



□ Prof. Dr. Mario Winter

(E-Mail: mario.winter@fh-koeln.de)

lehrt am Institut für Informatik der FH Köln Softwareentwicklung und Projektmanagement und ist dort Mitglied des Forschungsschwerpunktes „Softwarequalität“. Die Software-Qualitätssicherung bildet seit mehr als 20 Jahren den Schwerpunkt seiner wissenschaftlichen Arbeit. Im Rahmen seines Engagements für das GTB setzt sich Professor Winter für die Themen Software-Test, -Analyse und -Verifikation ein. Er ist seit 2003 Sprecher der entsprechenden Fachgruppe der Gesellschaft für Informatik (GI). Auch als Autor und Co-Autor mehrerer Bücher sowie vieler Fachartikel und Konferenzbeiträge zum Thema Softwaretest hat sich Mario Winter einen Namen gemacht.

Agile Software-Projekte brauchen qualifizierte Tester Entwicklung und Test wachsen zusammen

Etablierte Vorgehensmodelle zur iterativ-inkrementellen Software-Entwicklung stehen im Ruf, oftmals zu unflexibel und träge zu sein. Agile Entwicklungsansätze hingegen versprechen, schneller Software mit höherer Qualität zu erhalten. Was bedeutet dieser Paradigmenwechsel für das Software-Testen? Eines ist sicher: Da agile Entwicklung immer auf kurzen Release-Zyklen mit entsprechend häufig auszuführenden Tests basiert und in der Regel auch testgetriebene Entwicklung bedeutet, benötigt sie entsprechend geschulte Programmierer und Tester.

Agiles Testen ist kein Solitär. Es steht immer in direkter Verbindung zu agiler Entwicklung. Hat ein Unternehmen in einem Entwicklungsprojekt etwa die Absicht, bei der Entwicklung „Business as usual“ walten zu lassen und nur das Testen agil durchzuführen, bestehen kaum Chancen für eine erfolgreiche Umsetzung. Ein solches Projekt wird mit hoher Wahrscheinlichkeit seine Qualitäts-, Budget- und Terminvorgaben nicht einhalten können. Denn es bleibt dabei zu wenig Zeit für die in einem solchen Rahmen meist als letzten Schritt vorgesehenen Testphasen. Die Testzyklen sind dann zwar kürzer als üblich und werden als „agil“ bezeichnet, erfüllen aber in den seltensten Fällen die Kriterien, nach denen Agiles Testen definiert ist. Agiles Testen muss vielmehr immer integraler Bestandteil agiler Software-Entwicklung sein, und zwar von Projektbeginn an. Nur wenn die Entwicklung agil aufgesetzt wird, ist auch Agiles Testen sinnvoll.

Um Wissen über die Erfordernisse und Einzelheiten agiler Projekte zu vermitteln, kann die Ausbildung zum *Certified Tester* gemäß den Standards des International Software Testing Qualifications Board (ISTQB, [1]) sehr nützlich sein. Sie schafft die Grundlage für eine gemeinsame Begrifflichkeit – quasi die „agile Sprache“ – und unterstützt damit den bei agilen

Projekten erhöhten Bedarf an Kommunikation, besonders in Bezug auf Tests und Qualitätssicherung. Obligatorischer Bestandteil des Ausbildungsprogramms sind sowohl sämtliche Teststufen, von Komponenten- bis Abnahmetest, als auch die verschiedenen Testarten wie z.B. funktionale und nicht-funktionale Tests.

Darüber hinaus vermittelt die Certified-Tester-Ausbildung Methoden, Techniken und Best Practices, die sich in agilen Projekten ebenso gut anwenden lassen wie in Großprojekten, wenn sie in ausreichend feingranularer Form heruntergebrochen werden. Dann sind diese Vorgehensweisen sehr geeignet, um die Tests spezieller Funktionalitäten und Features vorzunehm-

men, die es jeweils im Rahmen eines agilen Release auszuführen gilt.

Verkürzte Release-Zyklen

Ein wesentlicher Unterschied zu herkömmlichen Projekten besteht in den sehr viel kürzeren agilen Release-Zyklen (**Abb. 1**). Sie werden statt in Quartalen oder Monaten eher in Wochen oder Tagen gemessen. Die einzelnen Release-Zyklen beziehen sich dabei immer auf kleinere Systeme beziehungsweise Systemteile. Wenn ein solches Software-Produkt erst lauffähig geworden und beim Kunden – respektive der auftraggebenden Fachabteilung im eigenen Haus – im Einsatz ist, wird es stetig agil erweitert.

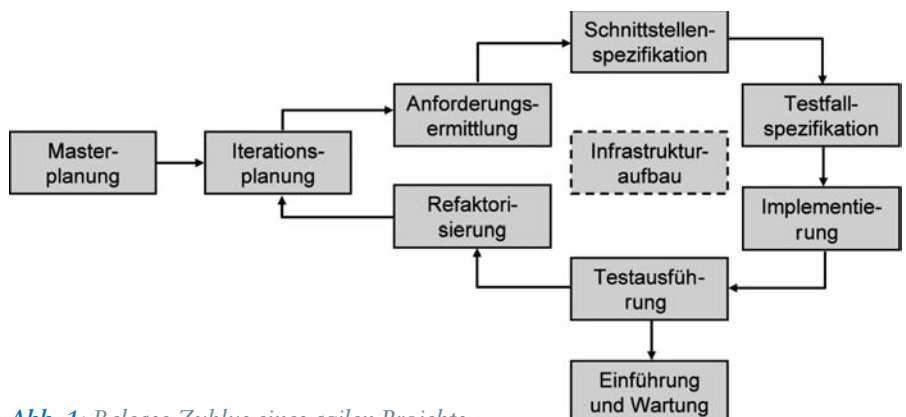


Abb. 1: Release-Zyklus eines agilen Projekts

Auf diese Weise kann sich Agiles Testen auch in Großprojekten lohnen, wenn auf Basis einiger weniger Kernanforderungen entwickelt wird und sich das Großprojekt sinnvoll in verschiedene kleine Teilprojekte aufspalten lässt. Unabhängig von der Projektgröße muss die Entwickler- und Testermannschaft inklusive Auftraggeber immer ein „eingeschworenes“ agiles Team bilden und in einem überschaubaren Rahmen agieren können. Dann sind die Voraussetzungen dafür gegeben, agile Techniken wie Extreme Programming einzusetzen, gemeinsam Code zu erarbeiten und tägliche Integrationstests durchzuführen. Dies erfordert in Großprojekten zwangsläufig ein Multiprojekt-Management und ein entsprechend angepasstes Testmanagement. Der Nutzen agiler Methoden in Großprojekten ist Gegenstand aktueller Debatten und Untersuchungen in der Forschung, ein abschließendes Ergebnis liegt noch nicht vor.

Grundsätzlich aber bedeutet Agiles Testen zunächst immer Testen in kleineren Projekten, wo sich seine Vorteile anhand von zahlreichen Studien eindeutig belegen lassen. Jedoch erfordert es geschultes Personal, damit bei gleichen Qualitätsanforderungen in kürzeren Abständen getestet werden kann. Agile Tester müssen geeignete Testfälle schneller erstellen und auch schneller durchführen können. Somit bedeutet Agiles Testen immer auch Testautomatisierung. Denn die erheblich verkürzten Release-Zyklen funktionieren nur unter der Voraussetzung, dass nicht nur die Tests der einzelnen Softwaremodule, sondern auch System- und möglichst viele Abnahmetests mit entsprechenden Werkzeugen automatisiert worden sind. Zudem haben automatisierte Testfälle gegenüber manuellen Tests den Vorteil, auch nachts ablaufen und dokumentiert werden zu können.

Erfolgsfaktor Automatisierung

Ebenso gehören automatisierte Regressionstests aufgrund der verkürzten Release-Zyklen zu den Schlüsselfaktoren der Agilität. Denn in jedes Release werden neue Features beziehungsweise User-Stories eingebaut und vorhandene erweitert oder angepasst. Dies kann wegen der knappen Zeitvorgaben nur mit automatisierten, regressionsfähigen Tests abgesichert werden. Zwar ist die Testautomatisierung zunächst mit einem gewissen Aufwand verbunden, da sie auch Unternehmensprozesse berührt. Dafür ermöglicht sie eine schnellere Erfolgskontrolle, da die Werkzeuge ein unmittelbares Feedback über die automatisierten Testläufe geben. Der Mehrwert von

Agilität und Automatisierung ist so für alle Beteiligten nachvollziehbar, und die Fachbereiche erhalten schneller ein vermarktbare Produkt.

In der grundlegenden Phase der Komponententests sollte Automatisierung ohnehin die Regel sein. So erstellen Entwickler und Tester zunächst Komponententests, um entsprechend dieser Testfälle zu entwickeln und Code zu schreiben. Hierbei werden die von den zu testenden Komponenten benutzten Komponenten aus Aufwandsgründen oft nicht durch eigens entwickelte Teststellvertreter (Mock-Objekte, Stubs) ersetzt, sondern unmittelbar für den Test herangezogen (dies wird dann von vielen Entwicklern als ausreichender Ersatz für spezielle Integrationstests angesehen, obwohl letztere vollkommen andere Testziele bzw. Testbedingungen adressieren). Nach Beendigung der Codierung werden die Tests sofort automatisch ausgeführt. Hieran zeigt sich auch, dass agile Entwicklung ihrem Wesen nach *testgetriebene* Entwicklung ist.

Anders liegt der Fall häufig bei System- und Abnahmetests. Sie sind meist vom Auftraggeber/Kunden definiert und können so häufig nicht automatisch ausgeführt werden. Ihre stiefmütterliche Behandlung erklärt sich aus der verbreiteten Annahme beziehungsweise Hoffnung, die in den vorherigen Teststufen erzeugte Qualität werde auf das Gesamtprodukt durchschlagen, so dass beim Abnahmetest keine nennenswerten Modifikationen mehr nötig seien. Bei agilen Projekten hingegen ist es immer erforderlich, nicht nur Komponententests, sondern auch Integrations-, System- und Abnahmetests mithilfe von Werkzeugen zu automatisieren. Andernfalls ließen sich die angestrebten kurzen Release-Zyklen nicht durchhalten.

Test- und Projektmanagement im agilen Verständnis

Dies bedeutet auch, dass der Grundsatz der Agilität in sämtliche Aktivitäten zur Planung einfließen muss. So lässt sich beispielsweise zu Projektbeginn noch kein ausführliches Testkonzept mit detaillierten Anforderungen (Requirements) erstellen. Vielmehr muss auch das Testkonzept agil konzipiert und für jedes Release im Rahmen der verkürzten Zyklen fortgeschrieben werden. Einer der Grundwerte der Agilität aus dem „agilen Manifest“ [2], das 2001 von 17 „Gründervätern“ der agilen Software-Entwicklung formuliert und bis heute von tausenden weiteren IT-Experten unterzeichnet wurde, besagt eindeutig, dass Programmcode über Doku-

mentation geht: Funktionierende Programme gelten mehr als ausführliche Dokumentationen (und werden definiert als „Programme, welche die automatisierten Tests bestehen“).

Daraus folgt unter anderem, dass die agilen Team-Mitglieder bei ihrer Testplanung immer schon das Folge-Release (Increment) einplanen sollten. Da die meisten Testfälle mithilfe von Werkzeugen automatisiert sind, sollten sich Testergebnisse problemlos nachverfolgen und analysieren lassen. Es stehen so sehr zeitnah Daten zur Verfügung, die Aufschluss darüber geben, wie gut der Durchlauf der Testfälle war. Durch diese Automatisierungskette entstehen gleichzeitig quasi ein Sicherheitsnetz sowie ein objektiver, jederzeit aktueller Fortschrittsanzeiger für das agile Projekt.

Um die Tests in kleinere Einheiten unterteilen zu können, haben sich mittlerweile aus agilen Projekterfahrungen gewonnene Methoden wie SCRUM bewährt. Bei SCRUM handelt es sich um ein iteratives, inkrementelles Verfahren, das nach jeder Iteration beziehungsweise jedem Release einen Satz von lieferbaren Funktionalitäten bereitstellt. Eine charakteristische SCRUM-Vorgehensweise besteht etwa darin, Teile des Product Backlogs (Arbeitspaket) als die im nächsten Release-Zyklus umzusetzenden Features oder UserStories in einen *Sprint* (Iterationsfrist) zu übertragen. So ergibt sich im agilen Testmanagement ein fortlaufender Test- und Projektplan. Dieser formuliert zielgerichtet die für das jeweils anstehende Release nötigen Testaufgaben und dient zugleich als Kontrollinstanz. Etablierte Testverfahren wie z. B. die Äquivalenzklassenbildung helfen dann beispielsweise, eine User-Story, die der Kunde im Projektteam umgangssprachlich formuliert hat, strukturiert und adäquat in einen Testfall umzusetzen und die geeigneten Parameter und Wertebereiche dafür zu bestimmen [3].

Agiles Testen setzt mithin die Bereitschaft des Managements voraus, ein Projekt anzugehen, bei dem zunächst nur Kernanforderungen abgesteckt werden können und weitere Requirements erst während der Projektdurchführung definiert werden. Dementsprechend flexibel sollte man auch Budget- und Zeitpläne inklusive Abnahmetermenin für das Endprodukt handhaben. Agil entwickeln heißt, das Endprodukt in kleineren, aber häufigeren Release-Zyklen zu erarbeiten, deren Umfang zu Projektbeginn noch nicht genau geplant werden kann.

Kommunikation ist (fast) alles

Grundsätzlich kommt in agilen Projekten Individuen und Interaktionen ein höherer Stellenwert zu als Prozessen und Werkzeugen [2]. Agiles Testen stellt erhebliche Ansprüche an die Kommunikation im Projektteam. Es erfordert eine spezielle Kultur der Verständigung und den täglichen, ja ständigen Austausch innerhalb eines Teams, das im gleichen Raum Hand in Hand arbeitet. Von großer Bedeutung ist die Einigung auf eine gemeinsame Projekt-Terminologie, bei der ausgebildete *Certified Tester* eine maßgebliche Rolle spielen und ihre Stärken einbringen können, wie beispielsweise bei der Formulierung sinnvoller Abnahmetests. Gerade mit Blick auf diese finale Testphase ist es von besonderer Bedeutung, mit dem ebenfalls im agilen Team involvierten Kunden eine gemeinsame Sprache zu sprechen.

Zudem können agile Techniken wie Refactoring oder Pair Programming nur optimal funktionieren, wenn Tester und Entwickler mit- und nicht gegeneinander arbeiten und in der Lage sind, sich über das gemeinsame Tun ständig auszutauschen. Eine Methode wie SCRUM legt darüber hinaus bestimmte Rollen für Team-Mitglieder fest, vom SCRUM-Master bis zu Entwicklern und Testern, und bindet in der Rolle des *Product Owner* auch Vertreter des Kunden beziehungsweise der beauftragenden Fachabteilung ein. Die fachliche Perspektive des Kunden müssen agile Tester bei der Konzipierung der Testfälle immer berücksichtigen. Daher ist es sinnvoll, dass alle Team-Mitglieder einschließlich der Entwickler in Bezug auf Tests und Qualitätssicherung eine gemeinsame Terminologie nutzen und die Begrifflichkeit des *Certified Tester* zumindest auf Foundation Level kennen.

Darüber hinaus gilt es bei der Entwicklung neuer Software-Produkte für den Markt immer, Marketing und Produktmanagement agil einzubinden. Unverzichtbar ist schließlich die Aufmerksamkeit und Unterstützung des mittleren und oberen Managements und dessen Bekenntnis und Bereitschaft zur Agilität. Die Leitungsebene muss den Mut haben, auf die festgeklopften Anforderungen der bekannten Fixpreis-Projekte – die sich erfahrungsgemäß dann im Projektverlauf doch ändern – zu verzichten. Wie es ein weiterer Grundsatz der Agilität ausdrückt: „Die stetige Zusammenarbeit mit dem Kunden ist wichtiger als ausgefeilte Verträge.“ [2] Um in einem Unternehmen ein günstiges Umfeld für Agilität und testgetriebene Entwicklung zu schaffen, empfiehlt es sich, den ersten agi-

len Versuch in einem nicht zu geschäftskritischen, kleineren Projekt zu wagen.

Last, but not least kommt der Zusammensetzung des Teams in agilen Projekten ebenfalls eine herausragende Rolle zu. Es ist nicht ratsam, ein bereits bestehendes, hierarchisch strukturiertes Team mit einem agilen Projekt zu beauftragen. Erfahrungsgemäß ist es sinnvoller, bei der Projektinitialisierung und Teambildung den Rat von (externen) Experten in Anspruch zu nehmen, die bereits agil tätig waren und Erfahrung in der Beurteilung von agilen Teams haben.

Gemeinsames Anforderungsmanagement

Dies ist nicht zuletzt deswegen von Bedeutung, weil das agile Team sich unter Mitwirkung aller Beteiligten seine Anforderungen selbst stellt und die Anforderungsdefinition gemeinsam schreibt. Deren integraler Bestandteil sind die Testfälle, welche die Anforderungen ja erst präzisieren. Beim herkömmlichen Vorgehen ist es umgekehrt, dort werden die Testfälle durch die Anforderungen bestimmt. Agile, testgetriebene Entwicklung ist hingegen dadurch charakterisiert, dass die Testfälle quasi die Spezifikation bilden, gegen die entwickelt wird.

Manche Beobachter betrachten dies als Schwachpunkt der Agilität, da das Vier-Augen-Kontrollprinzip verletzt werde, wenn der Entwickler schon „durch die Testerbrille“ entwickelt – und der Tester umgekehrt durch die Entwicklerbrille testet. In der Regel werden die Testfälle jedoch gerade durch die testgetriebene Entwicklung so gut, dass auch das Endprodukt die geforderte Qualität erreicht. Die Testfälle bilden als integraler Bestandteil der Anforderungen eine Art Versicherung für Anforderungs- und Test-Management, welche organisatorisch sehr stark zusammenwachsen. Hieran zeigt sich der Charakter des agilen Testens als „Mauerbrecher“: Das übliche Pingpongspiel zwischen Fachseite, Entwicklern und Testern verwandelt sich in ein Miteinander.

Zertifizierte agile Tester

Einen wesentlichen Anteil daran haben zertifizierte Tester, zu deren wichtigsten Aufgaben es gehört, mit einer einheitlichen Testbegrifflichkeit für eine störungsfreie Teamkommunikation zu sorgen, ob in agilen oder herkömmlichen Projekten. Zu den weiteren Inhalten, die ein Anbieter von qualifizierten und vom German Testing Board (GTB) geprüften Tester-Ausbildungen gemäß

ISTQB-Standard auf dem Foundation Level vermitteln können muss, zählen Kenntnisse sämtlicher Teststufen, ob Komponenten-, System-, Integrations- oder Abnahmetests sowie der wesentlichen Testarten wie z. B. funktionale und nicht-funktionale Tests.

Zudem befähigt die Schulung zum *Certified Tester (CT)* dazu, Risiken einschätzen und so entscheiden zu können, wann es etwa vertretbar ist, ein Feature *nicht* bis ins letzte Detail zu testen. Zertifizierte Tester sollten darüber hinaus in der Lage sein, die gelernten Methoden auf allen Granularitätsstufen anzuwenden, und Kenntnisse über die prozessualen Aspekte des Testens aufweisen. CT-Ausbildungen vermitteln auf allen Levels Vorgehensweisen und Best Practices, die in Projekten jeder Art zu verwenden sind und auch für Software-Entwickler sehr nützlich sein können. Besonders gut auf agile Projekte übertragen lassen sich die Rollen und Aufgaben, die auf *Advanced Level* dem (eher fachlich orientierten) ‚Test Analyst‘ und dem ‚Technical Test Analyst‘ zukommen.

Grundsätzlich ist das Ausbildungsschema für CT vorgehensmodellneutral und enthält somit auch Anteile, die für agile Projekte eher nicht nutzbar sind. Doch bieten die Lehrpläne genügend Raum für die Einbindung des agilen Testens. Unternehmen, die an einem speziellen agilen Training/Coaching für Mitarbeiter interessiert sind, ist aber ohnehin zu empfehlen, dieses in Absprache mit einem geeigneten Anbieter entsprechender Schulungen unternehmens- oder projektspezifisch auszurichten, um daraus einen Mehrwert ziehen zu können. Wer einen praktischen „Schnupperkurs“ in Agilität absolvieren möchte, kann auch mit Hilfe erfahrener Trainer ein nicht geschäftskritisches Pilotprojekt aufsetzen und gleichzeitig eine *Foundation Level*-Zertifizierung anstreben. Die vom GTB akkreditierten Seminaranbieter stellen entsprechend geknüpft Leistungspakete zur Verfügung. ■

Mehr Infos rund um die *Certified-Tester*-Ausbildung bietet:

www.german-testing-board.info

Literaturangaben

- [1] International Software Testing Qualifications Board. <http://www.istqb.org/>
- [2] Agile Manifesto. <http://agilemanifesto.org/>
- [3] Falk Fraikin, Matthias Hamburg, Stefan Jungmayr, Thomas Leonhardt, Andreas Schönknecht, Andreas Spillner, Mario Winter: Die trügerische Sicherheit des grünen Balkens. OBJEKTSpektrum, Heft 1, 2004, S. 46-50