

Die Einführung eines wieder verwendbaren Testframeworks in der Sozialversicherung

Armin Beer

Siemens IT Solution and Services PSE
Gudrunstrasse 11
A-1100 Wien
armin.beer@siemens.com

Abstract: Einen gut durchdachten und wirksamen Test für ein Softwareprojekt aufzusetzen ist anspruchsvoll und aufwendig. Daher wurde im Rahmen einer Beratung für einen Sozialversicherungsträger ein wieder verwendbares Testframework für einen iterativen Entwicklungsprozess eingeführt. Dessen wichtigste Merkmale sind eine enge Kooperation zwischen Analytikern, Entwicklern, Anwendern und Testern, der Aufbau von Test-Know-How im ganzen Team und Maßnahmen zur frühen Fehlerfindung durch automatisierte und systematische Tests auf Basis des V-Modells. Die Anwendung des Frameworks in einem Pilotprojekt ermöglichte die planmäßige Lieferung einer Applikation mit zufriedenstellender Qualität und war die Voraussetzung für dessen generelle Verwendung.

1 Einleitung

Bei einem Sozialversicherungsträger, der die gesetzliche Kranken- und Unfallversicherung für die österreichischen Beamten und Vertragsbediensteten durchführt werden sukzessive bestehenden Host-Applikationen durch moderne J2EE Applikationen abgelöst. Im Zuge dessen sollte ein für alle Software-Entwicklungsprojekte verbindliches Testframework, mit dem Ziel die Qualität der dort realisierten Produkte zu sichern, eingeführt werden. Dazu wurden Test-Fachleute von SIS PSE als Berater angefordert.

Folgende Anforderungen waren zu erfüllen:

- Eignung des Testframeworks für eine iterative Entwicklung
- Aufbau einer Testmanagementgruppe, die unabhängig von der Entwicklung Tests durchführt oder einzelne Projekte unterstützt
- Aufbau von Test-Know-how und *Skills* bei Entwicklern und Testern
- Anwendung von Tools – vorzugsweise *Open Source Software*

2 Realisierung des Testframeworks

Es wurde ein Framework, dessen Betonung auf einem klaren methodischen Vorgehen nach ISTQB [SL05] beruht, realisiert. Insbesondere wurden systematische und explorative Testfall-Entwurfsverfahren zur Reduktion des zu testenden Umfangs unter Beibehaltung des Anspruchs auf hohe Fehler-Aufdeckungs-Kapazität angewendet.

Das Testframework umfasst:

- Tool-gestütztes Testmanagement u. a. mit definierten Rollen, Abläufen und Metriken
- *Workflow* mit durchgängiger Verfolgbarkeit (Traceability) von den Anforderungen zu Use-Case-Beschreibungen und weiter zu Testfällen
- Richtlinien für methodisches Testfall-Design (zustandsbasiert, Äquivalenzklassen, exploratives Testen etc.)
- Curriculum für Fachtester, Testleiter und Entwickler (ISTQB-Certified Tester-Prüfung, Schulungen in Methoden und Tools, *Coaching* durch Berater etc.)
- Support verschiedener Projekte durch eine Testmanagement-Gruppe

Das dem Testvorgehen zugrundeliegende V-Modell wurde für die iterative Entwicklung adaptiert. Alle Rollen im Projekt-Team sind genau definiert. Zum Beispiel unterstützt der Testdaten-Administrator auch die Entwickler bei der Erstellung ihrer Testdaten. Zur besseren Team-Bildung und Förderung der Motivation für den Test werden Analytiker, Fachtester, Entwickler etc. in Workshops und durch begleitendes *Coaching* in der Anwendung der Methoden und Tools geschult. Der Know-How-Transfer erfolgte anhand von Test-Richtlinien, die das Rollen-spezifische Wissen berücksichtigt.

Automatisierte Regressions-Tests auf allen Testebenen und Last- und Performance-Tests werden regelmäßig durchgeführt. Um die Transparenz über den Status der einzelnen Arbeitspakete zu erzielen, können z.B. die Test-Ergebnisse jedes *Nightly Builds* von jedem Entwickler oder Fachtester über einen *Intranet-Link* abgefragt werden. Die Tätigkeiten des Testprozesses nach ISTQB (Planung und Steuerung, Analyse und Design, Realisierung und Durchführung, Auswertung und Bericht, Abschluss) werden durch Tools unterstützt. Für das Testmanagement werden das Tool *SiTEMPPPO*¹, für das Anforderungs- und Fehlermanagement das Tool *Bugzilla*², für die Testautomatisierung die Tools HP WinRunner und *QuickTestPro*³, für Lasttests die OpenSource Software *OpenSTA*⁴ und für die Komponenten-Tests die OpenSource Tools⁵ *JUnit*, *HttpUnit*, *CruiseControl* und *DbUnit* verwendet. Das Framework und die Dokumente sind so gestaltet, dass sie für andere Entwicklungsprojekte wieder verwendbar sind.

¹ Tool von SIS PSE (SiTEMPPPO: Test Execution, Managing, Planning and rePorting Organizer).

² bugzilla.mozilla.org

³ HP QuickTest Professional Software

⁴ <http://www.opensta.org/> .

⁵ <http://www.junit.org/>; <http://www.torsten-horn.de/techdocs/java-httpunit.htm>;

<http://cruisecontrol.sourceforge.net/>; <http://www.realsolve.co.uk/site/tech/dbunit-quickstart.php>

3 Anwendung und Evaluierung des Testframeworks im Projekt P1

Das Framework wurde zum ersten Mal bei einem mittleren Entwicklungsprojekt eingeführt und evaluiert. Das Projekt P1 beschäftigte sich mit der Entwicklung einer Applikation zur Verwaltung von Stammdaten von sozialversicherten Personen. Als Projektlaufzeit war ein Kalenderjahr vorgesehen. Das Projektteam umfasste ca. 30 Personen, zusammengesetzt aus Mitarbeitern des Sozialversicherungsträgers und Mitarbeitern von zwei großen Softwaredienstleisterfirmen.

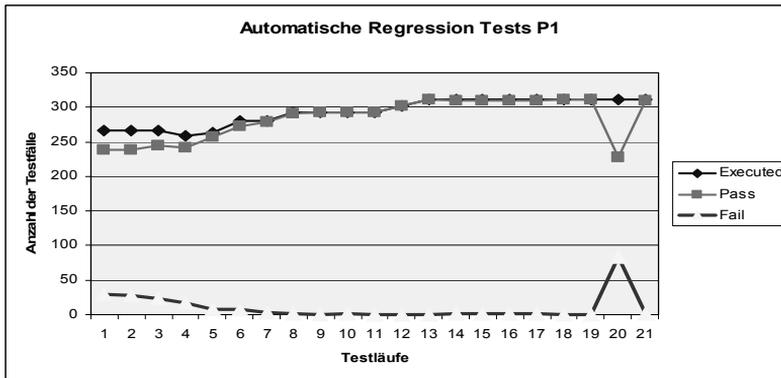


Abbildung 1: Ergebnisse der Regressionstests

Die Testmanagement-Gruppe wurde bereits am Beginn der in drei Iterationen geplanten Entwicklung involviert. Nach jeder Iteration wurde das Feedback der Anwender berücksichtigt. Das hatte zum Teil Rückwirkungen auf Funktionalität früherer Iterationen. Da ein Tool-gestütztes Anforderungs-Management vorhanden war und die Testfälle systematisch aus den Use-Case Beschreibungen abgeleitet wurden, konnte der Testfall-Pool aktuell gehalten werden. Insgesamt wurden 1200 Testfälle durch das Fach-Personal realisiert, wobei 10%-20% davon explorative Testfälle waren. 25% davon wurden mit dem Tool HP WinRunner automatisiert.

Zur Auswertung der Tests wurden Testfortschritts-Metriken und Fehlermetriken verwendet. Abb. 1 zeigt die Ergebnisse der Regressionstests. Ab dem 6.Lauf wurde die Qualität besser, beim 13.Lauf erfolgte die Abnahme, in der anschließenden Wartungsphase war die Qualität gut, bis auf dem Testlauf 20, vor dem ein *Refactoring* durchgeführt wurde. Die Fehlertrendanalyse in Abb. 2 zeigt die monatlichen Auswertungen aus Bugzilla. Zum Zeitpunkt M4 stieg die Anzahl der neuen Fehler stark an und die Gesamtzahl der offenen Fehler erreichte einen Höchstwert. Das Absinken der neu entdeckten und vor allem auch der insgesamt noch offenen Fehler zeigt dann die positive Entwicklung zum Produktivtermin hin. [KA03]

Zur Bewertung des Reifegrads des Testprozesses wurde das TPI-Modell [PK00] in P1 angewendet. Die Bewertung ergab, dass der Testprozess auf Basis des vorgestellten Frameworks insgesamt als „beherrschbar“ einzustufen ist. Einige Kernbereiche haben sogar bereits die Stufe „effizient“ erreicht.

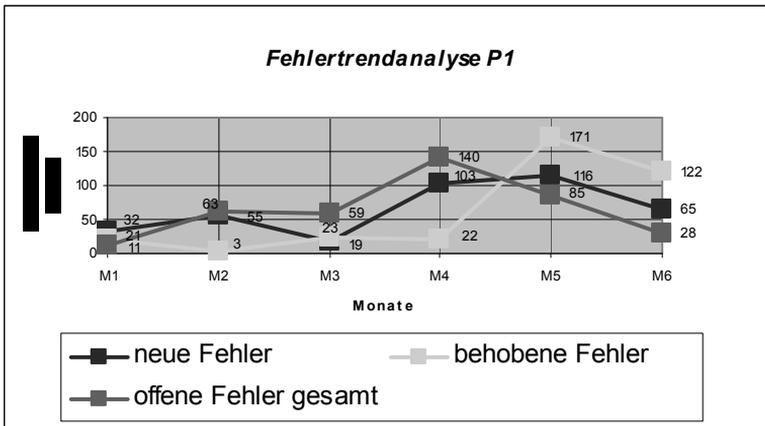


Abbildung 2: Fehlertrendanalyse P1

Eine große Herausforderung bestand darin, die Anwendbarkeit des Frameworks im Projekt P1 nachzuweisen, wobei dieses wie geplant erfolgreich im Jänner 2006 abgeschlossen wurde. Die anschließende Wartungsphase zeigte, dass keine gravierenden Fehler mehr auftraten und die geplanten Release-Zyklen eingehalten werden konnten. Inzwischen ist das Testvorgehen für alle Projekte verbindlich gemacht worden.

4 Zusammenfassung

Die wichtigsten Erfolgsfaktoren für den Einsatz des vorgestellten Testframeworks in der Sozialversicherung waren: (1) Aufbau von Test-Wissen und Erfahrung im Laufe einer iterativen Entwicklung durch spezifisches *Coaching* z.B. für Domänen-Experten und Entwickler, (2) Realisierung eines Testfall-Pools bestehend aus systematischen und explorativen Testfällen, (3) regelmäßige automatisierte Regressions-Tests für alle Teststufen und Last- und Stress-Tests. Die Wiederverwendung wird durch das einheitliche Testvorgehen nach ISTQB, den Know-How-Transfer innerhalb der Organisation, Metriken und *Templates* unterstützt. Als weitere Optimierungs-Maßnahme wird gerade die Einführung statischer Analyse-Werkzeuge vorbereitet.

Literaturverzeichnis

- [KA03] Kan, S.H.: Metrics and Models in Software Quality Engineering. Addison-Wesley, 2003.
- [PK00] Pol, M.; Koomen, T.; Spillner, A.: Management und Optimierung des Testprozesses. 1. Auflage, dpunkt.-Verlag, Heidelberg, 2000.
- [SL05] Spillner, A.; Linz, T.: Basiswissen Softwaretest – Aus- und Weiterbildung zum Certified Tester – Foundation Level nach ISTQB-Standard. 3. Auflage, dpunkt.-Verlag, 2005.