

# Agiles Testen – was ist das?

Christian Brandes



Christian Brandes

„Agiles Testen“ – die Formulierung tauchte und taucht in zahllosen Publikationen und Vorträgen auf. Sie ist inzwischen so vertraut, dass man auf die Frage „Was heißt das eigentlich?“ vermutlich mit leichter Verwirrung reagieren wird. Aber trotzdem: Wie würden Sie „agiles Testen“ definieren? Und wie würden Sie es in einem Software-Projekt aufsetzen?

Wenn man sich mit diesen Fragen auseinandersetzt, stellt man fest, dass „agiles Testen“ letztlich bedeutet, vorhandene und [bewährte Testtechniken so im agilen Entwicklungsprozess zu verankern, dass sie die Ziele des agilen Vorgehens unterstützen](#), nämlich:

- schnelles Feedback
- hoher Automatisierungsgrad
- geringer Management-Overhead (etwa für Testplanung oder Fehlermanagement)
- enge Zusammenarbeit im Team.

Wie ein Mapping agiler Werte auf das Testen aussehen kann, ist in [4] zu finden. In dieses Bild passen auch die beiden „typisch agilen“ Testtechniken „exploratives Testen“ und „testgetriebene Entwicklung“, die aber nicht nur in agilen Projekten einsetzbar sind und ihrerseits von bewährten Testmethoden massiv profitieren können.

Agiles Testen bedeutet also keineswegs, dass nur noch ad-hoc oder informell getestet wird. Dass es in der IT-Branche nach wie vor Unterstützungs- und Erklärungsbedarf beim Aufsetzen von agilem Testen gibt, ist an verschiedenen Stellen ersichtlich. So legen die Besucherzahlen einer kürzlich zum Thema durchgeführten Roadshow [5] nahe, dass auch langjährige Scrum-Praktizierer definitiv noch nach praxisverwertbaren Informationen zum Thema „Test & QS in Scrum“ suchen.

Das oben genannte Ziel – ein Testsetup, das agile Werte unterstützt – lässt sich durch verschiedene Maßnahmen in einem Scrum-Prozess erreichen, denn Scrum selbst fördert und fordert das Prinzip einer „built-in quality from the beginning“. Dabei sollte der Begriff des „Sprints“ nicht mit „hoher Geschwindigkeit“ übersetzt werden, sondern mit „kurzer Distanz“. Wie Nancy van Schooenderwoert in [6] korrekt betonte: „If you shoot for quality, speed will come as a side effect. If you shoot for speed, quality will NOT come as side effect!“

Die erste und wichtigste Maßnahme behandelt das „weiche“ Thema [Teambildung](#), denn es ist kriegsentscheidend für alle Folgethemen. Viel zu oft treffen wir an, dass Tester eben nicht ins Team integriert sind: Sie nehmen nicht an Scrum-Meetings teil, sie sind vielleicht sogar räumlich vom Team getrennt, oder sie stoßen erst in einer späteren Projektphase dazu. Dies alles sind Fehler, zu deren Behebung alle Beteiligten – als Hol- und Bringschuld auch die Tester – aktiv beitragen müssen. Akzeptanz finden Tester in einem agilen Setup erfahrungsgemäß dann, wenn sie ihr

Ist agiles Testen ein „neues“ oder „anderes“ Testen als das in „klassischen“ nicht-agilen Vorgehensmodellen? Oder die Nutzung bekannter Testtechniken in einem neuen oder geänderten Kontext und Setup? Im ISTQB-Glossar [1] wird agiles Testen so erklärt: „Testing practice for a project using agile methodologies (...). See also test driven development.“ Selbst in Lisa Crispins Buch gleichen Titels [2] ist keine geschlossene Begriffsdefinition zu finden, sondern vor allem die Aussage: „When we say ‚agile testing‘ in this book, we’re usually talking about business-facing tests.“ Verschiedene andere Antworten gehen in die Richtung: „Tests früher durchführen und häufiger wiederholen“ – sicherlich nicht falsch, aber als Definition unzureichend.

Das ist insofern bedauerlich, als aus Sicht eines ISTQB-zertifizierten Testers ein Blick auf ein agiles Framework wie Scrum durchaus spannende Fragen aufwirft:

- Wo ist der Testmanager geblieben? Ist nur die Rolle verschwunden oder auch die zugehörigen Aufgaben?
- Wie steht es mit Testkonzept, Teststrategie und Testplan? Benötigt man all das beim agilen Testen nicht mehr?
- Welche Skills benötigen Tester in einem „cross-funktionalen“ Team? Muss etwa ein Fachtester programmieren lernen, um in einem Scrum-Team mitarbeiten zu können?
- Darf man denn – im Gegensatz zu den Forderungen der Original-Scrum-Literatur [3], die dies ausdrücklich untersagt – überhaupt noch von den Rollen „Tester“ und „Entwickler“ reden?
- Sind User Stories eine ausreichende Testbasis? Wenn ja, für welche Tests?
- Wo sind die Teststufen des allgemeinen V-Modells geblieben? Reichen wirklich Unittests und Akzeptanztests aus, wie oft zu lesen?

Mindset anpassen, d.h. den Entwickler-Teamkollegen als Kunden betrachten, der vor allem ein schnelles Feedback erwartet. Wenn dieses Ziel nicht erreicht wird, wird man als Tester in einem agilen Team immer einen schweren Stand haben. Beide Seiten müssen zu offener Zusammenarbeit und Kommunikation bereit sein, was erfahrungsgemäß keine kleine Hürde ist.



Abbildung 1: Pairing von Tester und Entwickler in einem Scrum-Projekt

Die zweite Maßnahme betrachtet das Quality Gate für Anforderungen, die „Definition of Ready“. Good Practices hierfür wie das [INVEST-Prinzip](#) [7] fordern zwar prinzipiell „Testbarkeit“, aber die Praxis sieht häufig leider anders aus: Entweder gibt es überhaupt keine „Definition of Ready“, oder aber es entstehen Akzeptanzkriterien, die bei genauerem Hinsehen reine Anforderungen sind, deren Prüfung komplett unspezifiziert ist. Dabei sollten Akzeptanzkriterien vielmehr ausdrücken, was der Product Owner konkret [sehen](#) will, um davon überzeugt zu sein, dass eine User Story erfolgreich realisiert wurde. Hilfreich zur Findung guter Akzeptanzkriterien („examples“) ist folgendes einfache Gedankenexperiment: „Ihnen wird gesagt, dass ein Feature fertig sei, und Sie haben 10 Minuten, um über eine Abnahme zu entscheiden – was würden Sie in diesen 10 Minuten sehen wollen?“ Um frühzeitig die Testbarkeit von Anforderungen zu gewährleisten, kann ein Pairing zwischen Product Owner und Tester schon bei der Entstehung der User Stories Wunder wirken – übrigens eine exzellente Einsatzmöglichkeit für Fachtester ohne Programmierkenntnisse.

Der dritte Erfolgs-Baustein ist die „[Definition of Done](#)“. Aussagen darin wie z. B. „Die Unittests müssen erfolgreich gelaufen sein“ sagen leider nichts über die Qualität und Inhalte der Unittests aus. Das Fehlen einer [Teststrategie](#) und abgestimmter Testaktivitäten mit durchdachten Testzielen kann auch in agilen Projekten zu ernststen Problemen führen. In der Definition of Done – oder einer separaten schlanken „Definition of Test“ (ein ähnliches Konstrukt ist in [8] zu finden) – kann unter Zuhilfenahme der agilen Testquadranten [1] definiert werden, was pro Iteration wie

getestet werden soll. Und zwar nicht erst in einem letzten Test-Sprint oder jeweils am Sprintende, sondern vom ersten Sprinttag an als permanente Aktivität. Wir bezeichnen diese Einbettung des Tests in den Entwicklungsprozess als „Test Nonstop“. Eine kompakt formulierte Teststrategie in einem agilen Projekt darf den Wert des „so viel dokumentieren wie nötig (lies: nutzbringend) und so wenig wie möglich“ nicht verletzen und wird nach jeder Iteration – etwa in der Sprint-Retrospektive – kritisch auf den Prüfstand gestellt und verbessert („inspect and adapt“, natürlich auch für das Testvorgehen).

Apropos Teststrategie: Von Anfang an sollte begleitend ein [Systemtest](#) aufgesetzt werden. Scrum verlangt, dass nach jedem Sprint ein potenziell lieferbares – insbesondere also vorführbares – Softwareprodukt entstehen soll, das folglich auch von Anfang an und „nonstop“ einem Systemtest aus Nutzerperspektive methodisch und risikobasiert unterzogen werden kann und sollte. Unittests und Akzeptanztests alleine genügen in der Regel nicht, um die nötige Qualität sicherzustellen! Denn: Akzeptanztests haben ein ganz anderes Testziel – Überprüfung der Akzeptanzkriterien – als ein Systemtest, der gemeinsam im Mix mit Unittests und Integrationstests dem Ziel „Support the Team“ dienen soll.

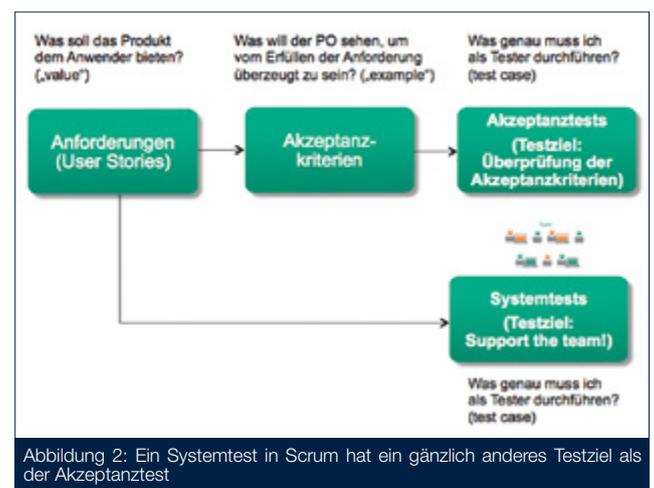


Abbildung 2: Ein Systemtest in Scrum hat ein gänzlich anderes Testziel als der Akzeptanztest

Hand in Hand damit geht eine [reife und gut modularisierte Testautomatisierung](#) auch der Systemtests. Die Testautomatisierung muss vor allem mit der Sprint-Taktung von 2–4 Wochen Schritt halten können. Im Idealfall lösen sich dann die Teststufen aus dem allgemeinen V-Modell in einem täglichen Zyklus auf, in dem sie als einzelne Testaktivitäten zu einer User Story vom ersten Sprinttag an durchgeführt werden. Die automatisierten Systemtests sind dann Bestandteil der Continuous Integration. Sogar ein „Test First“-Ansatz für Systemtests kann erfolgreich praktiziert werden.



Zusammengefasst steckt aus Testsicht in agilen Modellen wie Scrum eine **Win-Win-Situation** und damit eine riesige Chance: Agilität betont die Bedeutung des Testens z. B. durch Ansätze wie Test First. Umgekehrt können professionelle und bewährte Testmethoden helfen, die agilen Projektziele zu erreichen.

Derzeit stehen in der agilen Community Themen rund um „Enterprise Agile“ zur Klärung an, die über einzelne IT-Projekte hinausgehen: Wie etwa vertragen sich Unternehmensziele wie unternehmensweit einheitliche Werkzeugketten mit agilen Werten? Passen agile Teams und ein zentrales Testcenter zusammen? Ist ein Mischbetrieb zwischen agilen und nichtagilen Teams möglich? Vertragen sich schlanke Dokumentation und Modellierung, konkret etwa Agilität und modellbasiertes Testen? Wie können schlanke Testdokumentation und regulative Anforderungen unter einen Hut gebracht werden? Zu all diesen Themen bieten wir Antworten, die aber den Rahmen dieses Beitrags sprengen würden.

Festzuhalten ist: Auch Agilität ist kein Allheilmittel, und nicht jedes Projekt eignet sich gleich gut für den Einsatz agiler Methoden. Wenn agil entwickelt wird, muss auch

das Testen agil aufgesetzt werden. Dies bedeutet, dass sämtliche Testaufgaben als „Test nonstop“ nahtlos in den agilen Prozess eingebettet werden müssen. Die klassischen Testtechniken werden dabei weiterhin benötigt. Dies meinen wir mit „Change Your Mindset – Keep Your Methodset“: Agil zusammenarbeiten, aber weiterhin professionell testen.

#### Literatur:

- [1] ISTQB® Testing Glossary, <http://istqb.org/downloads/finish/20/14.html>
- [2] Lisa Crispin, Janet Gregory: „Agile Testing“, Addison-Wesley 2008
- [3] Ken Schwaber, Jeff Sutherland, „Scrum Guide“, [http://www.scrum.org/storage/scrum-guides/Scrum\\_Guide%20-%20DE.pdf](http://www.scrum.org/storage/scrum-guides/Scrum_Guide%20-%20DE.pdf)
- [4] <http://www.agiletestingmanifesto.de>
- [5] <http://www.imbus.de/veranstaltungen/trends-in-testing-2012/>
- [6] <http://www.qs-tag.de/software-qs-tag-archiv/2011/abstracts/abstract-vanschooenderwoert/#c12425>
- [7] Roman Pichler, „Scrum“, dpunkt 2007
- [8] <http://www.informit.com/articles/article.aspx?p=415981>

#### Der Autor

**Dr. Christian Brandes** arbeitet als Principal Consultant und Trainer für die imbus AG. Er ist ISTQB Certified Tester (Full Advanced Level) sowie Certified Professional Scrum Master und beschäftigt sich neben Tests in agilen Projekten mit Testautomatisierung, Testdesign und modellbasiertem Testen.