png>

Der Bereich Kosten wurde bei der Anforderungsdefinition bewusst vom Autor außer Acht gelassen, da der Fokus der Analyse auf den Funktionalitäten und Eigenschaften der Werkzeuge liegen soll. Auf dieser Basis kann eine generelle Entscheidung getroffen werden, ob ein externes Werkzeug zur Unterstützung von Unit-Tests geeignet ist. Außerdem sind die Kostenstrukturen der externen Werkzeuge zu unterschiedlich, um eine direkte Vergleichbarkeit zu erzielen. Hierfür wäre eine zusätzliche komplexe Analyse der Kostenstrukturen der Werkzeuge notwendig.

Kriterienkatalog

Der Anforderungskatalog wurde in sechs Bereiche unterteilt. Im Folgenden wird kurz erläutert, wie die einzelnen Kriterien zu verstehen sind. Um ein Kriterium genau zu benennen, wurde vom Autor jeweils eine Kurzbezeichnung des Kategorie und einer laufenden Nummer festgelegt.



Pressekontakt:

savignano software solutions
 Metin Savignano
 Königsallee 43
 71638 Ludwigsburg
 Tel. +49 (0) 7141 / 13345 11
 ms@savignano.net
 www.savignano.net

PRX
 Agentur für Public Relations GmbH
 Ralf M. Haassengier
 Kalkhofstraße 5
 70567 Stuttgart
 Telefon: 0711/71899-03/04
 Telefax: 0711/71899-05
 ralf.haassengier@pr-x.de
 www.pr-x.de

Integrierbarkeit in die Großrechner-Umgebung (IN)

Der erste Abschnitt des Anforderungskatalogs bezieht sich auf die Integrierbarkeit in die Host-Umgebung (Abk. IN). Dieser Punkt war wichtig, da die Cobol-Module als Testobjekte in einem Mainframe-Konfigurationsmanagement verwaltet wurden. Ein zukünftiges Werkzeug muss also die Testausführung auf dem Mainframe unterstützen. Als Bewertungskriterien sollen hier der einmalige Aufwand IN.1 und der ständige Aufwand IN.2 betrachtet werden. Der einmalige Aufwand umfasst alle Aktivitäten, die erfolgen müssen, damit das Werkzeug mit dem Mainframe interagieren kann (Installation und Konfiguration). Der ständige Aufwand gibt an, welche Arbeit bei der Testerstellung und der Testausführung an sich anfällt. Hierzu gehört beispielsweise die Frage, ob die Testausführung per Knopfdruck erfolgen kann oder, ob zuvor noch manuelle Arbeiten vorgenommen werden müssen, wie die Übertragung von Dateien auf den Großrechner.

Hilfeleistung bei der Testfallerstellung (TE)

Der zweite Bereich des Anforderungskatalogs enthält Kriterien, mit denen die Unterstützung des Werkzeugs bei der Erstellung von Testfällen bewertet werden soll. Diese Kriterien waren aufgrund der identifizierten Probleme bei der Testdatenerfassung erforderlich.

Mit dem ersten Kriterium TE.1 sollte getestet werden, wie gut der Abgleich von Soll- und Ist-Werten unterstützt wird. Mit TE.2 wurde geprüft, ob eine Erfassung von Testdaten auch in Excel möglich ist. Dadurch würde die Testfallerstellung einfacher und übersichtlicher werden. Außerdem würde dies auch die Erfassung und Kontrolle von Testdaten durch die Fachspezialisten ermöglichen. Als weiteren wichtigen Aspekt sollte die Nachvollziehbarkeit der Testszenarios TE.3 bestimmt werden, damit auch unbeteiligte Personen die Tests verstehen können. Dieser Aspekt ist außerdem mit Blick auf die Revision und die Prüfung durch die Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) sehr wichtig, da hier auch Tests eine wichtige Rolle spielen. Des Weiteren wurde mit TE.4 untersucht, ob der Tester mithilfe des Werkzeugs ohne Programmierkenntnisse auskommt. Das wäre wünschenswert, damit auch die Fachspezialisten die Unit-Tests erstellen oder verändern können. In TE.5 soll bewertet werden, ob das Werkzeug dem Entwickler die Eingabe der Testdaten dadurch erleichtert, dass es die Schnittstellenstruktur bei der Testdatenerfassung vorgibt. Durch das letzte Kriterium TE.6 aus dem Bereich der Unterstützung der Testfallerstellung sollte geprüft werden, ob eine Wiederherstellungsfunktion angeboten wird. Diese hat die Aufgabe einen Ausgangszustand zu definieren, der vor jedem Testfall wiederherzustellen ist.



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haabengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de

Verwaltbarkeit der Testfälle (VE)

Der dritte Abschnitt des Anforderungskatalogs enthält Anforderungen, die eine Unterstützung bei der Verwaltung der Testfälle abprüfen. Mit dem Kriterium VE.1 sollte sichergestellt werden, dass die Testfälle zentral verfügbar sind. Lokal abgelegte Tests sind wenig sinnvoll, wenn der jeweilige Entwickler gerade im Urlaub ist und ein Kollege aufgrund von Veränderungen die Tests ausführen muss. In VE.2 sollte geprüft werden, ob Testfälle zu Testsuiten zusammengefasst werden können. Diese Gruppierung ist bei mehreren Testfällen zu einem Themengebiet zweckmäßig, damit die Tests auch zusammen ausgeführt werden können. Punkt VE.3 betrachtet, wie gut die Veränderbarkeit bzw. Erweiterbarkeit der Tests ist. Das letzte Kriterium VE.4 bewertet, ob das Werkzeug erkennt, welche Testfälle bei einer Programmänderung betroffen sind.

Übersicht der Testergebnisse (ÜB)

Der vierte Bereich des Katalogs enthält zwei Kriterien zur Bewertung der Darstellung der Testergebnisse. Zuerst sollte in ÜB.1 geprüft werden, ob das Werkzeug einen Überblick bietet, bei dem auf einen Blick ersichtlich ist, welche Testfälle erfolgreich und welche fehlerhaft gelaufen sind. Der zweite Bewertungspunkt ÜB.2 bezieht sich auf die Detaillierung der Darstellung eines Fehlerfalls. Hierbei ist wichtig, dass der Tester die Informationen darüber erhält, in welchem Programm, welche Sollwerte nicht erreicht wurden.

Ausführung der Tests (AU)

Der fünfte Abschnitt des Anforderungskatalogs beinhaltet Kriterien zur Bewertung der Ausführung der Tests.

Mit AU.1 sollte geprüft werden, ob mithilfe des Werkzeugs die Tests automatisiert ausgeführt werden können. Dabei ist wichtig, dass mit wenig Aufwand einzelne, alle oder nur bestimmte, in Testsuiten zusammengefasste Testfälle ausgeführt werden können. Automatisiert heißt hier auch, dass die Soll- und Ist-Werte innerhalb des Tests miteinander abgeglichen werden, damit der Tester dadurch keine manuelle Prüfung vornehmen muss. Durch die Anforderung AU.2 soll kontrolliert werden, ob die Testfälle mit einem Debugger durchlaufen werden können. Dabei ist es irrelevant, ob das Werkzeug einen eigenen Debugger anbietet oder der Debugger der Entwicklungsumgebung genutzt werden kann. Dieser Aspekt ist wichtig, weil Fehler bei einer schrittweisen Ausführung des Programms leichter lokalisiert werden können. Mit AU.3 wird bewertet, wie gut das Werkzeug die Ergebnisse der Testausführungen dokumentiert. Eine Historisierung ist hierbei ebenfalls wünschenswert. Dadurch kann leichter nachvollzogen werden, aufgrund welcher Änderung am



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haasengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de

Programm ein Fehler im Test aufgetreten ist. Außerdem wird von der Revision und der BaFin ein Nachweis der Testaktivitäten in der Softwareentwicklung gefordert. Dies kann durch eine vollständige Dokumentation erfüllt werden.

Werkzeugeigenschaften (WE)

Der letzte Bereich des Anforderungskatalogs enthält sonstige Kriterien zum Werkzeug, Anbieter und Lieferumfang.

Mit WE.1 wird geprüft, ob das verwendete Werkzeug an interne Anforderungen angepasst werden kann. Außerdem sollte WE.2 den Aufwand bewerten, der bei der Einarbeitung in die Funktionsweise des Werkzeugs entsteht. Mit WE.3 wurde geprüft, ob das Werkzeug zukünftig weiter gepflegt und Fehler vom Anbieter behoben werden. Bei WE.4 wurde betrachtet, wie häufig innerhalb der letzten zwei Jahre neue Versionen des Werkzeugs veröffentlicht worden waren. Hier ist zu beachten, dass sowohl zu wenig als auch zu viele Releases negativ bewertet wurden. Mithilfe der Kriterien WE.5 und WE.6 wurde überprüft, ob die Werkzeuge mit Anwender- und Installationshandbuch bzw. einer entsprechenden Onlinedokumentation ausgeliefert werden. Diese Aspekte sind für den Installations- und den Einarbeitungsaufwand von großer Bedeutung. Die Anforderung WE.7 prüft, ob das Werkzeug an das jeweilige Testmanagement-Werkzeug angebunden werden kann. Mit dem letzten Punkt WE.8 wird das Werkzeug auf eine allgemeine Schnittstelle geprüft. Dies war wichtig, damit in der Zukunft eine Integration von anderen Werkzeugen möglich ist.

Bewertung

Die Bewertung der Werkzeuge wurde nach der Methodik der Nutzwertanalyse durchgeführt.

Zur Evaluierung der Anforderungen wurde eine Skala von null bis drei Punkten verwendet. Ein Kriterium, das nicht erfüllt wird, ist mit null Punkten zu bewerten. Je nach Güte der Erfüllung eines Kriteriums werden aufsteigend ein bis drei Punkte vergeben. Damit eine genaue Einordnung erfolgen kann, wurden vom Autor in Absprache mit erfahrenen Entwicklern zu den jeweiligen Kriterien mögliche Einstufungen definiert, indem spezifiziert wurde, wie eine vollständige Erfüllung der Anforderung aussehen müsste. Wo erforderlich und sinnvoll, wurde zusätzlich spezifiziert, wie eine unvollständige Erfüllung oder ein Fehlen der Anforderung zu erkennen ist. Diese werden in einem Anforderungskatalog zusammengefasst. (s. Tab. 5)

Tab. 5:



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haasengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de

| Nr | Kriterium | Einstufung | | | |
|--------------------------------------|---|---|---|---|--|
| | | 3 Punkte | 2 Punkte | 1 Punkt | 0 Punkte |
| Integrierbarkeit in die Host- | | | | | |
| IN.1 | geringer einmaliger Aufwand (Installation) | Installation ist innerhalb von einem Tag erledigt | Installation/Integration ist innerhalb von 3 Tagen erledigt | Installation/Integration ist innerhalb einer Woche erledigt | Installation/Integration dauert länger als eine Woche |
| IN.2 | geringer ständiger Aufwand (bei der Ausführung) | Testausführung erfolgt per Knopfdruck, kein weiterer Aufwand durch Anpassungen notwendig | - | - | es entsteht bei der Testausführung zusätzlicher Aufwand, z.B. manuelle Übertragung von Dateien auf den Host oder ähnliches |
| Hilfeleistung bei der | | | | | |
| TE.1 | Unterstützung durch Vergleichsmethoden (Assertionen) | es werden sinnvolle Vergleichsmethoden angeboten | - | - | es werden keine Vergleichsmethoden angeboten |
| TE.2 | Erfassung von Testdaten in Excel | Erfassung in Excel | - | ähnliche Erfassungsart wie Excel | keine Erfassung in Excel möglich |
| TE.3 | Nachvollziehbarkeit der Testskripte/-daten für zweite Person | die Testskripte/-daten sind für Fachspezialisten und Entwickler verständlich/nachvollziehbar | - | die Testskripte/-daten sind für einen zweiten Entwickler verständlich/nachvollziehbar | die Testskripte/-daten sind nicht für eine zweite Person nachvollziehbar |
| TE.4 | keine Notwendigkeit von Kenntnissen einer Programmier- oder Skriptsprache | es muss keine Programmiersprache beherrscht werden | - | es sind nur oberflächliche Kenntnisse der Programmiersprache erforderlich und das Tool liefert Unterstützung durch sinnvolle Beschreibungen | es sind umfassende Programmierkenntnisse erforderlich |
| TE.5 | Unterstützung bzgl. der Schnittstellen eines Moduls | das Tool gibt aufgrund der Schnittstellenstrukturen eine Struktur für die Testdaten vor | - | - | das Tool gibt keine Struktur für die Testdaten vor |
| TE.6 | Hilfeleistung bei Initialisierung | es kann ein Ausgangszustand definiert werden, der nach jedem Durchlauf wiederhergestellt wird | - | - | es existiert keine Unterstützung bei der Initialisierung bzw. der Definition eines Ausgangszustandes |

Bilddownload:

<http://www.pr-x.de/fileadmin/download/documents/savignano/Tabelle-5.jpg>

Untersuchung der Tools

Die Analyse der in Frage kommenden Werkzeuge wurde in einer kleinen Expertenrunde durchgeführt, um ein objektives Ergebnis zu erlangen. Dabei wurde bei jedem Tool an Hand der erarbeiteten Bewertungskriterien geprüft, ob und inwieweit dies erhobenen Anforderungen erfüllt waren. Die Punkte wurden in einer Tabelle dokumentiert. (s. Tab 3) Aus der jeweiligen Punktzahl und der Gewichtung der Anforderung wurde ein Prozentsatz errechnet, indem bei voller Punktzahl der volle Prozentsatz erreicht werden konnte. Bei geringerer Punktzahl wurde der Prozentsatz entsprechend anteilig ermittelt.

Die Gesamtsumme der so ermittelten Prozentsätze drückte schließlich in einer Zahl aus, wie gut das jeweilige Werkzeug zur Erfüllung der Anforderungen geeignet wäre.



Pressekontakt:

savignano software solutions
 Metin Savignano
 Königsallee 43
 71638 Ludwigsburg
 Tel. +49 (0) 7141 / 13345 11
 ms@savignano.net
 www.savignano.net

PRX
 Agentur für Public Relations GmbH
 Ralf M. Haasengier
 Kalkhofstraße 5
 70567 Stuttgart
 Telefon: 0711/71899-03/04
 Telefax: 0711/71899-05
 ralf.haassengier@pr-x.de
 www.pr-x.de

| Nr | Kriterium | Tool A | Tool B | Gewichtung | Tool A | Tool B |
|---|---|--------|--------|------------|--------|--------|
| Integrierbarkeit in die Host-Umgebung | | | | | | |
| IN.1 | geringer einmaliger Aufwand (Installation) | 3 | 3 | 3,84% | 3,84% | 3,84% |
| IN.2 | geringer ständiger Aufwand (bei der Ausführung) | 0 | 3 | 4,06% | 0,00% | 4,06% |
| Hilfeleistung bei der Testfallerstellung | | | | | | |
| TE.1 | Unterstützung durch Vergleichsmethoden (Assertionen) | 0 | 3 | 3,84% | 0,00% | 3,84% |
| TE.2 | Erfassung von Testdaten in Excel | 0 | 3 | 4,06% | 0,00% | 4,06% |
| TE.3 | Nachvollziehbarkeit der Testskript/-daten für zweite Person | 1 | 3 | 3,84% | 1,28% | 3,84% |
| TE.4 | keine Notwendigkeit von Kenntnissen einer Programmier- oder Skriptsprache | 0 | 3 | 2,93% | 0,00% | 2,93% |
| TE.5 | Unterstützung bzgl. der Schnittstellen eines Moduls | 0 | 3 | 3,61% | 0,00% | 3,61% |
| TE.6 | Hilfeleistung bei Initialisierung | 3 | 3 | 4,06% | 4,06% | 4,06% |

Bilddownload:

<http://www.pr-x.de/fileadmin/download/documents/savignano/Tabelle-3.jpg>

Das untersuchte generische Tool überzeugte besonders durch eine einfache und intuitive Bedienbarkeit sowie einen Erfüllungsgrad unserer Anforderungen von 96%. Dessen Einsatz kann somit den Prozess zur Erstellung von Unit-Tests deutlich erleichtern.

Anders als bei einer Framework-basierten Lösung, die vor einer Einführung hierfür umfangreichere Anpassungen erfordern würden, konnten die Anforderungen von dem Tool „out-of-the-box“ erfüllt werden. Auch die einfachere Anwendbarkeit ohne Programmierkenntnisse würde den Prozess zur Erstellung von Unit-Tests deutlich erleichtern.

((Infokasten))

Unit-Tests auf dem Mainframe

In der iterativen Vorgehensweise ist es unabdingbar, die entstehende Software am Ende jeder Iterationsstufe zu testen. Das stellt nicht nur die Qualität der implementierten Funktionalität sicher, sondern erlaubt auch – über die bloße Kenntnis des verbrauchten Aufwands hinaus – konkrete Aussagen zum inhaltlichen Projektfortschritt.

Unit-Tests sind dafür hervorragend geeignet, denn sie lassen sich schon früh einsetzen und sind nicht auf voll integrierte Anwendungen angewiesen. Unit-Tests verhindern auf diese Weise auch „technische Schulden“: Je früher die Softwaretests beginnen, desto höher ist die Kosten- und Zeitersparnis. Am Ende eines Entwicklungsprojekts steht dann mehr Qualität beziehungsweise eine größere Funktionalität der Mainframe-Software fürs gleiche Geld.

In der Client/Server-Welt ist dies selbstverständlich, wie aber stellt sich die Situation in der Welt der Mainframes dar? Meist werden hier so genannte Testtreiber verwendet, die speziell für jedes zu testende Modul implementiert werden müssen. Das kostet Zeit und Skill – und diese sind praktisch immer Mangelware. Das führt dazu, dass solche Tests oft nicht konsequent durchgeführt werden, die Treiberprogramme mangels Zeit veralten oder ganz vernachlässigt werden.



Pressekontakt:

savignano software solutions
 Metin Savignano
 Königsallee 43
 71638 Ludwigsburg
 Tel. +49 (0) 7141 / 13345 11
 ms@savignano.net
 www.savignano.net

PRX
 Agentur für Public Relations GmbH
 Ralf M. Haasengier
 Kalkhofstraße 5
 70567 Stuttgart
 Telefon: 0711/71899-03/04
 Telefax: 0711/71899-05
 ralf.haassengier@pr-x.de
 www.pr-x.de

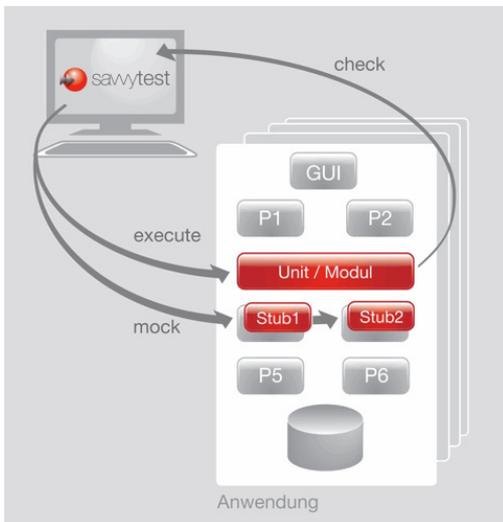


Bild: Isolierte Unit-Tests durch Stubs oder Mockups

BU: Stubs – auch Mockups genannt – ermöglichen es, Tests unabhängig von Umgebungsbedingungen durchzuführen, die oft schwer herzustellen sind. Dabei werden Komponenten außerhalb des Testobjekts simuliert, anstatt sie tatsächlich aufzurufen. Der Testaufwand kann so oft erheblich verringert werden.

Bilddownload:

http://www.pr-x.de/fileadmin/download/pictures/savignano/isolierte_unit_tests.JPG

((Infokasten, Ende))



Bild: Metin Savignano

BU: Metin Savignano ist Geschäftsführer von savignano software solutions in Ludwigsburg.

Bilddownload:

http://www.pr-x.de/uploadfiles/pictures//2904_163911_metin-savignano.jpg



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haaßengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de



Bild: Marcel Langer

BU: Marcel Langer ist bei der Finanz Informatik als Anwendungsentwickler im Host-Umfeld beschäftigt und hat dort auch sein duales Studium absolviert.

Bilddownload: http://www.pr-x.de/fileadmin/download/pictures/savignano/1marcel-Langer_FI.jpg

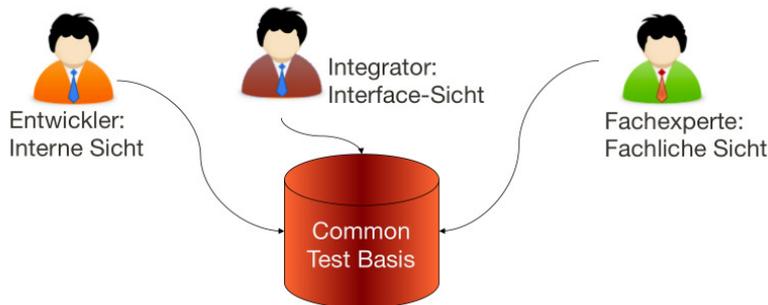


Bild: Breitere Testbasis

BU: Fachliche Sicht von Anfang an einbinden: Die frühe fachliche Prüfung einer Software spart Kosten und verbessert deren Qualität.

Bilddownload: http://www.pr-x.de/fileadmin/download/pictures/savignano/Bessere_Testbasis.jpg



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haassengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de

Eingabedaten referenzieren auf
Ausgabedaten vorhergehender Testfälle

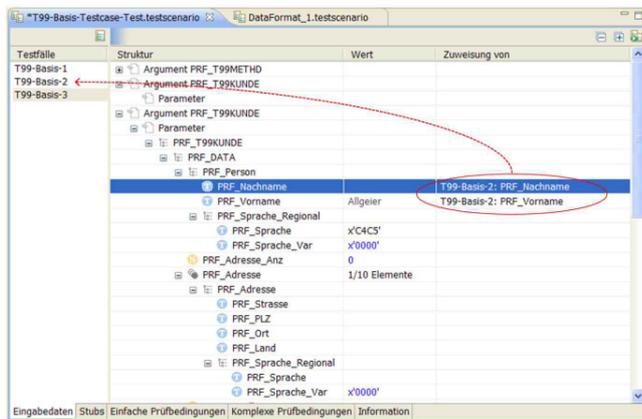


Bild: Grafik Testszenario

BU: Komplexe Testszenarien: Oft ist es erforderlich, Zusammenhänge zwischen mehreren Testschritten in einem Testszenario abzubilden. Eingabedaten referenzieren dabei auf Ausgabedaten vorhergehender Testfälle. Eine Testsoftware sollte dies schnell und einfach über Referenzen ermöglichen, ohne dass hierzu Programmcode geschrieben werden muss.

Bilddownload:

<http://www.pr-x.de/fileadmin/download/pictures/savignano/Testcases2.jpg>



Pressekontakt:

savignano software solutions
Metin Savignano
Königsallee 43
71638 Ludwigsburg
Tel. +49 (0) 7141 / 13345 11
ms@savignano.net
www.savignano.net

PRX
Agentur für Public Relations GmbH
Ralf M. Haaßengier
Kalkhofstraße 5
70567 Stuttgart
Telefon: 0711/71899-03/04
Telefax: 0711/71899-05
ralf.haassengier@pr-x.de
www.pr-x.de