



□ Dominique Mühlbauer

(dominique.muehlbauer@sogeti.de)

arbeitet als Berater für Softwarequalität und Softwaretest in unterschiedlichsten Branchen. Sein Schwerpunkt liegt auf der Integration von Qualitätssicherungsmaßnahmen in agile Vorgehens- und Projektmodelle sowie deren Anwendung in DevOps-Organisationen.

▶ <http://www.scrumguides.org/>

▶ <http://www.extremeprogramming.org/>

Das Q in Agile Qualitätssicherung in agilen Projekten

Agile Softwareentwicklungsmethoden finden immer weitere Verbreitung. Allerdings fehlt es häufig an einem stringenten Vorgehen zur Qualitätssicherung. In einem Umfeld, in dem ein unzufriedener Kunde nur einen Klick weiter eine Alternative findet, ist hohe Qualität überlebensnotwendig. Dieser Artikel beschäftigt sich mit der Frage, was zu einem ganzheitlichen Qualitätssicherungsansatz gehört und wie dieser in einen agilen Rahmen eingepasst werden kann.

Das Q in Agile

Agile Methoden versprechen, durch direktere Kommunikation und einen kürzeren Planungshorizont, höhere Flexibilität und Kundenzufriedenheit sowie eine deutliche Verkürzung der Time-to-Market-Zeit gegenüber traditionellen Projektmodellen zur Softwareentwicklung. Wenig verwunderlich finden sie in den letzten Jahren in der IT-Industrie zunehmend Verbreitung. Dies belegt beispielsweise die Studie „Status Quo Agile - Verbreitung und Nutzung agiler Methoden“ [Kom14], welche, einschließlich wahlweiser Nutzung von Mischformen, einen Einsatz von agilen Methoden in 85 Prozent der befragten Unternehmen ausweist (siehe [Abbildung 1](#)).

Parallel zur Verbreitung von agilen Methoden steigerte sich der Stellenwert der Softwarequalitätssicherung. Ein gewichtiger Grund hierfür ist, dass es Kunden heute zunehmend leicht fällt, bei Unzufriedenheit den Anbieter zu wechseln. Hieraus entsteht für Softwarehersteller ein nicht unerhebliches Geschäftsrisiko, welches durch die Erhöhung der Budgets für Qualitätssicherung und Test minimiert werden soll.

Umso erstaunlicher ist es, dass der World Quality Report [Bue14] zeigt, dass ein großer Anteil der befragten Unternehmen keinen spezifischen Ansatz zur Qualitätssicherung in agilen Projekten hat.

Erschwerend kommt hinzu, dass unter Qualitätssicherung meist nur Qualitätskontrolle – im Sinne von dynamischen Softwaretests – verstanden wird und Maßnahmen zur proaktiven Qualitätssicherung

und -steuerung häufig nicht erkannt und/oder eingesetzt werden.

Im Folgenden werden weitere, insbesondere proaktive und präventive, Methoden zur Qualitätssicherung vorgestellt und aufgezeigt, wie diese in agilen Projekten eingesetzt werden können.

Also: „Wie Sie das Q in Agile finden und nutzen können.“

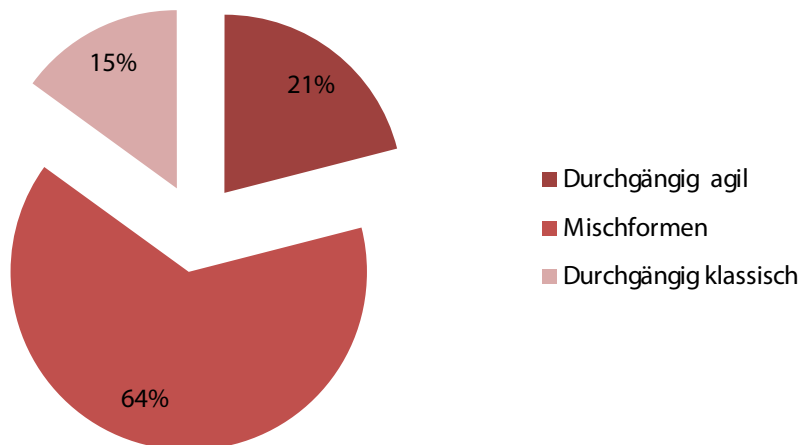


Abb. 1: Nutzung agiler Methoden [Kom14]

Das Q – Wie Sie Qualität sichern können

Zu den Hauptzielen von Qualitätssicherung zählen das Auffinden von Fehlern und die Schaffung von Vertrauen in die Qualität des erstellten Produkts. Um diese Ziele zu erreichen, werden in der Regel mehr oder weniger umfangreiche und strukturierte Softwaretests in den Projekten eingeplant.

Aber wäre es nicht viel besser und ökonomischer, Fehler gar nicht erst in die Software einzubringen?

Hier setzen die, neben der analytischen Qualitätssicherung, weiteren Säulen einer ganzheitlichen Qualitätssicherung an [Sch12]. Ein solches ganzheitliches Qualitätssicherungskonzept wird im Folgenden grob umrissen. Es enthält neben statischen und dynamischen Tests zur Fehlersuche und Qualitätskontrolle auch noch organisatorische und konstruktive Qualitätssicherungsmaßnahmen. Diese sollen die Einbringung von Fehlern von vornherein verhindern, oder zumindest verringern, und so Nacharbeiten vermeiden:

■ Organisatorische Qualitätssicherung: Ablauf- und Aufbauorganisation eines Projektes haben einen großen Einfluss auf die gelieferte Qualität. Daher ist es sehr wichtig, diese auch unter Qualitätsgesichtspunkten genau zu durch-

denken. So kann eine solide Basis für den Projekterfolg geschaffen werden. Wichtige Aspekte sind hier insbesondere die Einbindung der Projektorganisation in die Unternehmensorganisation, die verwendete Projektmanagementmethode und das genutzte Vorgehensmodell zur Softwareentwicklung.

■ Konstruktive Qualitätssicherung: Ziel von konstruktiven Qualitätssicherungsmaßnahmen ist die Verhinderung beziehungsweise Verringerung der Fehlereinbringung in das Produkt. Beispielsweise kann dies durch die Wiederverwendung erprobter Module und Architekturen oder die Nutzung von Frameworks für Design und Entwicklung erreicht werden. Eine weitere Methode, das Produkt von vornherein „richtig“ zu entwickeln, besteht darin, die Anforderungen an das Produkt durch den Einsatz von Prototypen zu überprüfen und zu verfeinern.

■ Analytische Qualitätssicherung: Auch wenn das Ziel eines ganzheitlichen Qualitätssicherungsansatzes die Verhinderung von Fehlern ist, so sind analytische Verfahren der Qualitätssicherung essenziell. Hiermit können fertige oder nahezu fertige (Zwischen-)Produkte der Softwareentwicklung geprüft werden. Sie dienen dem Auffinden von Fehlern und können durch geeignete

Metriken wichtige Aussagen über die Qualität der Software liefern. Zu den klassischen und häufig verwendeten Techniken gehören, neben dynamischen Softwaretests, auch statische Analysen und Reviews des Quellcodes.

Scrum als Rahmen

Als Rahmen für den Einsatz des oben beschriebenen ganzheitlichen Qualitätssicherungskonzepts dient die Projektmanagementmethode Scrum.

Neben seiner Praxisrelevanz aufgrund seiner hohen Verbreitung im agilen Umfeld verfügt Scrum noch über Eigenschaften, die eine Integration des beschriebenen Qualitätssicherungskonzepts erleichtern.

Zum einen ist Scrum ein sehr leichtgewichtiges Prozessrahmenwerk, welches es erlaubt, verschiedenste Techniken, Methoden und Prozesse zur Produkterstellung (und Qualitätssicherung) zu integrieren. Somit ist die Auswahl der verwendeten Qualitätssicherungsmethoden nicht von vornherein eingeschränkt.

Zum anderen enthält Scrum, trotz dieser hohen Freiheitsgrade, schon einige hilfreiche Qualitätssicherungsmaßnahmen.

Scrum gibt in erster Linie die Ablauf- und Aufbauorganisation des Projekts vor. Es werden keine Aussagen über die zur Produkterstellung verwendeten Methoden, beispielsweise für Analyse und Design, Entwicklung oder Qualitätssicherung, getroffen.

Abbildung 2 visualisiert den Prozessrahmen für einen Sprint, anhand dessen die wichtigsten Elemente der Methode kurz eingeführt werden sollen. Scrum im Detail wird an dieser Stelle nicht erklärt. Dem interessierten Leser sei der Scrum-Guide [SuSch13] ans Herz gelegt.

Basierend auf der Vision für das zu entwickelnde Produkt wird das Product Backlog zu Beginn des Projektes initial erstellt. Dieses Artefakt dient im Projektverlauf als einzige Anforderungsquelle, wengleich sein Inhalt jederzeit erweitert, verringert oder geändert werden kann.

Die Sprintplanung besteht aus zwei Teilen: der Festlegung des Sprintziels und des zugehörigen Entwicklungsumfangs, also das „Was“, sowie der Aufgabenidentifikation, dem „Wie“.

Im ersten Schritt erstellt das Entwicklungsteam zusammen mit dem Product Owner als dem Verantwortlichen für die Werthaltigkeit des zu entwickelnden Produkts eine Abschätzung, welche User Stories des Product-Backlogs sie im kommen-

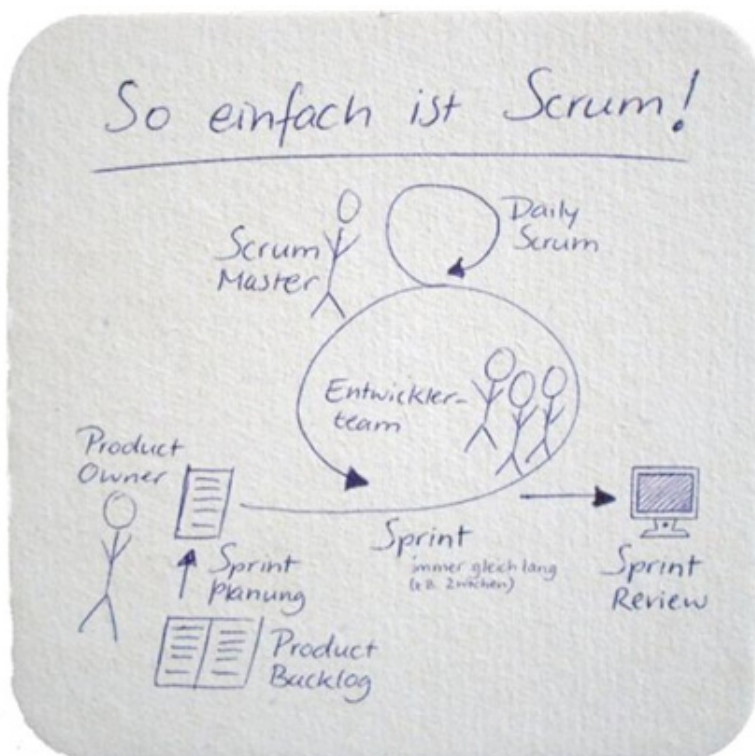


Abb. 2: Der Scrum-Prozessrahmen [Wol14]

den Sprint bearbeiten können. Auf dieser Basis definiert das gesamte Projektteam das Sprintziel und schafft ein gemeinsames Verständnis über die geplanten Inhalte.

Auf dieser Basis werden im zweiten Schritt die Aufgaben und Tätigkeiten, die zur Erreichung des Sprintziels nötig sind, durch das Entwicklungsteam identifiziert, geschätzt und geplant. Diese Aufgaben werden durch das Entwicklungsteam selbstorganisiert abgearbeitet.

Im eigentlichen Sprint entwickelt das Entwicklungsteam das Produktinkrement, welches im Sprint Review vorgestellt und abgenommen wird.

Zwischen zwei Sprints führt das Scrum-Team eine Retrospektive durch mit dem Ziel, die eigene Arbeitsweise zu optimieren.

Das gesamte Projekt lässt sich nun durch eine Aneinanderreihung von Sprints darstellen.

Qualitätssicherung in Scrum

Im folgenden Abschnitt werden konkrete Handlungsanweisungen zur Ausgestaltung eines Scrum-Projekts aus der Sicht der Qualitätssicherung gegeben. Allerdings können alle aufgezeigten Maßnahmen, mit gewissem Anpassungsaufwand, auch in anderen agilen Methoden zum Einsatz gebracht werden.

Projektinitiierung

Das Scrum-Team, welches ein Scrum-Projekt selbstorganisierend durchführt, stellt das Herzstück des Projekts dar. Aus diesem Grund ist die Teamzusammenstellung der kritische Erfolgsfaktor für das Projekt. Aus Sicht der Qualitätssicherung sind hier insbesondere zwei Maßnahmen entscheidend.

Der Product Owner ist jederzeit im Projektverlauf, die Anforderungen betreffend, entscheidungsfähig. Dies setzt voraus, dass er die fachliche Kompetenz und das Mandat hat, Anforderungen zu priorisieren und zu konkretisieren.

Die Teammitglieder besitzen neben den Fähigkeiten, die zur Entwicklung qualitativ hochwertiger Software nötig sind (z. B. Kenntnisse von Designmethoden oder der verwendeten Programmiersprache) auch Kenntnisse in Methoden der Qualitätssicherung. Dies liegt darin begründet, dass eine gesonderte Rolle für diese Aufgaben, zum Beispiel ein dedizierter Tester, in Scrum nicht vorgesehen ist.

Weiterhin werden im Rahmen der Projektinitiierung die Rahmenbedingungen

für das Projekt festgelegt. Aus Sicht der Qualitätssicherung sollten durch das Scrum-Team hier Vorgaben hinsichtlich der Programmierung in Form von Programmierrichtlinien festgelegt werden, um die Wartbarkeit der Software zu gewährleisten. Die Einhaltung der Richtlinien sollte automatisiert in Form statischer Analysen im Integrationsprozess geprüft werden.

Diese Vorgaben bilden zusammen mit den im Team festgelegten Qualitätszielen die Definition of Done (DoD). Die DoD enthält somit die zentralen Kriterien, die erfüllt werden müssen, damit ein (Zwischen-)Produkt als fertiggestellt betrachtet werden kann.

Da im laufenden Projekt aufgrund der festgelegten Sprintlänge Zeit eine wertvolle Ressource darstellt, ist es empfehlenswert, auch die Architektur der Lösung vorab auf die Infrastruktur der Organisation abzustimmen und festzulegen. Sind die technischen Rahmenbedingungen festgelegt, sollten auch die Testumgebung für das System und die zugehörigen Bereitstellungs-, Integrations- und Wartungsprozesse definiert und eingeführt werden, um einen reibungslosen Testablauf zu gewährleisten. Diese Tätigkeiten sind in vielen Scrum-Implementierungen Teil einer sogenannten „Iteration Zero“.

Sprintplanung

Die Sprintplanung hat den Charakter eines Quality Gates vor Beginn der Entwicklung. Diese Tatsache sollte ausgenutzt werden, um die zu implementierenden Anforderungen einem Review durch das Scrum-Team zu unterziehen. So kann sichergestellt werden, dass nur verständliche und eindeutige Anforderungen im Rahmen des Sprints bearbeitet werden. Dies kann durch ein Walkthrough Review im Rahmen der Sprintplanung nahezu aufwandsneutral gewährleistet werden. Jede User-Story, welche für den kommenden Sprint eingeplant werden soll, wird in diesem Rahmen besprochen, gegebenenfalls konkretisiert und ihre Akzeptanzkriterien werden dokumentiert.

Sprint

Qualitätssicherungsmaßnahmen während des Sprints dienen der Prüfung des Produktinkrements. Neben bereits angesprochenen statischen Analysen des Codes sollte dieser als weitere statische Qualitätssicherung auch einem Review unterzogen werden. Zu diesem Zweck empfeh-

lenswert ist die Praxis des Pair Programming aus dem Extreme Programming. Hierbei arbeiten zwei Entwickler am selben Stück Code. Während einer die eigentliche Programmierung vornimmt, prüft der zweite Entwickler den geschriebenen Code per Sichtprüfung.

Neben den statischen Qualitätssicherungsmaßnahmen wird der Code auch einem dynamischen Test unterzogen. Hierbei werden zwei Schwerpunkte gesetzt: zum einen die technische Richtigkeit der Software in Form von Whitebox-Tests und zum anderen der fachliche Test in Form von Blackbox-Tests.

Für die Whitebox-Tests wird eine weitere Praxis des Extreme Programming, das sogenannte Test-Driven Development, empfohlen. Hierbei werden vor dem Beginn der Programmierung ein oder mehrere automatisierte Unit-Tests geschrieben, um die technische Richtigkeit des zu entwickelnden Moduls nachzuweisen. Zur Qualitätssteigerung der Testfälle empfiehlt sich der Einsatz gängiger Testdesign-Techniken, wie Äquivalenzklassen, sowie die Nutzung von Überdeckungsmaßen zur Sicherstellung eines hohen Abdeckungsgrades. Die Testfälle können im Rahmen der automatisierten Integration der Lösung immer mitlaufen um sicherzustellen, dass jedes integrierte Modul technisch einwandfrei funktioniert.

Die fachlichen Tests sollten anhand eines strukturierten Testprozesses, wie TMap Next®, in den Ablauf integriert werden. Hinsichtlich der Testmethodik und -dokumentation muss an dieser Stelle zwischen Regressionstest und Tests der neuen Funktionalitäten unterschieden werden. Während Tests, die für einen Regressionstest vorgesehen sind, ausführlich und genau in Form von wiederholbaren automatisierten Testfällen dokumentiert werden, werden neue Funktionalitäten manuell mit explorativen Testmethoden getestet. Diese Ergebnisse der explorativen Testsessions werden zwar auch dokumentiert, allerdings eher rudimentär in Form von sogenannten Test-Chartern, also Anweisungen von Testzielen und möglichen Testideen, wie getestet werden soll.

Dieses Vorgehen gewährleistet zwar die Nachvollziehbarkeit der Testidee, hat aber nicht den Anspruch, den Test exakt wiederholen zu können. Der Vorteil dieses Vorgehens liegt darin, dass das System mit relativ geringem Aufwand intensiv getestet wird und Fehler frühzeitig gefunden werden.

Nachteil ist allerdings die fehlende Möglichkeit, Tests exakt zu wiederholen. Dieser Nachteil ist aber zu verschmerzen, da die exakte Wiederholung eines Tests in den seltensten Fällen neue Erkenntnisse bringt.

Sprintreview

Zum Abschluss des Sprints sieht Scrum das Sprintreview vor. Dieses Ereignis entspricht einem zweiten Quality Gate. Hierbei wird das im Rahmen des Sprints erstellte Produktinkrement den Stakeholdern vorgeführt und durch den Product Owner die Abnahme anhand der Definition of Done und der Akzeptanzkriterien bestätigt.

Hierdurch wird gewährleistet, dass das richtige Produkt entwickelt wurde beziehungsweise dass der nächste Sprint auf einem korrekten Zwischenergebnis aufbaut.

Sprintretrospektive

Um die Arbeit des Scrum-Teams zu optimieren, ist zwischen den Sprints noch eine weitere organisatorische Qualitätssicherungsmaßnahme in die Methode integriert. Im Zuge der Retrospektive analysiert das Team seine Arbeitsweise und identifiziert Verbesserungsmaßnahmen zur Umsetzung in den folgenden Sprints.

Diese kritische Reflexion der eigenen Arbeitsweise und der hierdurch erzielten Ergebnisse führt zu einem höheren Qualitätsbewusstsein des gesamten Teams und so zu einer höheren Produktqualität der folgenden Inkremente.

Ziel: „No Faults Forward“

Durch das oben beschriebene Vorgehen ist es möglich, Fehler möglichst nahe am Ursprungspunkt zu identifizieren und zu beheben und somit nicht in Folgeprodukte zu übernehmen.

Durch das Quality-Gate Sprintplanung werden nur User Stories in den Entwicklungsprozess übernommen, die verständlich und korrekt sind. Sollten im Rahmen der Weiterverarbeitung noch Unklarheiten bestehen, können diese schnell und direkt durch den Product Owner aufgeklärt werden.

Jeder Teil des Produktinkrements durchläuft im beschriebenen Prozess mehrere Qualitätssicherungsstufen. Ziel dieses Vorgehens ist es, wie oben beschrieben, dass Fehler möglichst früh gefunden und behoben werden. Die erste Stufe erfolgt noch während des Programmierens. Hier werden durch Pair Programming sowohl der Komponentenentwurf als auch der Code kontinuierlich einem Review unterzogen. Durch den Ansatz des Test-Driven Development besteht für jedes Stück Code eine ausreichende Anzahl automatisierte Tests, welche im Buildprozess zusammen mit statischen Analysen des Codes durchgeführt werden.

Das so qualitätsgesicherte System wird nun noch mittels explorativer Tests manu-

ell auf Richtigkeit und Angemessenheit geprüft.

Als letzte „Verteidigungslinie“ vor der Auslieferung an den internen oder externen Kunden sieht die Methode Scrum noch ein zweites Quality Gate, das Sprintreview, vor. Hier können sich die Stakeholder nochmals selbst überzeugen, ob das System ihren Ansprüchen genügt.

Durch die relativ kurzen Zeiträume, die ein Sprint einnimmt, werden, im Gegensatz zu traditionellen sequenziellen Ansätzen, selbst hier auftretende Fehler noch „frühzeitig“ erkannt.

Diese Vielzahl an aufeinander abgestimmten Qualitätssicherungsmethoden und -techniken wird durch den integrierten kontinuierlichen Verbesserungsprozess der Methode in Form der Sprintretrospektive abgerundet. So wird gewährleistet, dass nicht nur von vornherein qualitativ hochwertige Software entwickelt werden kann, sondern das Team im Projektverlauf auch noch immer besser wird. ■

Literatur & Links

- [Bue14] M. Buenen, M. Teje, World Quality Report 2014-15, Studie Capgemini, Sogeti und HP, 2014
- [Kom14] A. Komus, Status Quo Agile 2014, BPM Labor HS Koblenz, 2014
- [Sch12] K. Schneider, Abenteuer Softwarequalität - Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement, dpunkt.verlag, 2012
- [SuSch13] J. Sutherland, K. Schwaber, Der Scrum-Guide – Der gültige Leitfaden für Scrum: Die Spielregeln, Juli 2013, siehe <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-DE.pdf>
- [Wol14] H. Wolf, R. van Solingen, E. Rustenberg, Die Kraft von Scrum - Inspiration zur revolutionärsten Projektmanagementmethode, dpunkt.verlag, 2014

TMap NEXT® ist ein eingetragenes Warenzeichen der Sogeti Nederland B.V.