

SCRUM: Untergang der Test Center?

Klaus Haller

Inhaltliche Strukturierung		Temporale Ordnung				
Typ	Validiert... / Testaspekt	V-Modell			Scrum	
Unit Test	Module, Klassen etc.	1				
Unit Integration Tests	Zusammenspiel von Modulen etc.		2			
System Tests	Ganzes System			3		
System Integration Tests	System mit Umsystemen				4	
User Acceptance Tests	Anforderungen					5

Abbildung 1: Testziele und Strukturierung (verschieden tiefe Blauschattierungen symbolisieren, wie intensiv getestet wird.)

Tester haben das V-Modell mit seinen Teststufen verinnerlicht. Sie reagieren verunsichert, wenn neu agil Software entwickelt wird. Das V-Modell sammelt am Projektbeginn alle Anforderungen – vollständig und abschliessend. Die Software wird entwickelt, durchgetestet und danach dem Kunden übergeben. Das ist bei der agilen Softwareentwicklung anders. Das Agile Manifest [1] akzeptiert auch spätere oder sich ändernde Anforderungen nach Projektbeginn. Weiter fordert es regelmässig lauffähige Zwischenversionen. Agile Entwicklung stellt also bekannte Regeln in Frage und hat damit Erfolg. Jenseits anekdotischer Jubelberichte agiler Projekte und ihrer Projektleiter belegen Studien die Vorteile. So loben Vijayarathy und Turk: „The ability to meet client needs and the delivery of quality software products on time are significant benefits of agile development ...“ [2].

In der heutigen agilen Entwicklungspraxis dominiert Scrum. In einer Studie aus dem Jahr 2011 bekennen sich 57% zu Scrum, 27% verwenden hauseigene Methoden – andere Methoden erreichen kaum 5% Verbreitung [3]. Daher diskutiert dieser Artikel zwei Fragekomplexe für Scrum:

- Was wird getestet? Wann?
- Wie ist das Testen in der Organisation verankert?

Beim V-Modell folgt erst nach Analyse, Design und Umsetzung das Testen mit den Phasen Unit Test, Unit Integration Test, System Test, System Integration Test und User Acceptance Test. Mit den Phasen strukturiert das V-Modell das Testen inhaltlich (Was wird getestet?) und temporal (wann werden die verschiedenen Tests gemacht?).

Scrum strukturiert Projekte in Sprints. Ein Sprint hat eine feste Dauer, zum Beispiel zwei oder vier Wochen. Das Entwicklungsteam wählt am Anfang eines Sprints die User Stories (Aufgabenpakete) aus, die es in dem Sprint umsetzt. Am Ende des Sprints sind diese User Stories nicht nur implementiert, sondern vollständig getestet und auslieferbereit. Scrum kennt keine nachgelagerten Testphasen mehr. Das erfordert eine Neuinterpretation der Testphasen. Aus

Testphasen werden Testaspekte. Scrum braucht alle Testaspekte des V-Modells wie Unit Tests oder System Tests. Neu ist einzig, dass die feste Reihenfolge – erst Unit Test, dann Unit-Integration-Test etc. – entfällt. Das illustriert Abbildung 1. Die Farbintensität veranschaulicht den Bedarf an Testaspekten in der jeweiligen Phase (V-Modell) oder in Sprints (Scrum). Beim V-Modell gibt es die bekannte Ordnung, bei Scrum ist die Verteilung dynamischer. Ein Sprint beinhaltet meist überlappend verschiedene Testaspekte, wobei es auch einzelne reine Integrations- oder Testsprints geben kann.

Welche Testaspekte wichtig sind hängt vom Unternehmen ab. Große IT-Firmen wie Microsoft und Google entwickeln intern Software mit eigenen Entwicklern und Testern. Ihre Bücher [4][5] prägen das Testing. Für sie sind alle Testaspekte wichtig. Doch viele Tester in Deutschland, Österreich und der Schweiz arbeiten in einem anderen Umfeld. Kundenunternehmen und ihre IT-Abteilungen (Kunden-IT) lagern Entwicklungsprojekte oft aus oder kaufen Standardsoftware ein. Standardsoftware wird „nur“ installiert und konfiguriert. Softwarelieferanten und Kunden-IT haben folglich andere Bedürfnisse für Tests (Abbildung 2). Der Softwarelieferant übernimmt Unit Tests, Unit-Integration Tests und System Tests und steigert so die Produktqualität. Er managet je nach Projektgrösse zusätzlich Haftungs- und Reputationsrisiken, beispielsweise durch Reviews.

Die Kunden-IT hat beim Testen einen anderen Fokus. Sie muss verhindern, dass eine neue Software die Bank oder Produktion lahmlegt. Ihr Fokus sind System Integration Tests und User Acceptance Tests. System Tests werden nur bei stark konfigurierbaren Systemen wie ERP-Systemen notwendig. Ziel ist nicht, die Softwarequalität zu verbessern. Das ist die Aufgabe des Softwarelieferanten. Ziel ist eine Abnahme oder Rückweisung einer konkreten Softwareinstallation.

Wie wirkt sich die neue Welt des Testens auf die Testorganisation in Firmen aus? Softwarelieferant und Kunden-IT

haben, wie oben beschrieben, andere Bedürfnisse beim Testen. Es bieten sich unterschiedliche Organisationsformen an. Bei einer Kunden-IT gibt es keinen Änderungsbedarf. Sie muss entscheiden, ob die Software produktiv eingesetzt werden kann. Ein unabhängiges Test Center hilft.

Ein Softwarelieferant hat dagegen kaum Freude, wenn ein Test Center erst am Projektende kurz vor Auslieferung sagt „Zu viele Fehler - „zurück zum Start!“. Bei Softwarelieferanten geht es um „move quality upstream“ [4]. Dafür werden die Tester in Scrum-Teams integriert und eingebettet. Organisatorisch können die Tester aus einem zentralen Tester-Pool stammen oder fixen Produkt- oder Entwicklungsteams zugeordnet werden. Solche Tester sind natürlich weniger unabhängig als bisher. Ein kleines Quality Assurance-Team sollte daher ergänzend Prozesse und Testqualität überwachen. So verhindern Softwarelieferanten, dass Projektteams unter grossem Druck Software ausliefern, die die Reputation ruiniert oder Schadenersatzansprüche verursacht.

In grossen Non-IT-Unternehmen wie Banken ist es komplizierter. Sie entwickeln intern Software. Dafür benötigen sie die eingebetteten Tester für Scrum-Projekte plus die Quality Assurance-Funktion zur Begleitung. Gleichzeitig ist eine Ja/Nein-Entscheidung nicht nur bei intern entwickelter, sondern auch bei extern gekaufter Software notwendig. Von der Organisation her bietet sich ein klassisches Test Center an, das neu die Quality Assurance-Funktion für die Begleitung der agilen Entwicklung beinhaltet. Die Tester für Scrum-Projekte können als Pool dem Test Center angegliedert werden oder aber den verschiedenen Entwicklungsprojekten und Produkten.

Ist Scrum der Untergang der Test Center? Wenn Test Center die Produktion schützen wie bei einer Kunden, dann verändert Scrum bei ihnen nichts. Ist der Fokus von Test Centern auf Software-Entwicklung, gibt eine Reorganisation nach dem Motto „form follows function“ Sinn. Doch egal ob Test Center reorganisiert oder aufgelöst werden oder unverändert bleiben - auch mit Scrum benötigt Softwareentwicklung kompetente Tester. Scrum mag bessere Qualität für weniger Geld liefern. Der Ausspruch von Selfridge gilt trotzdem: „Qualität bleibt bestehen, wenn der Preis längst vergessen ist.“ ■



Klaus Haller arbeitet als Expert im Test-Consulting bei Swisscom IT Services in Zürich. Seine Schwerpunkte sind (Test-)Organisation und Prozesse, Testdatenmanagement, Compliance Testing und IT Risk/DLP.

Ziel	Organisationsform
Improve Quality	Eingebettete Tester in Scrum-Teams
Manage Risks	QA-Organisation (Prozess und/oder Testing)
Protect Operations	Unabhängiges Test Center

Abbildung 3: Ziele und passende Organisationsformen

LITERATURVERZEICHNIS [1] Agile Manifesto, <http://agilemanifesto.org/>, letzter Abruf 31.3.2013 [2] L. R. Vijayarathy, D. Turk: Agile Software Development: A Survey of early Adopters, Journal of Information Technology Management Volume XIX, Number 2, 2008 [3] Haberl, P., et al.: Umfrage 2011 - Softwaretest in der Praxis, dpunkt-Verlag, Heidelberg, 2012 [4] A. Page, K. Johnston, B. Rollison: How We Test Software at Microsoft, Microsoft Press, Redmond, WA, 2009 [5] J. Whittaker, J. Arbon, J. Carollo: How Google Tests Software, Addison-Wesley, Westford, MA, 2012

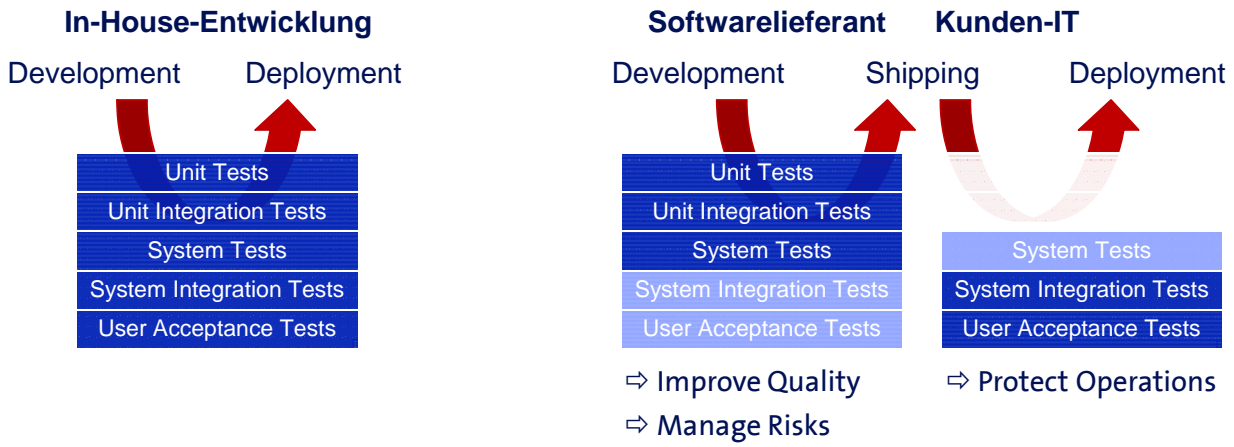


Abbildung 2: Testbedürfnisse bei In-house Softwareentwicklung und bei Standard-Software/Externer Software-Entwicklung