

Iterativ-inkrementelle Automatisierung von Integrationstests – ein Fallbeispiel

Beitrag für den 18. TAV-Workshop der GI-FG 2.1.7 „Test, Analyse und Verifikation von Software“
20./21.6.2002, Hasso-Plattner-Institut, Universität Potsdam

Uwe Hehn, method park Software AG (Erlangen)
Uwe.Hehn@methodpark.de

Zusammenfassung: Integrationstests können bekanntlich einen wichtigen Beitrag zur Softwarequalitätssicherung liefern: Einerseits sollen Probleme frühzeitig aufgezeigt werden, andererseits ist eine vollständige Durchführung aller Testfälle eine übliche Voraussetzung für die Freigabe der Software. Das heisst, dass für jede Softwarerelease Integrationstests mehrfach und dennoch mit geringem personellen und zeitlichen Aufwand durchführbar sein sollten; eine Anforderung, die in der Praxis nicht ohne weiteres erfüllt werden kann.

Im Vortrag wird ein Konzept für die iterativ-inkrementelle Automatisierung von Integrationstests vorgestellt, das in den Projektablauf eingebettet ist und zu deutlicher Aufwandsreduzierung führt, ohne Anforderungen der Qualitätssicherung zu vernachlässigen. Die praktische Umsetzbarkeit des Konzepts wird am Beispiel eines großen Softwareprojekts vorgestellt.

1. Nutzen von Integrationstests

Unbestritten ist die Planung, Spezifikation und Durchführung von Integrationstests ein integraler Bestandteil der Qualitätssicherung von Software. Nach „Feature freeze“ und „Final build“ ist eine vollständige Durchführung des Integrationstests erforderlich.

Einen noch höheren Nutzen kann man erwarten, wenn der Integrationstest bzw. Teile des Integrationstests bereits vor „Feature freeze“ durchgeführt werden, weil damit eine sehr frühe Rückmeldung etwaiger Schwachstellen und Fehler an die Entwicklung möglich ist.

2. Projektrealität

Häufig dauert ein Integrationstest-Zyklus jedoch so lange, dass aus Zeitgründen nur sehr wenige Durchläufe des Integrationstests möglich sind. Ursache für den hohen Aufwand ist oft, dass die Tests weitgehend manuell auszuführen sind.

Natürlich liegt es auf der Hand, die Tests zu automatisieren, um sie dann in einer deutlich kürzeren Zeit durchführen zu können (z.B. „über Nacht“ oder „übers Wochenende“). Leider ist die Automatisierung von Tests eine nicht einfache und vor allem aufwändige Arbeit. Typischerweise steht menschliche Arbeitskraft nur begrenzt zur Verfügung, und diese wird – solange die Tests nicht automatisiert sind – nahezu vollständig zur Durchführung der manuellen Tests benötigt. Daher ist meist nur wenig Zeit, um die Testautomatisierung voranzutreiben. Wie lässt sich dieses Zirkelproblem lösen?

3. Konzept

Wir beschreiben einen Migrationspfad, der auf realen Erfahrungen in einem großen Softwareentwicklungsprojekt basiert. Die grundlegende Idee besteht darin, die Testautomatisierung nach Festlegung eines Konzepts inkrementell durchzuführen, so dass bereits bei der ersten Durchführung der Tests Aufwandsreduzierungen zu erwarten sind. Die Zeit, die dadurch gespart wird, wird für das nächste Automatisierungsincrement genutzt, usw. Der Effekt durch die Automatisierung wächst daher schnell an.

Wir sind davon überzeugt, dass die Grundidee leicht auf andere Softwareprojekte übertragen werden kann.

3.1. Anforderungen

Die wesentlichen Punkte sind

- Definition einer einheitlichen Test-Schnittstelle, die sowohl manuelles (dialogorientiertes) Testen als auch automatisiertes (szenariobasiertes) Testen erlaubt,
- Definition eines einheitlichen Verfahrens zur Übergabe von Parametern und Ergebnissen, das auch den Test von verteilten Systemen erlaubt,
- Ermöglichen von automatischen Prüfungen sowohl auf Testschnittstellen- als auch auf Szenarioebene (Vergleich Ist/Erwartet)
- skriptbasiertes Testen.

3.2. Zur Vorgehensweise

Folgende Vorgehensweisen werden festgelegt:

- Reengineering und Restrukturierung von vorhandenem Integrationstest-Code, um mit begrenztem Aufwand zu automatisierten Integrationstests zu kommen
- Inkrementelle Verbesserung des Testcodes; die Priorität liegt auf möglichst frühen, nicht auf möglichst vollständigen Tests (Nutzen der Zeit, die durch Teilautomatisierung erspart wird, für weitere Automatisierung des Integrationstests).

4. Fallbeispiel

Das beschriebene Konzept wurde in einem großen Softwareprojekt anhand einer umfangreichen Komponente mit mehreren Hundert Schnittstellenfunktionen umgesetzt. Die zugehörige Testsoftware mit über 100000 Zeilen Code war nahezu vollständig auf manuellen dialogbasierten Test abgestellt; für eine komplette Durchführung eines Integrationstestzyklus wurden mehrere Personenmonate benötigt. Da für diese Komponente mit voraussichtlich noch mindestens 10-15 Releases gerechnet wurde, wurde der Aufwand für die Umsetzung der Testautomatisierung als sinnvoll erachtet.

4.1. Automatisierungsplanung

Um einen möglichst frühen Effekt der geplanten Maßnahmen zu erreichen, wurden sowohl Testfälle als auch die Schnittstellen-Funktionen klassifiziert. Nach dem Grundsatz: „Mit wenig Aufwand viel Effekt erreichen“ wurden zu-

nächst Testschnittstellen für die Schnittstellenfunktionen implementiert, die a) einfach umzusetzen waren und b) wo die Automatisierung einen deutlichen Zeitgewinn erwarten ließ.

Die Planung für die Umsetzung des Konzepts erfolgte durch Schätzungen: Aufgrund von Messungen des Zeitaufwandes für die Umstellung von manueller Testdurchführung auf automatische Testdurchführung – jeweils pro Schnittstellenfunktion – wurde der Gesamtaufwand der Umstellung geschätzt.

Analog wurde der Aufwand für die Durchführung von Integrationstests bestimmt und extrapoliert: Basis war die Messung des Aufwandes für die manuelle sowie die automatische oder halbautomatische Abarbeitung der Testfälle bzgl. einiger ausgewählter Schnittstellenfunktionen.

4.2. Skalierbarkeit

Der Aufwand für die Umstellung von manuellem Test auf automatischen Test war für einen großen Teil der Schnittstellenfunktionen wie erwartet. Ebenso ergab sich für einen großen Teil der Testfälle eine deutliche Aufwandsreduzierung durch die Automatisierung von Testszenarien.

Ein erheblicher Anteil der Umstellungen der Testschnittstelle erforderte jedoch einen deutlich höheren, von vorneherein schwer abzuschätzenden Aufwand, wodurch der Automatisierungsprozess weniger schnell voranschritt als erhofft.

Die ursprüngliche, vereinfachende Hypothese eines linearen Aufwands für die Umstellung aller Schnittstellen musste daher entsprechend korrigiert werden.

5. Ausblick

Bisher ist das beschriebene Testautomatisierungsprojekt parallel zu zwei Releasezyklen durchgeführt worden. Der Testaufwand für die zweite Durchführung der Integrationstestfälle war gegenüber der ersten Durchführung bereits um etwa 30% reduziert. Während der nächsten Releasezyklen ist zu klären, wieweit die Automatisierung getrieben werden soll, bzw. wann die zusätzliche Reduktion des Aufwands nicht mehr wirtschaftlich ist.