Manfred **BAUMGARTNER**
Martin **KLONK**
Christian **MASTNAK**
Richard **SEIDL**

AGILE TESTING

Der agile Weg zur Qualität

3. Auflage

Im Internet: Mit begleitender Homepage
www.agile-testing.eu

HANSER

Baumgartner/Klonk/Mastnak/Seidl

Agile Testing

Manfred Baumgartner
Martin Klonk
Christian Mastnak
Richard Seidl

Agile Testing

Der agile Weg zur Qualität

3., überarbeitete Auflage

HANSER

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Lektorat: Brigitte Bauer-Schiewek
Copy editing: Petra Kienle, Fürstenfeldbruck
Umschlagdesign: Marc Müller-Bremer, www.rebranding.de, München
Umschlagrealisation: Max Kostopoulos
Titelmotiv: © gettyimages.de/Daniela Garling
Satz: Eberl & Koesel Studio, Kempten
Druck und Bindung: Hubert & Co. GmbH & Co. KG BuchPartner, Göttingen
Printed in Germany

Print-ISBN 978-3-446-47767-4
E-Book-ISBN 978-3-446-47841-1
E-Pub-ISBN 978-3-446-48030-8

Inhalt

Geleitwort

Im Winter 2001 rief eine verschworene Clique bekannter Softwareentwickler in einer abgelegenen Skihütte im Bundesstaat Utah zu einer Revolution in der Softwarewelt auf. Sie schufen das Agile Manifest. Mit diesem Manifest definierte die Gruppe, was sie mit Extreme Programming bereits praktizierte. Mit ihrer schriftlichen Formulierung gelang ihnen ein publizistischer Coup, der weltweit Aufmerksamkeit erregte. Die auf dieser Skihütte versammelten Entwicklungsexperten hatten genug von starren Prozessregeln, unsinnigen bürokratischen Richtlinien und weltfremden Vorgehensweisen der damaligen Software-Engineering-Disziplin. Sie erkannten, dass monotones „Arbeiten nach Vorschrift“ in der neuen, schnelllebigen Zeit nicht mehr zeitgemäß war. Sie wollten sich von den Fesseln der Projektbürokratie befreien, um Software gemeinsam mit den Anwendern nach Bedarf zu entwickeln. Die bisher schwerfällige, phasenorientierte, dokumentengetriebene Softwareentwicklung sollte durch eine flexible, menschengesteuerte Entwicklung in kleinen, überschaubaren Schritten ersetzt werden. Agile Softwareentwicklung sollte der Ansatz des neuen Jahrhunderts sein!

Bei der agilen Entwicklung steht nicht das Projekt, sondern das Produkt im Mittelpunkt. Da die Softwareentwicklung mehr und mehr zu einer Expedition ins Unbekannte wurde, sollte das Produkt nach und nach in kleinen, inkrementellen Schritten erstellt werden. Anstatt ausufernde Anforderungsdokumente über Dinge zu schreiben, die man zu diesem Zeitpunkt noch nicht wissen konnte, sollte man lieber etwas programmieren, das einem zukünftigen Benutzer schnelles Feedback entlocken kann. Es sollte nicht Monate oder gar Jahre dauern, bis man feststellt, dass das Projekt auf dem falschen Weg oder das Projektteam mit der Aufgabe überfordert ist. Dies sollte innerhalb weniger Wochen erkannt werden.

Das Grundprinzip der agilen Entwicklung ist also die inkrementelle Bereitstellung. Ein Softwaresystem soll Stück für Stück fertiggestellt werden. Dies gibt dem Benutzervertreter im Team die Gelegenheit, sich an jedem Schritt der Entwicklung zu beteiligen. Nach jeder neuen Lieferung kann er das gelieferte Zwischenprodukt mit seinen Vorstellungen abgleichen. Das Testen ist somit in den Prozess integriert. Von Anfang an wird die Software kontinuierlich getestet. Ursprünglich wurde offengelassen, ob ein Tester auch am agilen Prozess teilnehmen sollte. Die Autoren des Agilen Manifests sprachen sich gegen eine strikte Arbeitsteilung aus. Die Aufteilung in Analysten, Designer, Entwickler, Tester und Manager erschien ihnen zu künstlich und sie befürchteten, dass dadurch zu viele Reibungsverluste entstehen. Natürlich sollte das Projektteam über all diese Fähigkeiten verfügen, aber die Rollen innerhalb des Teams sollten austauschbar sein. Das Entwicklungsteam sollte für alles verantwortlich sein. Die Rolle eines speziellen Testers im Team wurde erst durch die

Beiträge von Lisa Crispin und Janet Gregory eingeführt. Sie setzten sich dafür ein, dass sich jemand im Team um Qualitätsfragen kümmert.

Die Entwicklung von Software erfordert sowohl Kreativität als auch Disziplin. Gegen Ende des letzten Jahrhunderts hatten die Verfechter von Ordnung und Disziplin die Oberhand und bremsten mitunter die Kreativität der Entwickler durch starre Prozesse und Qualitätssicherungsmaßnahmen aus. Doch wenn etwas übertrieben wird, schlägt das Pendel zurück. Zu viel Disziplin wurde den traditionellen Entwicklungsprojekten aufgezwungen. Die Reaktion darauf ist die agile Bewegung, die Spontaneität und Kreativität in die Softwareentwicklung zurückbringen soll. Das ist sicherlich zu begrüßen, aber auch dieser Aspekt darf nicht übertrieben werden. Die Kreativität von Anwendern und Entwicklern sollte geerdet bleiben – und ein erfahrener, unparteiischer Tester kann dies unterstützen.

Jedes Entwicklungsteam sollte mindestens einen Tester haben, der die Interessen der Qualität vertritt. Dieser stellt sicher, dass der resultierende Code und das Produkt sauber bleiben und die vereinbarten Qualitäts- oder Abnahmekriterien erfüllen. In dem Drang, schneller voranzukommen, können nichtfunktionale Qualitätsanforderungen hinter den funktionalen Anforderungen zurückbleiben. Es ist die Aufgabe eines Testers, dem Team zu helfen, ein Gleichgewicht zwischen Produktivität und Qualität zu wahren. Der Tester ist quasi der gute Geist, der das Team davor bewahrt, Fortschritte auf Kosten der Qualität zu machen. Mit jedem neuen Release soll nicht nur Funktionalität hinzugefügt, sondern auch die Qualität erhöht werden. Der Code soll regelmäßig bereinigt bzw. refaktoriert, dokumentiert und von allen Mängeln befreit werden. Es ist die Aufgabe des Testers, das ganze Team dazu zu befähigen und dafür zu sorgen, dass dies geschieht.

Natürlich hat die agile Projektorganisation auch Auswirkungen auf das Testen und die Qualitätssicherung. Die für die Qualitätssicherung zuständigen Personen sitzen nicht mehr in einem entfernten Büro, von wo aus sie die Projekte überwachen, die Projektergebnisse zwischen den Phasen prüfen und das Produkt in der letzten Phase testen, wie es bei traditionellen Projekten oft der Fall war. Sie sind jetzt fest in die Entwicklungsteams integriert, wo sie ständig die neuesten Ergebnisse überprüfen und validieren. Es ist ihre Aufgabe, auf Mängel in der Architektur und im Code hinzuweisen und eventuelle Fehler im Verhalten des Systems zu erkennen. Sie weisen auf Probleme hin und helfen dem Team, die Qualität der Software zu verbessern. Die Rolle des Testers entwickelt sich zu einem agilen Qualitätscoach, der das Team in allen qualitätsrelevanten Fragen unterstützt. Im Gegensatz zu dem, was in den Anfängen der agilen Bewegung von einigen behauptet wurde („Tester sind in agilen Projekten nicht mehr notwendig“), ist ihre Rolle heute wichtiger denn je. Ohne ihren Beitrag wachsen die technischen Schulden und diese bringen das Projekt früher oder später zum Stillstand.

Dieses Buch beschreibt das agile Testen in zehn Kapiteln. Das erste Kapitel schildert den Kulturwandel, den die agile Entwicklung mit sich gebracht hat. Mit dem Agilen Manifest wurden die Weichen für eine Neugestaltung der IT-Projektlandschaft gestellt. Die Entwicklung sollte nicht mehr starr nach dem Phasenkonzept erfolgen, sondern flexibel und in kleinen Iterationen. Nach jeder Iteration ist vom Team ein lauffähiges Teilprodukt zu präsentieren. Auf diese Weise werden Lösungen erforscht und Probleme frühzeitig erkannt. Die Rolle der Qualitätssicherung ändert sich. Anstatt als externe Instanz für die Projekte zu agieren, werden die Tester in das Projekt eingebettet, so dass sie ihre Tests unmittelbar vor Ort als Teilnehmer am Entwicklungsprozess durchführen können. Natürlich müssen die Unternehmen ihre Managementstrukturen entsprechend anpassen: Anstatt abseits auf ein

Endergebnis zu warten, sind die Fachanwender gefordert, sich aktiv am Projekt zu beteiligen und die Entwicklung über ihre Anforderungen, die „Storys“, zu steuern. Auf der Entwicklungsseite arbeiten sie mit den Entwicklern zusammen, um die gewünschte Funktionalität zu analysieren und zu spezifizieren. Auf der Testseite arbeiten sie mit Testern zusammen, um sicherzustellen, dass das Produkt ihren Erwartungen entspricht.

Letztlich müssen sich alle – Entwickler, Tester und Benutzer – anpassen, um das gemeinsame Ziel zu erreichen. Viele traditionelle Rollen, wie die des Projektleiters und des Testmanagers, sind im Schwinden oder zumindest im Wandel begriffen. Dafür gibt es neue Rollen, wie Scrum Master und Lead Tester oder Agile Quality Coach. Projektmanagement im traditionellen Sinne findet nicht mehr auf Teamebene statt. Jedes Team verwaltet sich selbst. Die IT-Welt verändert sich und damit auch die Art und Weise, wie Menschen Software entwickeln.

Im zweiten Kapitel, das sich mit agilen Vorgehensmodellen befasst, konzentrieren sich die Autoren auf die Rolle der Qualitätssicherung in agilen Entwicklungsprojekten. Sie scheuen sich nicht, die verschiedenen Zielkonflikte objektiv zu betrachten, zum Beispiel zwischen Qualität und Termintreue, zwischen Qualität und Budget und zwischen Qualität und Funktionalität. Die Vereinbarkeit dieser Zielkonflikte ist eine der Herausforderungen für das agile Testen.

Entgegen der immer noch weit verbreiteten Meinung, dass in agilen Projekten nicht viel getestet werden muss, ist in Wirklichkeit sehr viel Testarbeit erforderlich. Testgetriebene Entwicklung (TDD) sollte nicht nur für den Unittest, sondern auch für den Integrationstest und den Systemtest gelten, nach dem Motto: erst die Testfälle, dann der Code. In diesem Fall heißt das: erst die Testspezifikation, dann die Implementierung. Die Testautomatisierung spielt dabei eine entscheidende Rolle. Nur wenn ein Test automatisiert wird, kann die geforderte Qualität in der nötigen Geschwindigkeit erreicht werden. An der Automatisierung sollte das gesamte Team beteiligt sein, denn ein Tester allein wird es nicht bewältigen können. Neben dem Test sind zu bestimmten Zeiten während der Entwicklung des Softwareprodukts auch Audits erforderlich. Ziel der Audits ist es, Schwachstellen und Mängel in der Software aufzudecken. Ein guter Zeitpunkt dafür ist am Ende jedes Sprints in einem Scrum-Projekt (Retrospektive). Das Team legt dann anhand der Ergebnisse der Audits die Prioritäten für den nächsten Sprint fest. Diese kurzen Audits, oder Momentaufnahmen der Produktqualität, können von externen QS-Experten in Zusammenarbeit mit dem Team durchgeführt werden. Ziel ist es, dem Team zu helfen, Risiken rechtzeitig zu erkennen.

Neben dem Scrum-Prozess wird im zweiten Kapitel auch auf Kanban und den schlanken Softwareentwicklungsprozess (Lean Software Development) eingegangen. Anhand von Beispielen aus der Projekterfahrung der Autoren erhält der Leser zahlreiche Tipps, wie er die Qualitätssicherung in diese Prozesse einbinden kann.

Das dritte Kapitel beschäftigt sich mit der agilen Testorganisation und der Positionierung des Testens in einem agilen Team. Zu diesem Thema gibt es recht unterschiedliche Ansichten. Mit Hilfe der vier Testquadranten von Crispin und Gregory untersuchen die Autoren, welcher Test zu welchem Zweck passt. Dabei wird zum einen die Frage gestellt, ob der Test geschäfts- oder technologieorientiert ist, und zum anderen, ob er sich auf das Produkt oder das Team bezieht. Daraus lassen sich vier Testarten ableiten:

1. Unit- und Komponententest: technologieorientiert/teamunterstützend
2. Funktionstest: geschäftsorientiert/teamunterstützend
3. Explorativer Test: geschäftsorientiert/produktbezogen
4. Nicht-funktionaler Test:.technologieorientiert/produktbezogen

Zur Verdeutlichung dieser Testansätze wird anhand von Beispielen aus der Testpraxis gezeigt, welche Art von Test welchem Zweck dient.

In diesem Kapitel gehen die Autoren auch auf das Thema Test in skalierenden, agilen Modellen und Frameworks, wie SAFe oder LeSS ein. Sie betonen dabei, wie wichtig es ist, den Testprozess nach Belieben erweitern zu können. Es gibt Kernaktivitäten, die stattfinden müssen, und Randaktivitäten, die je nach Ausbaustufe hinzugefügt werden. Es gibt also nicht nur eine, sondern viele mögliche Organisationsformen, die von der Art des Produkts und den Projektbedingungen abhängen.

Das Umfeld, in dem das Projekt stattfindet, und die Produkteigenschaften wie Größe, Komplexität und Qualität sind entscheidend für die Wahl der geeigneten Organisationsform. In jedem Fall darf das Hauptziel, nämlich die Unterstützung der Entwickler, nicht aus den Augen verloren werden. Das Ziel aller Testansätze ist es, Probleme möglichst schnell und gründlich aufzudecken und die Entwickler auf unaufdringliche Weise zu informieren. Wenn mehrere agile Projekte nebeneinander laufen, empfehlen die Autoren die Einrichtung eines Test Competence Center. Aufgabe dieser Instanz ist es, die Teams in allen Fragen der Qualitätssicherung zu unterstützen, zum Beispiel welche Methoden, Techniken und Werkzeuge sie einsetzen sollten. Am Ende des Kapitels werden zwei Fallstudien zur Testorganisation vorgestellt, eine aus dem Telekommunikations- und eine aus dem Gesundheitsbereich. In beiden Fällen orientiert sich die Testorganisation an der Projektstruktur und den jeweiligen Qualitätszielen.

In Kapitel 4, „Die Rolle des Testers in agilen Projekten“, wird die Frage aufgeworfen, ob ein agiler Tester ein Generalist oder ein Spezialist sein sollte. Die Antwort lautet, wie so oft in der Literatur zur agilen Entwicklung: sowohl als auch. Es kommt auf die jeweilige Situation an. Es gibt Situationen, wie z. B. zu Beginn einer Iteration, in denen der Tester mit dem Benutzer oder dem Product-Owner über Akzeptanzkriterien verhandelt, in denen der Tester sowohl technisches als auch allgemeines Wissen benötigt. Es gibt andere Situationen, in denen der Tester mit automatisierten Testwerkzeugen umgehen muss, für die er spezielle technische Kenntnisse benötigt. Ein agiler Tester muss in der Lage sein, viele Rollen zu übernehmen, doch das Wichtigste ist, dass der Tester sich als Teamplayer in das Team einfügt, ganz gleich, welche Rolle er gerade übernehmen muss. Soft Skills sind eine unabdingbare Voraussetzung. In jedem Fall sind die Tester die Verfechter der Qualität und müssen dafür sorgen, dass die Qualität erhalten bleibt, auch wenn die Zeit drängt. Dazu müssen sie an allen Diskussionen über die Produktqualität teilnehmen und gleichzeitig die Software prüfen und testen. Sie sollten Probleme rechtzeitig aufdecken und dafür sorgen, dass sie so früh wie möglich behoben werden. Natürlich können die Tester dies nicht allein tun; sie sind auf den Beitrag der anderen Teammitglieder angewiesen. Deshalb müssen Tester als eine Art Qualitätscoach fungieren und ihren Teamkollegen helfen, Probleme zu erkennen und zu lösen. Schließlich liegt die Qualität der Software in der Verantwortung des gesamten Teams.

Im Zusammenhang mit der Rolle des Testers in einem agilen Team befasst sich das Kapitel mit dem Erfahrungsprofil der Mitarbeiter. Wie sehen die Karrieremodelle in der agilen Welt

aus? Tatsache ist, dass es in der agilen Entwicklung keine festen Rollen mehr gibt. Die Rollen ändern sich je nach Situation, auch die des Testers. Mitarbeiter, die ausschließlich Erfahrungen mit traditionellen Entwicklungsmethoden haben, können sich nicht mehr auf traditionelle Rollen zurückziehen, sondern müssen sich anpassen. Dies wird nicht für jeden Mitarbeiter einfach sein. Die Autoren schlagen ein Trainingsprogramm vor, das auf die Rolle des agilen Testers vorbereitet. In dem Programm betonen sie positive Erfahrungen und schließen mit der Zuversicht, dass flexible Mitarbeiter, ob alt oder jung, in die Rolle des agilen Testers hineinwachsen können.

In Kapitel 5 wenden sich die Autoren den Methoden und Techniken des agilen Testens zu. Dabei betonen sie die Unterschiede zum herkömmlichen, phasenorientierten Testen. Der Prozess beginnt mit der Testplanung, wobei der Plan viel unverbindlicher ist als in traditionellen Projekten. Er sollte flexibel bleiben und leicht zu aktualisieren sein. Agiles Testen ist viel stärker mit der Entwicklung verwoben und kann nicht mehr separat als Teilprojekt im Projekt betrachtet werden. In jedem Entwicklungsteam sollte es mindestens einen Tester geben, der dort voll integriert ist. Die Tester sollten nur dem Team gegenüber verantwortlich sein. Es kann einen projektübergeordneten Testmanager außerhalb des Teams geben, der als Bezugsperson für die Tester in mehreren Teams dient; aber dieser darf keinen Einfluss auf die Arbeit innerhalb des Teams haben und hat allenfalls eine beratende Funktion. Die bisherige Planung, Organisation und Steuerung eines separaten Testteams unter der Leitung eines Teammanagers sind nicht mehr notwendig. Sie passen nicht zur agilen Philosophie der Teamarbeit.

Bei den Testmethoden werden die Methoden hervorgehoben, die sich am besten für den agilen Ansatz eignen: risikobasiertes Testen, wertgetriebenes Testen, exploratives Testen, sitzungsbasiertes Testen und abnahmetestgetriebene Entwicklung. Konventionelle Testtechniken wie Äquivalenzklassenbildung, Grenzwertanalyse, Zustandsanalyse und Entscheidungstabellen oder -bäume sind nach wie vor wichtig und werden durch agile Testmethoden nicht abgelöst, sondern in diesen angewendet. Die Bedeutung der Wiederverwendung von Tests und der Wiederholung von Tests wird betont. Alle Techniken müssen diese Kriterien erfüllen. Der Integrationstest ist eine fortlaufende Geschichte und der Abnahmetest wird immer wieder wiederholt. Die zyklische Natur eines agilen Projekts führt zu einer Neudefinition der Testendekriterien. Der Test endet nie – solange das Produkt weiterwächst.

In Kapitel 6, „Agile Testdokumentation“, beschreiben die Autoren, welche Dokumente die Tester in einem agilen Projekt erstellen müssen, sollen oder können. Dazu gehören eine testbare Anforderungsspezifikation aus den User-Storys, ein Testdesign, eine Benutzerdokumentation und Testberichte. Die Testfälle gelten nicht als Dokumentation, sondern als Testware. Ein Anliegen der agilen Entwicklung ist es, die Dokumentation auf ein Minimum zu reduzieren. In der Vergangenheit wurde es mit der Dokumentation tatsächlich übertrieben. In einem agilen Entwicklungsprojekt wird nur noch das dokumentiert, was unbedingt notwendig ist. Ob eine Teststrategie oder ein Testdesign notwendig ist, sei dahingestellt. Testfälle sind unerlässlich, aber sie sind ebenso Teil des Softwareprodukts wie der Code. Daher werden sie nicht als Dokumentation betrachtet.

Das wichtigste Dokument ist die Anforderungsspezifikation, die sich aus den User-Storys ergibt. Sie dient als Grundlage für den Test, das sogenannte Testorakel. Aus ihr werden die Testfälle abgeleitet und gegen sie wird getestet. Sie enthält auch die Abnahmekriterien. Die einzigen Testberichte, die wirklich benötigt werden, sind der Testüberdeckungsbericht und

der Fehlerbericht. Der Testüberdeckungsbericht zeigt, was getestet wurde und was nicht. Die Tester benötigen dieses Dokument als Nachweis, dass die Tests ausreichend durchgeführt wurden. Der Benutzer braucht ihn, um Vertrauen in das Produkt zu gewinnen. Im Fehlerbericht wird festgehalten, welche Abweichungen aufgetreten sind und wie sie behandelt werden. Diese beiden Berichte sind die besten Indikatoren für den Stand des Tests.

Tester wären prädestiniert, das Benutzerhandbuch zu schreiben, weil sie das System am besten kennen und wissen, wie man es benutzt. Jemand muss das Handbuch schreiben und die Tester sind die richtigen Kandidaten dafür. Sie sorgen dafür, dass dieses Dokument nach jeder Iteration oder jedem Release aktualisiert wird. Ansonsten folgt das Buch dem agilen Prinzip, die Dokumentation auf das Wesentliche zu beschränken. Der wichtigste Aspekt ist, dass es immer eine solide Anforderungsspezifikation und eine verständliche Benutzerdokumentation gibt. Eine strukturierte, semiformale Anforderungsspezifikation bildet die Basis für den Test und kein Anwender möchte auf eine Benutzeranleitung verzichten.

Kapitel 7 beschäftigt sich mit dem wichtigen Thema „Testautomatisierung“. Testautomatisierung ist in der agilen Entwicklung besonders wichtig, da sie das wichtigste Werkzeug zur Projektbeschleunigung darstellt und zur Unterstützung moderner DevOps-Ansätze unerlässlich ist. Nur durch Automatisierung kann der Testaufwand auf ein akzeptables Maß reduziert werden, während bei kontinuierlicher Integration (Continuous Integration) die Produktqualität erhalten bleibt. Die Autoren unterscheiden dabei zwischen Unittest, Komponenten-Integrationstest und Systemtest. Der Unittest wird am Beispiel von JUnit detailliert dargestellt. Es wird gezeigt, wie Entwickler testgetrieben arbeiten müssen, wie man Testfälle aufbaut und wie man die Testüberdeckung misst. Der Komponenten-Integrationstest wird am Beispiel des Apache Maven Integrationsservers erläutert. Dabei ist es wichtig, die Schnittstellen der integrierten Komponenten zu den noch nicht vorhandenen Komponenten über Platzhalter zu simulieren. Der Systemtest wird durch einen fachlichen Test mit FitNesse beschrieben. Das Entscheidende dabei ist die Umsetzung der Testfälle in Testskripte, die beliebig erweitert und wiederholt werden können. Die Autoren betonen auch, wie wichtig es ist, die Testware – Testfälle, Testskripte, Testdaten etc. – bequem und sicher verwalten zu können, damit der Test möglichst reibungslos abläuft. Auch dafür werden Werkzeuge benötigt.

Kapitel 8 fügt Beispiele aus der Testwerkzeugpraxis hinzu und erweitert die Testautomatisierung um Testmanagementfunktionalität. Zunächst wird das Werkzeug Broadcom Rally beschrieben, das den agilen Lebenszyklus von der Verwaltung der Storys bis zum Fehlermanagement unterstützt. Der agile Tester kann dieses Werkzeug zur Planung und Steuerung seiner Tests verwenden. Eine Alternative zu Broadcom Rally ist Polarion QA/ALM, das sich besonders für die Erfassung und Priorisierung von Testfällen und für die Fehlerverfolgung eignet. Weitere Testplanungs- und Tracking-Tools sind die Tools Pachno, das insbesondere die Testaufwandsschätzung unterstützt, Atlassian JIRA, das eine umfangreiche Fehleranalyse bietet, und Azure Test Plans.

Für Tester in einem agilen Projekt ist der kontinuierliche Integrationstest entscheidend. Er muss die letzten Komponenten so schnell wie möglich mit den Komponenten des letzten Release integrieren und bestätigen, dass sie reibungslos zusammenarbeiten. Dazu muss er die Tests nicht nur über die Benutzeroberfläche, sondern auch über die internen Systemschnittstellen durchführen. Mit Tricentis Tosca können sowohl externe als auch interne Schnittstellen generiert, aktualisiert und validiert werden.

Die Autoren beschreiben anhand ihrer eigenen Projekterfahrungen, wie diese Werkzeuge eingesetzt werden und wo ihre Grenzen liegen.

Das neunte Kapitel des Buchs ist dem Thema „Ausbildung und ihre Bedeutung“ gewidmet. Die Autoren betonen die Rolle der Mitarbeiterschulung für den Einstieg in die agile Entwicklung. Schulungen sind für den Erfolg bei der Anwendung der neuen Methoden unerlässlich und dies gilt insbesondere für die Tester. Die Tester in einem agilen Team müssen genau wissen, worauf sie sich konzentrieren müssen, und das können sie nur durch entsprechende Schulungen lernen. Es gibt viele Interpretationen von agilen Ansätzen, dennoch muss die Qualität des Produkts sichergestellt werden – und dazu braucht es professionelle Tester, die für die Arbeit in einem agilen Team ausgebildet sind. Ausbildungsprogramme, die auf die Bedürfnisse des agilen Testens ausgerichtet sind, werden in diesem Kapitel diskutiert.

Das zehnte Kapitel „Retrospektive“ blickt noch einmal auf die Gründe und die Motivation hinter der agilen Revolution zurück und weist nachdrücklich darauf hin, dass die agile Transformation ein Veränderungsprozess ist, der nicht nur einzelne Projektteams, sondern die gesamte Organisation betrifft.

Zusammenfassend deckt dieses Buch die wesentlichen Aspekte des agilen Testens ab und bietet einen wertvollen Leitfaden für das Testen in einer agilen Umgebung. Der Leser erhält viele Anregungen, wie er in agilen Projekten vorgehen kann. Er lernt, wie man das agile Testen vorbereitet und durchführt. Als Buch, das von Testpraktikern geschrieben wurde, hilft es Testern, sich in der oft verwirrenden agilen Welt zurechtzufinden. Es gibt ihnen klare, fundierte Anleitungen für die Umsetzung agiler Prinzipien in der Testpraxis. Es gehört damit in die Bibliothek jeder Organisation, die agile Projekte durchführt.

Harry M. Sneed

Vorwort

Im Jahr 2013 erschien die erste Auflage von „Agile Testing – Der agile Weg zur Qualität“, 2018 folgte die zweite Auflage. Jetzt, im Jahr 2024, also sechs Jahre später, freut es uns, die bereits dritte Auflage präsentieren zu können. Es überrascht nicht, dass auch in diesem Zeitraum unglaublich rasante Entwicklungen im Bereich der Softwareentwicklung stattgefunden haben. Agilität ist nicht mehr nur ein Thema für einzelne Entwicklungsprojekte und -teams, sondern hat die Unternehmen als Ganzes erfasst. Das Mindset in der Softwareentwicklung, insbesondere in Bezug auf das Testen und die Rolle der Tester, hat sich – aus unserer Sicht – sehr positiv verändert. Den Herausforderungen großer, skalierender Projekte und Programme wird mit entsprechenden Modellen begegnet. Automatisierung in allen Phasen des Softwarelebenszyklus und DevOps sind integrale Bestandteile vieler Vorgehensmodelle in den Unternehmen. Und es gibt auch noch eine Vielzahl von Unternehmen, die eher in traditionellen oder überwiegend hybriden Ansätzen arbeiten und damit erfolgreich sind. Auch für sie bietet dieses Buch Ansätze für die agile Transformation der Softwareentwicklung im Hinblick auf die Qualitätssicherung.

Deshalb haben wir in dieser Ausgabe von Agile Testing die Aspekte Mindset, Skalierung und DevOps erweitert und wollen auch betonen, dass agiles Testen keineswegs eine Abkehr von altbewährten Testtechniken bedeutet, sondern vielmehr eine Anwendung dieser Handwerkszeuge in einem veränderten Vorgehen.

Als das „Manifest für agile Softwareentwicklung“ im Jahr 2001 von einer Gruppe von Softwareingenieuren in Utah/USA unterzeichnet wurde, leitete es den wohl bedeutendsten Wandel in der Softwareentwicklung seit der Einführung der Objektorientierung Mitte der 1980er-Jahre ein. Das „Agile Manifest“, quasi die Zehn Gebote der agilen Welt, kann durchaus als Ausdruck einer Gegenbewegung zu den sich ab Ende der 1980er-Jahre verbreitenden, stark regulierenden Prozess- und Planungsmodellen wie PRINCE, dem V-Modell oder auch der ISO9001 gesehen werden. Diese Modelle versuchten, den bis dahin chaotischen und willkürlichen Entwicklungsprozessen durch Planung, Strukturierung der Prozesse und Dokumentation entgegenzuwirken. Das Agile Manifest positioniert sich bewusst zu diesen Aspekten und gibt seinen zentralen vier agilen Werten – Interaktion, Zusammenarbeit mit dem Kunden, Reagieren auf Veränderungen und schließlich funktionierende Software – eine höhere Relevanz für eine erfolgreiche Softwareentwicklung.

Die Art und Weise, wie das Agile Manifest in Werten und Prinzipien formuliert ist, war ein Grund dafür, dass der Siegeszug der agilen Softwareentwicklung in den Jahren seit der Veröffentlichung des Agilen Manifests von vielen Glaubenssätzen geprägt war. Wir, die Autoren dieses Buchs, erlebten dies nicht zum ersten Mal. In den Jahrzehnten unserer Be-

rufserfahrung sind wir immer wieder mit neuen Lösungen für das „Softwareproblem“ konfrontiert worden: strukturierte Programmierung, objektorientierte Programmierung, CASE (Computerunterstützte Softwareentwicklung), RUP (Rational Unified Process), V-Modell, ISO9001, SOA (Service-Orientierte Architektur) … – eine lange Liste sogenannter „Silver Bullets“, die die Probleme lösen sollten. Allerdings wuchsen die Herausforderungen und ihre Komplexität schneller, als sich neue Ansätze etablieren konnten. So können wir uns bereits jetzt auf künftige Ansätze freuen, die beispielsweise auf den Konzepten der künstlichen Intelligenz und maschinellem Lernen basieren.

Auf dem Weg dieser ständigen Weiterentwicklung ist man jedoch gut beraten, nicht immer das Kind mit dem Bade auszuschütten. Vor zehn Jahren war eine Motivation für dieses Buch, dass einige Unternehmen der Meinung waren, dass Tester in agilen Projekten nicht mehr benötigt würden, weil das Testen nun von den Programmierern und dem Product-Owner erledigt würde. Sie trennten sich sogar von Testern und mussten später schmerzlich zur Kenntnis nehmen, dass sich die Programmierer nicht dem detaillierten funktionalen oder auch nicht-funktionalen Test der Storys und deren Integration auf Applikationsebene widmen konnten oder wollten und auch die Product-Owner und Anwender eher die Validierung der Epics und End-to-End-Tests im Auge hatten als eine Verifikation der Applikation im Sinne eines Systemtests.

Wir, die Autoren, waren schon immer unglücklich mit der dogmatischen Umsetzung neuer Ansätze in der Softwareentwicklung. Im Gegensatz dazu sehen wir die Veränderungen als Chance für einen Prozess der kontinuierlichen Verbesserung und Optimierung.

In den letzten Jahren wurden wir Autoren in agilen Projekten jedoch auch immer wieder damit konfrontiert, dass vieles, was wir als Tester an Selbstverständnis, Methoden, Techniken und Standards erarbeitet und gelernt haben, nicht mehr gelten soll. Das mag auch daran liegen, dass die agile Community in der Vergangenheit vor allem von Softwareentwicklern geprägt war. Diese Tatsache ist auch einer der Gründe, warum die Aufgaben und die Rolle des Softwaretesters in agilen Methoden und Projekten oft nicht oder nur sehr vage definiert sind. Dazu tragen auch unterschiedlich interpretierte Terminologien bei: Wenn z. B. bei Scrum von einem interdisziplinären Entwicklungsteam die Rede ist, dachten oder denken manche, dass das Team nur aus Entwicklern (im Sinne von Programmierern) besteht, die für alles verantwortlich sind, und dass durch testgetriebene Entwicklung mit der Implementierung eines automatisierten Unittest-Sets die Testaufgaben in der Entwicklung ausreichend erfüllt werden. Der Rest liegt in der Verantwortung des Anwenders im User Acceptance Test. Wo sind also die bekannten Testphasen und Teststufen, insbesondere der System- und Systemintegrationstest? Wo und wie finden wir uns als Tester in agilen Projekten wieder? Der agile Ansatz warf und wirft für uns Tester offensichtlich mehr Fragen auf, als er Antworten auf bisherige Probleme lieferte.

Lösungsansätze für diese Probleme wollen wir mit unserem Buch präsentieren, das von Testern für Tester geschrieben wurde. In den einzelnen Kapiteln geben wir Antworten auf die zentralen Fragen, denen wir in unseren Projekten immer wieder begegnet sind. Es geht um generelle und gleichsam kulturelle Veränderungsprozesse, um Fragen der Vorgehensweise und Organisation im Softwaretest, um den Einsatz von Methoden, Techniken und Werkzeugen, insbesondere der Testautomatisierung, und um die neu definierte Rolle des Testers in agilen Projekten. Ein breites Spektrum, das im Rahmen dieses Buchs sicher nicht abschließend und umfassend behandelt werden kann, von dem wir uns aber dennoch erhoffen, es mit Ideen und Anregungen für den Leser abzudecken.

Um die beschriebenen Aspekte noch greifbarer zu machen, werden die einzelnen Themen des Buchs von Erfahrungsberichten aus konkreten Softwareentwicklungsprojekten verschiedener Unternehmen begleitet.

Die Beispiele sollen zeigen, dass unterschiedliche Ansätze zu guten Lösungen führen können, die den spezifischen Herausforderungen agiler Projekte gerecht werden.

In diesem Sinne wünschen wir der Leserin bzw. dem Leser viel Erfolg bei der Umsetzung der hier vorgestellten Inhalte in ihren bzw. seinen eigenen Projekten und laden sie/ihn gleichzeitig ein, uns, die Autoren, auf unserer Internetplattform *www.agile-testing.eu* zu besuchen.

Manfred Baumgartner, Wien 2024

Martin Klonk, St. Pölten 2024

Christian Mastnak, Wien 2024

Richard Seidl, Essen 2024

Die Praxisbeispiele im Buch

Das Praxisbeispiel für EMIL in diesem Buch stammt von einem Unternehmen aus der Gesundheitsbranche, das auf 25 Jahre erfolgreiche Produkt- und Softwareentwicklung zurückblicken kann. Mit dem Wachstum des Unternehmens, neuen Kundenanforderungen und strengeren regulatorischen Vorgaben wuchs auch die Notwendigkeit, die Entwicklungs- und Testprozesse zu optimieren und effizienter zu gestalten. Der Gedanke, vom traditionellen auf den agilen Entwicklungsprozess umzusteigen, kam bereits hier und da im Unternehmen auf. Das Softwareentwicklungsprojekt EMIL war der Ausgangspunkt für diese Initiative. Ziel des Projekts war die Neuimplementierung einer Analysesoftware, die seit zehn Jahren weltweit erfolgreich eingesetzt und von verschiedenen Entwicklern entwickelt worden war. Insbesondere aus technischer und architektonischer Sicht konnten die neuen Anforderungen nicht mehr problemlos umgesetzt werden; viele Funktionen waren im Laufe der Jahre als „provisorische Anbauten“ hinzugekommen, aber nie entfernt oder integriert worden. Ein grober Zeitrahmen für die Neuimplementierung aller Funktionen der bestehenden Software wurde auf etwa zweieinhalb Jahre geschätzt. Die größten Herausforderungen auf dem Weg zur agilen Entwicklung waren die mangelnde Erfahrung mit der Zieltechnologie und die regulatorischen Anforderungen, die das Gesundheitswesen mit sich bringt. Die positiven und negativen Erfahrungen, die aufgetretenen Probleme und die Lösungsversuche aus den ersten eineinhalb Jahren des Projekts finden sich in diesem Buch wieder und sind in den entsprechenden Kapiteln entsprechend gekennzeichnet.

Ein weiteres Praxisbeispiel liefert OTTO. Als Online-Händler agiert OTTO in einem sehr agilen Marktumfeld und nutzt innovative Technologien, um auf otto.de und in seinen Filialen ein positives Einkaufserlebnis zu bieten. Als Teil der Otto Group ist OTTO eines der erfolgreichsten E-Commerce-Unternehmen in Europa und Deutschlands größter Online-Händler für Mode und Lifestyle im B2C-Bereich. Über 90 Prozent des Gesamtumsatzes werden online generiert. In den Praxisbeispielen berichtet Frau Diana Kruse von ihren Erfahrungen auf dem Weg von der Testerin und Testmanagerin zum Quality Specialist bzw. Quality Coach, visualisiert durch Grafiken ihres Kollegen Torsten Mangner.

Die Autoren

Manfred Baumgartner

Manfred Baumgartner verfügt über mehr als 30 Jahre Erfahrung in der Softwareentwicklung, insbesondere in der Softwarequalitätssicherung und im Softwaretest. Nach dem Studium der Informatik an der Technischen Universität Wien war er als Softwareentwickler bei einem großen Softwareunternehmen im Bankensektor und später als Quality Director eines CRM-Lösungsanbieters tätig. Seit 2001 hat er die QS-Beratungs- und Schulungsangebote der ANECON, später Nagarro GmbH, eines der führenden Dienstleistungsunternehmen im Bereich Softwaretest, auf- und ausgebaut. Er ist Vorstandsmitglied im Arbeitskreis für Softwarequalität und Fortbildung (ASQF) und Mitglied des Austrian Testing Board (ATB). Seine umfangreichen Erfahrungen sowohl in der klassischen als auch in der agilen Softwareentwicklung bringt er als beliebter Referent auf international renommierten Konferenzen und als Autor und Co-Autor einschlägiger Fachbücher ein: „Der Systemtest – Von den Anforderungen zum Qualitätsnachweis“ (2006, 2008, 2011), „Software in Zahlen“, (2010), „Basiswissen Testautomatisierung“ (2012, 2015, 2021), „Agile Testing – Der agile Weg zur Qualität“ (2013, 2018, 2023).

Martin Klonk

Martin Klonk ist Senior-Testexperte bei Sixsentix Austria GmbH. Ausgebildet als Wirtschaftsingenieur an der Technischen Universität Berlin (und der Université Libre de Bruxelles), begann er seine Karriere 1996 als Softwaretestspezialist bei SQS Software Quality Systems (heute Expleo) in Köln und München. Später wechselte er zu SQS, ANECON und Nagarro Austria in Wien. Martin Klonk hat in den unterschiedlichsten Branchen gearbeitet und war in fast allen Bereichen des Softwaretestens aktiv. Als Präsident des Austrian Testing Board des ISTQB arbeitet er regelmäßig an Lehrplänen und deren deutscher Übersetzung mit und führt auch selbst Schulungen durch. Seit ihm 2007 die Umsetzung erfolgreicher Teststrategien in einem agilen Projekt gelungen ist, ist Martin Klonk ein starker

Verfechter agiler Praktiken im Testen und hat mehrere agile Projekte als Testspezialist geleitet. Er ist zertifizierter Agiler Tester, Projektmanager und Scrum Master.

Christian Mastnak

Christian Mastnak arbeitet als Principal Software Testing Consultant bei Nagarro und verfügt über mehr als 15 Jahre Erfahrung im Bereich QA. Er leitet Nagarros globale „Agile Testing Practice“ und implementiert innovative Lösungen für internationale Kunden in verschiedenen Branchen. Da er in all seinen Projekten einen sehr praxisorientierten Ansatz verfolgt, wechselt er gerne zwischen verschiedenen QA-Rollen – vom Testmanager zum Agile Quality Coach, vom Testautomatisierungsarchitekten zum Testberater. Diese Rollen ermöglichen es ihm, seiner Leidenschaft für die kontinuierliche Verbesserung von QS-Prozessen und -Methoden nachzugehen. Christian Mastnak gibt sein Fachwissen als gefragter Trainer und Redner auf internationalen Konferenzen weiter, darunter EuroSTAR, Agile Testing und die Software Quality Days.

Richard Seidl

Richard Seidl ist Agile Quality Coach und Softwaretestexperte. In seiner abwechslungsreichen beruflichen Laufbahn hat er schon viel Software gesehen und getestet: gute und schlechte, große, kleine, alte und neue. Seine Erfahrungen bündelt er nun zu einem ganzheitlichen Ansatz, denn Entwicklungs- und Testprozesse können nur dann erfolgreich sein, wenn die unterschiedlichsten Kräfte sowie Stärken und Schwächen ausbalanciert sind. So wie ein Ökosystem nur mit allen Aspekten in seiner ganzen Qualität harmonisch existieren kann, müssen die Prozesse im Testumfeld als ein Netzwerk verschiedener Akteure betrachtet werden. Agilität und Qualität wird dann zu einer Haltung, die wir wirklich leben können, anstatt sie nur abzuarbeiten. Als Autor und Co-Autor hat er verschiedene Fachbücher und Artikel veröffentlicht, darunter „Der Systemtest – Von den Anforderungen zum Qualitätsnachweis“ (2006, 2008, 2011), „Der Integrationstest – Von Entwurf und Architektur zur Komponenten- und Systemintegration“ (2012) und „Basiswissen Testautomatisierung“ (2012, 2015, 2021).

Danksagungen

Wir danken den Firmen Nagarro GmbH, GETEMED Medizin- und Informationstechnik AG und Otto (GmbH & Co KG) für ihren Beitrag zur Arbeit an diesem Buch.

Wir danken auch unseren Kolleginnen und Kollegen für deren eifrige Unterstützung und unseren Reviewern, die uns mit kritischen Anmerkungen geerdet und einen wertvollen Beitrag zu diesem Buch geleistet haben:

Sonja Baumgartner (Grafik), Stefan Gwihs, Diana Kruse & Torsten Manger (Praxisbeispiele und Grafik Otto.de), Anett Prochnow, Petra Scherzer, Michael Schlimbach, Silvia Seidl und Harry Sneed. Unser herzlicher Dank gilt auch unseren Lektorinnen Brigitte Bauer-Schiewek, Petra Kienle sowie Kristin Rothe, die unzählige Fehler im Lektorat behoben haben, welche wir übersehen hatten.

Ein besonders großes Dankeschön und eine hohe Wertschätzung gehen an unsere Mitautoren der ersten und zweiten Auflage, Helmut Pichler und Siegfried Tanczos, die uns auch bei der Überarbeitung der vorliegenden Auflage mit ihrer Erfahrung unterstützt haben.