

Testing-Suite

SL PROVISOR

Optimierung der Software-Qualitätssicherung

- √ Signifikante Kosteneinsparung
- √ Signifikante Verbesserung der Effektivität/Effizienz
- √ Fachexperten und Nicht-Entwickler erstellen Testfälle
- √ Testfälle datengetrieben und modular strukturiert
- √ Signifikante Erhöhung der Automation

September 2013

SoftwareLoft IT-Solutions GmbH | Dammtorstrasse 14 | D-20354 Hamburg

Tel.: +49.40.32 08 77 -80
products@softwareloft.de
www.softwareloft.de

Herausforderung „Straight-Through-Processing“

Die Entwicklung geschäftskritischer Anwendungen, Kernbanksysteme, Online-Applikationen oder innovativer Infrastrukturlösungen, ist bereits für sich genommen eine Herausforderung. Werden diese später in einem heterogenen Umfeld, z.B. in einer SOA-Landschaft (Service Orientierte Architektur), betrieben, dann ist jede Transaktion geprägt vom Zusammenspiel sehr vieler Einzelkomponenten und -diensten in einer verteilten Produktionslandschaft.

Vor allem bei Anwendungen, welche durch Straight-Through-Processing (STP) geprägt sind, geschieht die Interaktion mit den Anwendungskomponenten weniger über GUIs, als über Dateitransfers, Messaging-Queues, Datenbanken, Shellskripte und ähnlichen Mechanismen. Dies stellt die Entwicklung neuer Anwendungen, oder die Integration neuer Bausteine in eine bestehende Landschaft vor ganz besondere Herausforderungen.

Eine ganz besondere Rolle bekommt die Qualitätssicherung solcher Projekte, die sich auf die (technischen) Besonderheiten verteilter Systeme mit STP-Charakter einstellen muss. Mittels spezieller Modultests können zwar Entwickler mit ihrer technischen Expertise einfache Testszenarien abdecken, aber wegen der Vielzahl von Schnittstellen und Technologien wird die Fehlersuche bei komplexeren Anwendungstransaktionen nicht ohne spezielle Maßnahmen zu bewältigen sein. An solchen Projekten sind jedoch Mitarbeiter unterschiedlichster Organisationseinheiten beteiligt, die über keine (oder nur minimale) Entwickler-Skills verfügen.

Anwendungen durchleben in ihrem Lebenszyklus unterschiedliche Versionen, z.B. Version 1 in der Produktion, Version 1.1 in der Abnahme und Version 2 in der Entwicklung. Wie steht es dann mit der Möglichkeit, eine bestimmte Anwendungsversion jederzeit einem notwendigen funktionalen Komplett-Test zu unterziehen?

Das vorliegende Dokument richtet sich an Programm-/Produkt-Manager.

Für technische Details, tieferen Einstieg in die Grundkonzepte und für einen Überblick der Bedienoberfläche der Testing Suite „SL PROVVISOR“ steht ein separates Dokument „SL PROVVISOR Schnelleinstieg“ zur Verfügung (ca. 100 Seiten).

Fachexperten entwickeln Testfälle – mit „SL PROVISOR“

Mit der Testing-Suite „SL PROVISOR“ können jetzt auch Fachexperten ohne Programmierkenntnisse komplexe Testfälle modular erstellen, pflegen, verwalten und ausführen.

Dies liegt vor allem an folgenden Eigenschaften und Funktionen:

Datengetriebene Testfälle

Mit der Testing-Suite werden primär *datengetriebene* Testfälle erstellt.

Excel-Anbindung

Testfälle können nicht nur mit der Eclipse-basierten eigenen Oberfläche erstellt werden, sondern sind auch mit allen Excel-Funktionen bearbeitbar - einfach und bidirektional.

Copy & Paste

Für die einfache Vervielfältigung und Erstellung „ähnlicher“ Testfälle steht die Funktion Copy & Paste zur Verfügung. Es gibt keinen Grund mehr, sich mit wenigen exemplarischen Testreihen zu begnügen, obwohl eine größere Testabdeckung erforderlich wäre.

Test-Dokumentation

Die Dokumentation der Testfälle kann im Testobjekt selbst eingebettet werden.

Beschreibung der Applikation

Die Beschreibung der Applikation und der Schnittstellen kann im Testobjekt selbst eingebettet werden und dient weiterhin als Dokumentation.

Beschreibung der Installation

Installationsspezifische Details der Applikation, können ebenfalls im Testobjekt eingebettet werden.

Fertige Schnittstellen zu Datenbanken, Dateisystem, JMS, XML, WebSphere-MQ etc.

Um mit den unterschiedlichsten Anwendungskomponenten kommunizieren zu können, werden bereits viele Technologien unterstützt.

Connectoren und Konverter

Connectoren und parametrisierbare Standard-Konverter für eine Vielzahl gängiger Schnittstellen stehen bereits zur Verfügung.

Dabei definieren Connectoren mögliche Operationen auf eine Anwendungskomponente und Konverter übersetzen die Testdaten in das spezifische Format der Applikation. Mit diesen einfachen Instrumenten können fehlende und sehr individuelle Schnittstellen hinzugefügt werden.

Cloud-Anwendungen

Anwendungen in einer Cloud (z.B. Amazon-Cloud AWS) können separat oder im Zusammenspiel mit anderen Applikationskomponenten getestet werden.

Automatisierung in der SW-QS – Das perfekte Rüstzeug

Datengetriebene Testfälle und von Fachexperten erstellte Test-Suiten sind der Einstieg in eine erfolgreiche Qualitätssicherung für Softwareapplikationen. Mit „SL PROVISO“ erhält das Produktmanagement über den gesamten Lebenszyklus der Anwendung ein mächtiges Rüstzeug:

- Baselines und reproduzierbare Tests
- Lokale, oder zentral geführte Testfall-Datenbank
- Nightly- / Daily-Build mit der Möglichkeit, Tausende von Testfällen einzubinden
- Es können jederzeit gezielte Einzeltests, aber auch ganze Testbäume, oder sogar funktionale Komplett-Tests initiiert und ausgeführt werden
- Versionierung der Testfälle: „SL PROVISO“ kann zu jeder Anwendungsversion eigene Testfall-Suiten führen
- Identifikation ungewollter Seiteneffekte bei Änderungen in einem bestimmten Bereich der Anwendung
- Regressionstests mit Vorher-/Nachher-Vergleich
- Reporterstellung / Regressionsbericht

Und wann werden automatisierte Tests absolut unverzichtbar?

- ⇐ Mit steigender Zahl parallel zu bearbeitender Applikationen / Applikationsversionen!
- ⇐ Mit sinkenden Testzeiten, die dem Projekt für die Qualitätssicherung übrig bleibt und dadurch die Gefahr besteht, die Test-Disziplin zu vernachlässigen.
- ⇐ Wenn die erforderliche Hardware-Umgebung nur für begrenzte Zeit zur Verfügung steht, muss von der Installation bis Abschluss der Komplett-Tests der Grad der Automation maximiert werden.
- ⇐ Mit steigender Zahl der Auslieferungen.
- ⇐ Mit sinkenden Reaktions- und Auslieferungszeiten.

Nicht nur hohe Qualitätsansprüche machen dieses Erfordernis greifbar, sondern oft bereits striktere „Service-Level-Vereinbarungen“ mit dem Kunden.

Praxisbeispiele

Zwei Anwendungsszenarien aus der Vergangenheit für den bewussten Einsatz von „SL PROVISOR“ – mit großem Erfolg:

Ersetzen einer Anwendung – Regressionstests:

» Nur durch Einsatz von SL PROVISOR wurde das Projekt überhaupt erst genehmigt!

Eine Applikation muss durch eine Variante neuerer Generation ersetzt werden, wobei folgende Rahmenbedingungen gelten:

- Weil die betroffene Applikation für die gesamte Geschäftssteuerung innerhalb eines Kernbanksystems zuständig ist und demzufolge mit einer Vielzahl von eigenständigen Anwendungen interagiert, muss die äußere Sicht auf die Geschäftssteuerung erhalten bleiben, also
 - Schnittstellen der Applikation müssen stabil und unverändert bleiben
 - Verhalten der Applikation an den Schnittstellen muss ebenfalls unverändert bleiben und den gleichen Output erzeugen, die gleichen Ereignisse auslösen usw.
- Ausnahmen werden zuvor bewusst festgelegt
 - speziell erstellte Testfälle verifizieren die Ausnahmen
 - unvorhersehbare Seiteneffekte sind zu identifizieren und zu bewerten

Integrationstests in einem sehr komplexen Umfeld:

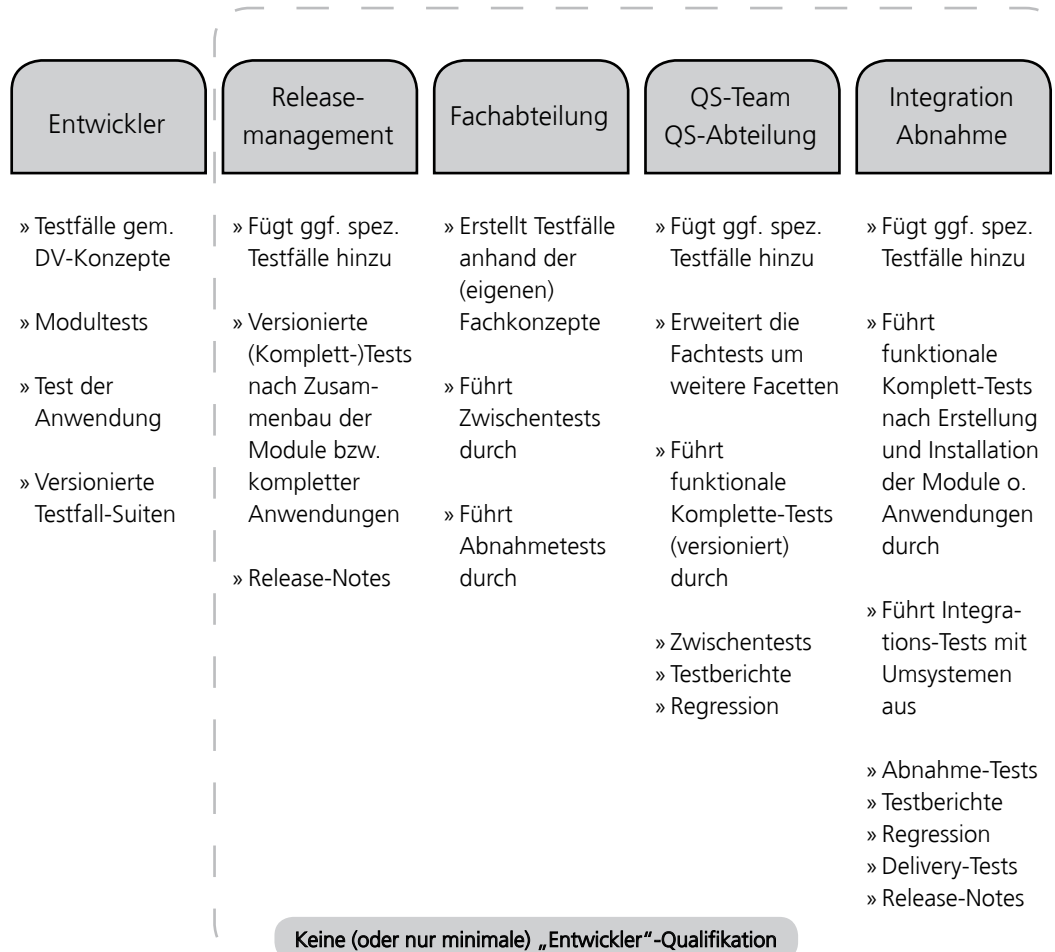
» Laufzeit von geplanten 18 Monaten auf 7 Monate verkürzt

» Personaleinsatz für Test/QS gegenüber Planung um 6 Mannjahre verringert

Ein neues komplexes Wertpapiersystem wird in ein bestehendes Kernbanksystem integriert:

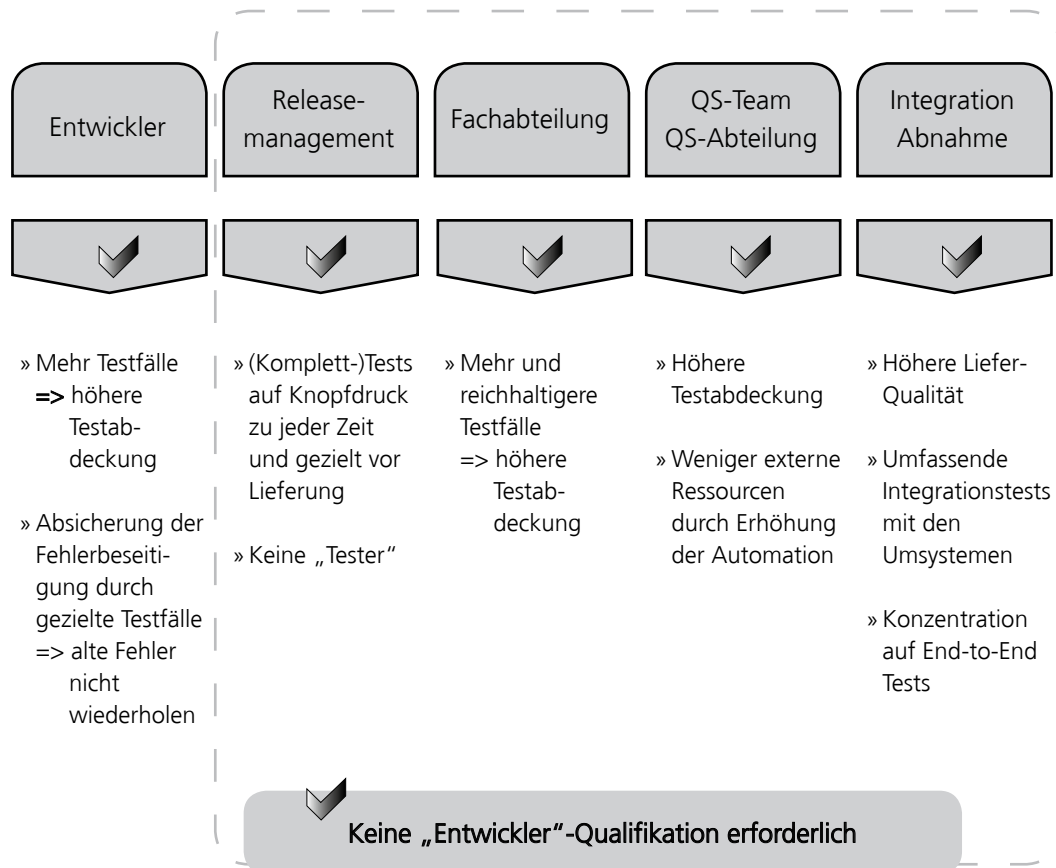
- Es sind ca. 35 Teilsysteme betroffen
- Die Kommunikation zwischen den Teilsystemen wird über viele unterschiedliche Technologien und ca. 95 Schnittstellen realisiert
- Wertpapier- und Kernbanksystem sind modular aufgebaut (SOA-Ansatz)
- Eingesetzt wird ein bunter Mix unterschiedlicher Schnittstellen-Technologien:
 - Dateien
 - Datenbanken (JDBC, ...), DB-Replikationen (incentage, ...)
 - MessageQueue (JMS, WebSphere MQ)
 - XML, EJB, Webservice, ...

„SL PROVISOR“ – Ein Werkzeug, viele Anwendergruppen



Von der Gruppe der Entwickler abgesehen, sollte in den anderen Bereichen **keine (oder nur minimale) Entwickler-Qualifikation** erwartet werden. Für die Durchführung ihrer originären Aufgaben, setzen die oben genannten Anwendergruppen „SL PROVISOR“ ein und erhalten umfassende Unterstützung durch rollenbasierte Funktionen.

„SL PROVISOR“ – Erhebliches Potenzial für Kostensenkung und/oder höhere Effizienz/Effektivität



Durch den Einsatz der Testing-Suite „SL PROVISOR“ wird sichergestellt, dass

- Anwendergruppen ihren QS-Anteil vollumfänglich beitragen können,
- Notwendigkeit zur temporären Personal-Aufstockung im Testumfeld entfällt, oder minimiert wird,
- Engpass an Betriebsmitteln und Ressourcen unterbunden wird, weil automatische Tests zu jeder beliebigen Zeit initiiert werden können (bessere Verteilung und Ausnutzung der Kapazitäten),
- dank der hilfreichen Mechanismen, die Produktivität aller Beteiligten erheblich gesteigert wird.

Fazit – Nutzen der Testing-Suite „SL PROVISOR“

Konsequent eingesetzt bietet die Testing-Suite „SL PROVISOR“ also in mehrfacher Hinsicht erhebliches Potenzial:

<p>Kostensenkung</p>	<p>Durch eine große Zahl von Testfällen (fachlich und technisch), welche automatisiert und zu jeder Zeit unbeaufsichtigt ausgeführt werden können, eröffnet sich die Chance für neue QS-Strategien und ganz nebenbei auch ein enormes Einsparungspotenzial.</p> <p>Nur noch Testfälle, die ausschließlich manuell ausgeführt werden müssen (z.B. wegen Abhängigkeit zu Hardware), müssen tatsächlich durch geschulte Anwender getestet werden.</p> <p>Für die gleiche Anzahl an auszuführenden Testfällen ergibt sich nach den vorliegenden Erfahrungen sowohl eine beträchtliche Zeitersparnis (bis zu 85%), als auch eine deutliche Entlastung des meist extern besetzten Test- und QS-Personals (bis zu 65%).</p>
<p>Erhöhung der Effektivität und Effizienz</p>	<p>Testfälle für die Fach- und Abnahmetests können bereits mit Vorliegen der Fach- und Designkonzepte erstellt werden.</p> <p>Mit „SL PROVISOR“ wird eine wesentlich bessere Testabdeckung erreicht, weil die zeitintensive Erstellung „ähnlicher“ und abgeleiteter Testfälle und deren Verwaltung extrem vereinfacht wird.</p> <p>Dies vermeidet „musterhaft“ durchgeführte Tests, die aus Zeit- und Personalmangel nur in geringer Anzahl und nur stark vereinfacht eingesetzt werden.</p>