

# **Agile Entwicklungsprozesse im normativ regulierten Umfeld**

## **Die Rolle der Qualitätssicherung auf dem Weg zur Zertifizierung**

Rosalinde Schuster & Christoph Legat, Berner & Mattner Systemtechnik GmbH

### **Einleitung**

**Um in einem dynamischen, internationalen Markt wettbewerbsfähig zu bleiben, streben Unternehmen nach stets kürzeren und flexibleren Entwicklungszyklen. Im Zuge dieser Entwicklung gewinnen agile Entwicklungsmethoden zunehmend an Bedeutung [1]. Entsprechend dem sogenannten Agilen Manifests [2] wird dabei Individuen und deren Interaktionen, ein funktionierender Produktstand, die Zusammenarbeit mit dem Kunden und die Reaktion auf Veränderungen mehr Gewicht beigemessen als Prozessen und Werkzeugen, einer umfassenden Dokumentation und der Stabilität eines einmal erstellten Plans. Demgegenüber steht jedoch die Anforderung, spezifischen Normen für den Entwicklungsprozess und/oder Produkten zu entsprechen. Eine Norm ist dabei eine Leitlinie, welche Informationen zur Vereinheitlichung über ein Produkt oder einen Prozess enthält und einen anerkannten Stand der Technik beschreibt [3]. Dies führt zu einem Spannungsfeld zwischen Agilität und der durch Normen geforderten durchgängigen Dokumentation qualitätssichernder Maßnahmen. Dennoch bieten einige häufig in der agilen Entwicklung eingesetzten Elemente bereits eine solide Grundlage für die Erfüllung von Normen und damit einer Zertifizierung, wenn diese geeignet genutzt und adäquat angepasst werden.**

In diesem Zusammenhang stellt Scrum [4] einen einfachen, weit verbreiteten Rahmen für die Organisation der Arbeit in agilen Teams dar: ein fixe Länge von Iterationen (sog. Sprints) mit jeweils einem lauffähigen Inkrement als Produktversion. Scrum wird daher im Folgenden als Ausgangspunkt für ein erweitertes Vorgehensmodell – Scrum QA – genutzt, welches Vorteile agiler Entwicklung mit der Zielsetzung vereint, eine Grundlage für eine Zertifizierung durch Festlegung von Anforderungen und regelkonforme Verifizierung des Entwicklungsartefakten durch Test und Rückverfolgbarkeit zu schaffen.

Nachfolgend wird eingangs ein Überblick über den Scrum QA zu Grunde liegenden Entwicklungsprozess und dessen Aktivitäten gegeben. In den anschließenden beiden Abschnitten werden die Rollen und Artefakte des vorgeschlagenen Vorgehensmodells detailliert beschrieben. Der Beitrag schließt mit einer Zusammenfassung und gibt einen Ausblick über weitere Entwicklungen.

### **Scrum QA-Prozess: Aktivitäten im Überblick**

Um eine normgerechte Entwicklung und angestrebte Zertifizierung geeignet zu berücksichtigen, müssen die in Scrum genutzten Aktivitäten (siehe hierzu beispielsweise Schwaber und Sutherland [4]) entsprechend den Anforderungen, die sich aus etwaigen Normen ergeben, ergänzt werden. Einen Überblick über den durch den Scrum QA zu Grunde gelegten Entwicklungsprozess gibt Abbildung 1.

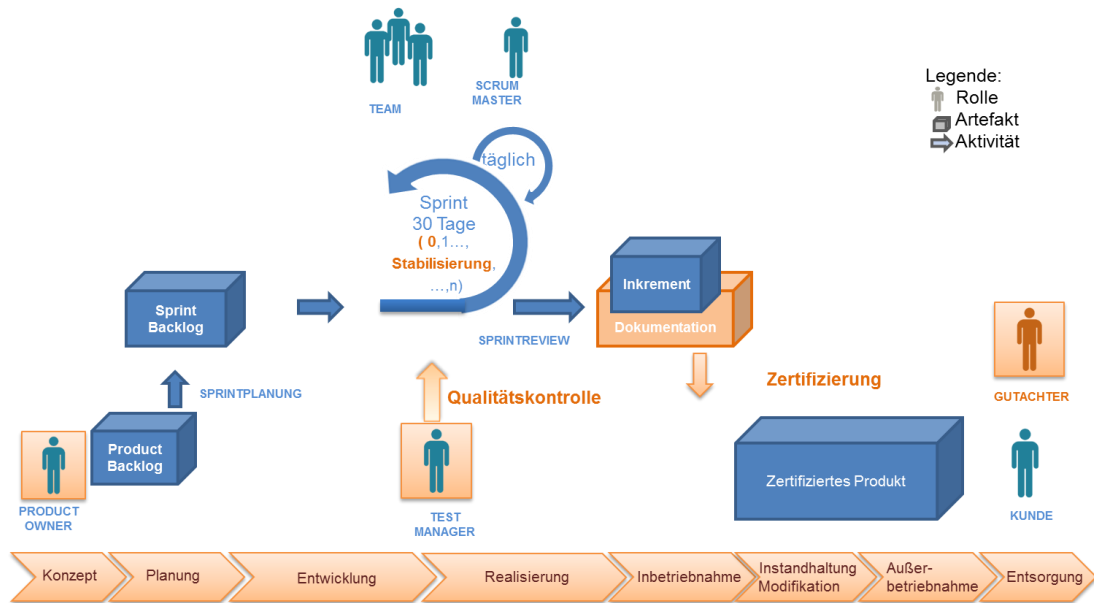


Abbildung 1: Überblick der Aktivitäten und Artefakte von Scrum QA im Überblick. Im Vergleich zu Scrum ergänzte oder angepasste Elemente sind hervorgehoben.

Entsprechend Scrum werden initiale Vorbereitungen in **Sprint 0** durchgeführt. Während dieser Sprint im klassischen Scrum optional durchgeführt werden kann, ist er für die Entwicklung im normativ regulierten Umfeld essentiell. Bereits hier kann die Grundlage für eine spätere Zertifizierung gelegt werden: So werden im Zuge von Sprint 0 geeignete Überwachungsmaßnahmen zur Erkennung zufälliger Fehlern und Maßnahmen zur Verhinderung systematischer Fehlern in der Entwicklung definiert. Ferner werden aus den zu Grunde gelegten Normen entsprechende Qualitätsanforderungen, Risikobeseitigungs- und Risikominimierungsstrategien sowie Anforderungen hinsichtlich der Architektur des Gesamtsystems (z.B. zur Sicherstellung der Testbarkeit wird in Normen zur funktionalen Sicherheit eine konsequente, modulare Architektur gefordert) abgeleitet. Darauf basierend kann ein adäquates Testkonzept, welches der Testplanung mit Beschreibung der entsprechenden Testobjekte und der zugehörigen Testaufgaben entspricht, erstellt werden. Die sogenannte „Definition of Done“, welche eine Liste von Fertigstellungskriterien (z.B. Qualität, Skalierbarkeit) des Produktes darstellt, wird ferner um die üblicher Weise in Normen geforderten Aspekte hinsichtlich Begutachtung und Dokumentation erweitert.

Das sogenannte **Product Backlog** bildet in Scrum die zentrale Datenbasis etwaiger, zu realisierender Funktionen und Zielsetzungen. Bereits bei der Erstellung von Einträgen des Backlogs kann die Testbarkeit geeignet sichergestellt werden. Hierdurch können häufige Herausforderungen bei der Zertifizierung – z.B. der Nachweis der funktionalen Sicherheit – zielgerichtet berücksichtigt werden.

Entsprechend dem klassischen Scrum werden die Entwicklungen im Zuge einzelner Iterationen (**Sprints**) mit vorgegebener Länge durchgeführt. Zielsetzung jedes Sprints ist stets ein lauffähiges Produkt, entsprechend den für einen Sprint aus dem **Product Backlog** ausgewählten, zu realisierenden Funktionen zu entwickeln. In das **Sprint Backlog** werden bei der Sprintplanung die im Rahmen eines Sprints zu erledigende Aufgaben übernommen und abgearbeitet. Daher ist im Zuge jedes

Sprints eine Vorbereitung und (automatische) Durchführung von Komponenten- und Integrationstests notwendig. Wird die Sprintdauer zu kurz gewählt, können diese, für eine spätere Zertifizierung notwendigen Aktivitäten als Teil der benötigten Regressionsteststrategie, nicht geeignet durchgeführt werden. In der Praxis erwies sich deshalb eine Sprintdauer von 30 Tagen als geeignetes Mittelmaß zwischen Agilität und Kontinuität.

Bedingt durch die Dokumentationspflicht, die sich aus den Normen ergibt, hat sich der **Stabilisierungssprint** als Sonderform eines Sprints bewährt, welcher üblicherweise vor einem größeren Release eingeschoben wird. Während in klassischen Sprints die Realisierung spezifischer Funktionalität fokussiert wird, dient der Stabilisierungssprint dem Nachweis der Normkonformität des Produktes und der Dokumentation. So werden im Zuge dieses Sprints auch fehlende Nachweise zur Erfüllung etwaiger Sicherheitsanforderungen (soweit benötigt) ergänzt bzw. an geänderte Funktionalitäten angepasst. Den Stabilisierungssprint charakterisieren eine vermehrte Durchführung von Inspektionen und Reviews sowie deren entsprechende Dokumentation. Auch etwaige Rückmeldungen durch **Gutachter** in Bezug auf die Normeneinhaltung werden eingearbeitet und finden damit Berücksichtigung in der Realisierung – d.h. in die Entwicklungsartefakte, wie der Quellcode in der Softwareentwicklung – und dessen Dokumentation.

An dieser Stelle sei ferner noch die Bedeutung geeigneter Werkzeugumgebungen betont, welchen im Kontext einer agilen Entwicklung und speziell im normativ regulierten Umfeld für die Qualitätssicherung eine besonders große Bedeutung zukommen: Um eine kontinuierliche Qualitätssicherung zu gewährleisten und die regelmäßige Aktualisierung notwendiger Dokumentation für eine spätere Zertifizierung effizient erstellen zu können, ist neben einem Konfigurationsmanagement, um die Verfolgbarkeit von Änderungen über den Lebens- bzw. Entwicklungszyklus sicherzustellen, eine automatisierte Testumgebungen für die effiziente Wiederholbarkeit von Test (Regressionstrategie) essentiell. Während die sog. Continuous Integration – die fortlaufende Zusammenführung entwickelter Teilkomponenten und Funktionen – in der Softwareentwicklung bereits im praktischen Einsatz weit verbreitet ist [1], stellen automatisierte Erstellungs- und Testprozesse in interdisziplinären Systementwicklungsprozessen noch stets eine individuelle Herausforderung dar.

### **Rollen und Verantwortlichkeiten in Scrum QA**

Die Verwendung von Rollen und Verantwortlichkeiten für einzelne Aktivitäten resultiert in klare Zuständigkeiten für die Umsetzung bestimmter, wichtiger Aufgaben – insbesondere solcher, um zu Grunde gelegte Normen und eine entsprechende Zertifizierung erreichen zu können. Daher werden im Folgenden die Anpassungen und Besonderheiten von Rollen und Verantwortlichkeiten des hier vorgeschlagenen Vorgehensmodells Scrum QA beschrieben.

Der **Product Owner** zeichnet sich für die Produktentwicklung verantwortlich und stimmt sich hierfür mit allen relevanten Stakeholdern ab, um Entscheidungen treffen zu können. Im Scrum QA Prozess ist er zusätzlich verantwortlich für das Risikomanagement und nimmt Kontakt zu entsprechenden, externen **Gutachtern** für die Zertifizierung auf.

Der **Scrum Master** ist der Coach, Moderator und Vermittler für alle Prozessbeteiligten. Zu seinen Aufgaben bei der Unterstützung des Teams gehört auch die Bereitstellung verwendeter Normentexte und entsprechender Vorlagen zur normenkonformen Dokumentation. Aufgrund seiner Kommunikationsfähigkeiten kann der Scrum Master auch als Moderator bei durch Normen geforderten Inspektionen und Audits eingebunden werden.

Das **Team** zeichnet sich für Umsetzung verantwortlich. In Abhängigkeit der durch die zu Grunde gelegten Normen gestellten Anforderungen werden – sofern möglich – Tests unter Berücksichtigung der in Sprint 0 bereits festgelegten Qualitätskriterien (z.B. spezifische Testabdeckungen) getestet. Ebenso fällt die Erstellung der Dokumentation in den Aufgabenbereich des Teams, um die Rückverfolgbarkeit von Anforderung bis zum Test sicherzustellen.

Ergänzend zu den drei in Scrum üblicherweise verwendeten Rollen, fordert Scrum QA den Einsatz eines **Testmanagers**, dessen Hauptaufgabe die Einhaltung der Normen und die Qualitätssicherung als unabhängige Instanz darstellt [3]. Zusätzlich fungiert der Testmanager als Ansprechpartner in Fragen hinsichtlich technischer oder organisatorischer Anforderungen, die sich aus den zu Grunde gelegten Normen ergeben. Die Definition des Testkonzepts unter Berücksichtigung etwaiger durch Normen implizierter Anforderungen gehört ebenfalls zu seinen Aufgaben. So ist er beispielsweise im Falle von funktionaler Sicherheit, für die Ergänzung technischer Berichte mit den Ergebnissen der Qualitätssicherungsmaßnahmen, wie z.B. Tests, Durchführung von Reviews und Inspektionen für den Nachweis von Fehlerfreiheit verantwortlich.

### **Artefakte von Scrum QA**

Die Ergebnisse von Aktivitäten innerhalb des Entwicklungsprozesses werden als Artefakte bezeichnet. Bei der Erstellung des **Inkrement**s, welches in Scrum nur implizit genannt wird, bei Scrum QA jedoch als explizites Artefakt genutzt wird, ist darauf zu achten, dass etwaige Entwicklungsartefakte instrumentalisierbar sind, d.h. dass das Design durch geeignete Maßnahmen für unterschiedliche Teststufen prüfbar gemacht werden kann.

Für die Nachverfolgbarkeit von der Anforderung bis zum Testergebnis – und letztendlich für eine erfolgreiche Zertifizierung – werden zusätzliche **Dokumentationen** benötigt und in Scrum QA als Artefakt explizit genutzt. Die Erstellung geforderter, detaillierter Dokumentation widerspricht dabei teilweise dem Agilen Manifest, ist jedoch in normativ reguliertem Umfeld unumgänglich. So werden durch Normen beispielweise Risikolisten oder Ergebnisdokumentationen von Tests funktionaler und nicht-funktionaler Anforderungen gefordert. Werden ferner spezifische, funktionale Sicherheitsanforderungen gestellt, müssen die Ergebnisse von Verifikation (d.h. Nachweis der richtigen Erstellung des Entwicklungsergebnisses) und Validierungstests (d.h. Prüfung des Entwicklungsergebnisses hinsichtlich Erfüllung der Anforderungen bezüglich beabsichtigter Nutzung) ebenso wie andere qualitätssichernden Maßnahmen in einem technischen Sicherheitsbericht zusammen mit sicherheitsrelevanten Kennwerten des Systems verknüpft werden. Dieser Bericht dient dem durch den Product Owner beauftragten Gutachter als Basis für die Zertifizierung.

## **Zusammenfassung und Ausblick**

Im Rahmen dieses Beitrags wurde Scrum QA als eine Erweiterung von Scrum für die agile Entwicklung in normativ reguliertem Umfeld beschrieben. Durch Erweiterung der Aufgaben bereits existierender Rollen sowie der Einführung zusätzlicher Rollen kann eine kontinuierliche, agile Qualitätssicherung und letztendlich eine erfolgreiche Zertifizierung sichergestellt werden.

Agile Teams in normativ reguliertem Umfeld benötigen eine hohe Testerfahrung und methodische Testexpertise, um die durch Normen geforderten Test(automatisierungs)-Strategien zu bestimmen und entsprechend der relevanten Normen und Richtlinien anzuwenden. Als wesentlich erweist sich dabei die durchgängige Sicherstellung der Testbarkeit von Design bis Realisierung. Auf Grund der Agilität des Vorgehens und der damit verbundenen erhöhten Kommunikation zwischen Beteiligten können Test-Management-Umgebungen schlanker gehalten werden; kontinuierliche Integration entwickelter Lösungskomponenten und durchgängiges Konfigurationsmanagement stellen wesentliche Eckpfeiler für die Dokumentation und Zertifizierung dar.

In der Praxis konnte positiv evaluiert werden, dass das in diesem Beitrag beschriebenes Vorgehen sowie damit verbundene Maßnahmen einen positiven Einfluss auf eine effiziente Projektdurchführung mit erfolgreicher Zertifizierung aufweist. Als erfolgskritisch erwies sich dabei jedoch die geeignete Auswahl von Projektbeteiligten in Bezug auf der korrekten Ableitung und Auswahl von Maßnahmen zur Qualitätssicherung aus den zu Grunde gelegten Normen sowie deren geeignete, agile Evolution.

## **Literaturverzeichnis**

- [1] Frank Simon, Manuel Fischer, Karin Vosseberg, Andreas Spillner, Kai Lepler und Mario Winter: „ Management-Summary der GTB Softwaretestumfrage 2015/2016“ , April 2016
- [2] Peter Hruschka und Chris Rupp: "Agile Softwareentwicklung für Embedded Real-Time Systems mit UML", Carl Hanser Verlag München Wien, 2002.
- [3] Andreas Spillner, Thomas Roßner, Mario Winter und Tilo Linz: „Praxiswissen Softwaretest Testmanagement“, dpunkt.verlag, 4. Auflage, Mai 2014.
- [4] Ken Schwaber und Jeff Sutherland: „Der Scrum Guide. Der gültige Leitfaden für Scrum: Die Spielregeln“, Juli 2013.

## Autoren



**Rosalinde Schuster** (ISTQB® Full Advanced Certified Testerin) hat mehr als 25 Jahre praktische Industrieerfahrung in den Bereichen Softwareentwicklung, Test und Training von (eingebetteten) Softwaresystemen für unterschiedliche Domänen und Anwendungen. Seit diesem Jahr führt sie zusätzlich die ISTQB® Certified Tester - Foundation Level, Extension Agile Schulung durch.



**Christoph Legat** studierte Informatik mit Schwerpunkt Softwaretechnik an der LMU München. Seit mehr als 8 Jahren arbeitet er in Forschung und Entwicklung flexibler, intelligenter Automatisierungssoftware (u.a. TU München, Siemens AG, Berner & Mattner). Er war/ist in einer Vielzahl von öffentlich geförderten sowie privatwirtschaftlichen Projekten im Kontext von Industrie 4.0 involviert und ist Autor von mehr als 50 wissenschaftlichen Publikationen sowie mehr als 10 erteilten Patenten.

## Kontakt

Berner & Mattner Systemtechnik GmbH  
Erwin-von-Kreibitz-Straße 3  
80807 München

Internet: [www.berner-mattner.com](http://www.berner-mattner.com)  
Email: [rosalinde.schuster@berner-mattner.com](mailto:rosalinde.schuster@berner-mattner.com)  
[christoph.legat@berner-mattner.com](mailto:christoph.legat@berner-mattner.com)