Qualitätssicherung (QS) im agilen Umfeld

Veröffentlicht am 15. August 2018

Ihre Ansprechpersonen: Sandro Rao

Tags: Fachbeiträge, Business Analyse und Requirements Engineering

Agile Vorgehensmodelle und SCRUM

Der Projektalltag ist oft geprägt von laufend ändernden Rahmenbedingungen und Anforderungen. Um die daraus resultierende Komplexität bei der Entwicklung von Systemen zu meistern, kommen vermehrt agile Vorgehensmodelle zum Einsatz.

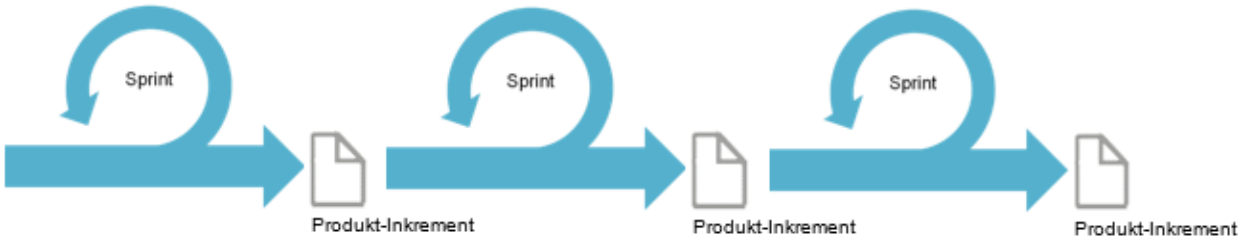
Agile Vorgehensmodelle haben zum Ziel, die Entwicklung flexibler und schlanker zu gestalten, um so schnell auf geänderte Rahmenbedingungen reagieren zu können. Die agile Entwicklung zeichnet sich durch ein iteratives und inkrementelles Vorgehen aus. Dabei werden in Entwicklungszyklen kontinuierlich Teile bzw. Verbesserungen eines Systems ausgeliefert. Im Gegensatz zu klassischen Wasserfallprojekten kann so rascher auf Änderungen reagiert werden. Das klassische Wasserfallmodell beruht auf vorgängig definierten und aufeinander aufbauenden Phasen. Dabei muss eine Phase komplett abgeschlossen sein, damit die nächste Phase beginnen kann. Wenn Fehler nicht direkt während einer laufenden Phase entdeckt werden, ist es deshalb oft schwierig, diese nachträglich zu korrigieren.

Der Einsatz eines agilen Vorgehensmodells an sich garantiert allerdings noch kein erfolgreiches Ergebnis. Mit den steigenden Erwartungen der Kunden nimmt auch die Wichtigkeit der Qualität des Endprodukts zu. Der zentrale Aspekt der Qualitätssicherung soll nachfolgend anhand des wohl bekanntesten agilen Vorgehensmodells SCRUM beleuchtet werden.

Ausgangslage und Problemstellung

SCRUM sieht, im Gegensatz zu klassischen Vorgehensmodellen, keine eigenständige Phase zur Qualitätssicherung vor, sondern verlangt, dass alle diesbezüglichen Aktivitäten während des Entwicklungszyklus durchgeführt werden. In SCRUM wird ein solcher Entwicklungszyklus Sprint genannt. Der Sprint hat eine fixierte Zeitdauer und Ziel des Sprints ist es am Ende ein fertiges, nutzbares und

potenziell auslieferbares Produkt?Inkrement auszuliefern (siehe Abbildung).



Gemäss SCRUM Guide werden für die Qualitätssicherung nur wenige, explizite Vorgaben formuliert:

- Das SCRUM-Team muss sich auf eine «Definition of Done» einigen, anhand welcher festgelegt wird, wann die Umsetzung einer Anforderung abgeschlossen ist.
- Eine Anforderung gilt als umgesetzt, wenn die Entwicklung des entsprechenden Features die «Definition of Done» erfüllt.
- Im Sprint Review Meeting muss das Entwicklungsteam den aktuellen Entwicklungsstand präsentieren und darüber Auskunft geben, welche Anforderungen gemäss der «Definition of Done» abgeschlossen worden sind.

Darüber hinaus müssen die Aktivitäten zur Qualitätssicherung im SCRUM-Team selber gestaltet und definiert werden. Aus der Erfahrung der APP hat sich gezeigt, dass diesem Punkt in der Projektplanung in der Praxis oftmals zu wenig Beachtung geschenkt wird und das Vorgehen mit Verunsicherungen und Unklarheiten verbunden ist. Dies führt im Projektverlauf dazu, dass das Projektteam beispielsweise mit folgenden Fragen konfrontiert wird:

- Wie und wann wird getestet?
- Wann werden zum Beispiel Regressions- und Integrationstests durchgeführt?
- Wie viel muss in jedem Sprint getestet werden?
- Auf Basis welcher Testfälle wird getestet?
- Wer testet?
- Wer ist für die Qualität der Applikation verantwortlich?

?Vorgehen

Zur Beantwortung dieser Fragestellungen und Definition einer zuverlässigen und zielführenden Qualitätssicherung ist die Berücksichtigung von projektspezifischen Faktoren wie Grösse und Komplexität essentiell. Unabhängig davon haben sich in den durch die APP begleiteten SCRUM-Projekten folgende Faktoren als zentral erwiesen:

1. Grundlagen schaffen

Die Grundlagen für die Qualitätssicherung sind sauber dokumentierte Anforderungen. Diese werden üblicherweise als User Stories mit entsprechenden Akzeptanzkriterien erhoben und im Product Backlog gesammelt. Dabei ist von Anfang an sicherzustellen, dass diese ausreichend präzis und detailliert ausformuliert werden, damit anschliessend nach Akzeptanzkriterien getestet werden kann.

Da sich Anforderungen über den gesamten Entwicklungs- und Lebenszyklus ändern können, ist es zudem zentral, dass Änderungen für alle Beteiligten des SCRUM-Teams nachvollziehbar gemacht werden, damit diese auch bei der Qualitätssicherung berücksichtig werden können.

2. Vorgehen definieren

Bevor die Entwicklungsarbeiten begonnen werden, muss definiert werden, welche Art von Test (bspw. Unit-Test, Regressionstest etc.) zu welchem Zeitpunkt und in welchem Intervall durchgeführt werden muss. Hier sind insbesondere Aufwand und Nutzen der vorgesehenen Qualitätssicherungsmassnahmen im Auge zu behalten. Bspw. können komplette Regressionstests in jedem Sprint für die Qualität sinnvoll sein, führen aber auch zu hohen Aufwänden.

Ein wichtiger Punkt in diesem Zusammenhang ist auch, dass in einem Sprint genügend Zeit für das Bugfixing sowie das anschliessende Retesting eingeplant wird.

3. Rollen klären

Um die Qualität sicherzustellen, muss klar geregelt sein, wer in der Qualitätssicherung welche Aufgaben und Verantwortungen übernimmt. Insbesondere muss definiert werden, wer schlussendlich die Verantwortung für die Qualität des Endergebnisses übernimmt sowie die Qualitätsziele mit dem Auftraggeber definiert, in der Projektorganisation einfordert und überwacht.

Grundsätzlich sollte im SCRUM-Team das Bewusstsein gestärkt werden, dass jedes Teammitglied seinen Beitrag zur Qualität leisten kann und soll.

Umgesetzte Lösungsansätze

Mit den beschriebenen Faktoren konnte die APP bereits in einigen SCRUM-Projekten Erfahrungen sammeln. Im folgenden Abschnitt werden verschiedene Problematiken aus der Praxis sowie daraus erarbeitete und umgesetzte Lösungsansätze geschildert.

1. Grundlagen schaffen

Problematik: Business Analyse

Im Rahmen der Business Analyse wird dem Testing zu wenig Beachtung geschenkt und die User Stories sind nicht so formuliert, dass ein Testing nach Akzeptanzkriterien unterstützt wird.

Lösungsansätze:

Definieren und Involvieren der Verantwortlichen für die Qualitätssicherung bereits bei der Formulierung der User Stories, um sicherzustellen, dass die User Stories sowie der Sprint-Umfang verstanden werden.
Ein grober Aufwand für das Testing sollte zudem bereits bei der Schätzung der User Stories eingeplant werden, damit diese Aufgabe nicht unterschätzt wird.

Der Einsatz eines agilen Coaches kann zielführend sein, um das SCRUM-Team bei der Formulierung und Dimensionierung der User Stories sowie bei der Festlegung von sinnvollen Akzeptanzkriterien zu unterstützen.

2. Vorgehen definieren

Problematik: Regressionstests

Die Systemarchitektur besitzt eine hohe Komplexität und/oder mehrere SCRUM-Teams arbeiten parallel. Dadurch haben Regressionstests eine grosse Bedeutung, sind jedoch auch mit hohem Aufwand verbunden.

Lösungsansätze:

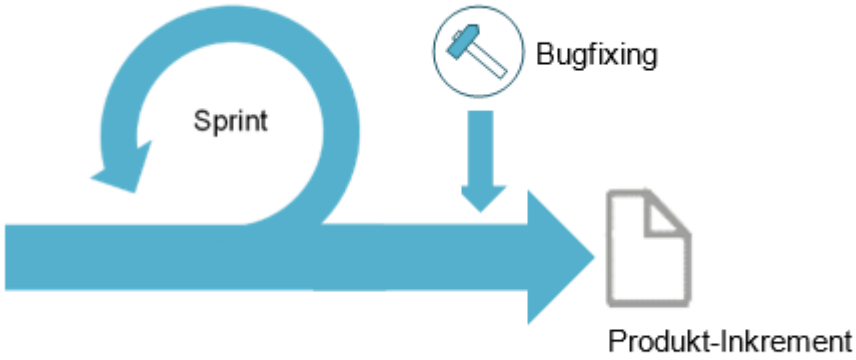
Einführen automatisierter Regressionstests, die regelmässig durchgeführt werden können. Gleichzeitig sicherstellen, dass nicht-automatisierbare Regressionstests trotzdem eingeplant und durchgeführt werden. Auch in agilen Entwicklungsprojekten gilt, dass je später Fehler entdeckt werden, umso höher sind die mit der Behebung verbundenen Kosten sind.

Problematik: Zeit für Bugfixing

Oftmals findet das Sprint Planning für den nächsten Sprint in einem kurzen zeitlichen Abstand nach dem Sprint Review des vergangenen Sprints statt. Werden nach dem Sprint Review noch Bugs festgestellt, kann es vorkommen, dass der neue Sprint bereits gestartet hat und die Behebung der Bugs verzögert sich um bis zu 4 Wochen.

Lösungsansätze:

Um dies zu verhindern, sollte das Entwicklungsteam 2 – 3 Tage vor Sprintende eine möglichst stabile Lösung auf der Testumgebung deployen. Somit bleiben 2 – 3 Tage, um zu testen und allfällige Bugs gleich zu beheben oder für den nächsten Sprint einzuplanen.



3. Rollen klären

Problematik: Verantwortung für die Qualität

Die bisher klar getrennten Rollen von QS und Entwicklung verschmelzen im SCRUM-Team, da SCRUM keine explizite Rolle für das Testing vorsieht. Dadurch wird das Gefühl für die Zuständigkeit und die Verantwortung für die Qualität verwässert. Das Bewusstsein „Ich bin für die Qualität der Applikation verantwortlich.“ kann verloren gehen.

Lösungsansätze:

Es kann hilfreich sein, Tester im Entwicklungsteam zu definieren und diese Verantwortungen im SCRUM-Team auch zu formalisieren. In diesem Zusammenhang kann auch der dedizierte Einsatz eines Testadministrators, der sich hauptsächlich um die Qualitätssicherung kümmert, sinnvoll sein.

Problematik: Testing bei paralleler Entwicklung durch mehrere SCRUM-Teams

Wenn mehrere SCRUM-Teams parallel ein Produkt entwickeln, kann es schwierig sein, die Zuständigkeit und Verantwortung für die Qualitätssicherung zuzuordnen.

Lösungsansätze:

Testing durch ein sogenanntes Testcenter durchführen lassen, welches das Testing für alle SCRUM-Teams übernimmt. Die Verantwortung für die Qualität für die einzelnen Entwicklungsprojekte liegt bei den einzelnen Product Owner, die Verantwortung für das Gesamtergebnis beim Gesamtprojektleiter.

Fazit

Ein wesentlicher Erfolgsfaktor für die Qualitätssicherung in agilen Projekten ist sicherlich, dass man Grundlagen für die Qualitätssicherung schafft, ein konkretes Vorgehen definiert und im Team ein gemeinsames Rollenverständnis schafft.

Es ist aber in diesem Zusammenhang ebenso wichtig, dass man die Zusammenarbeit im Projektteam regelmässig überprüft, Verbesserungspotentiale identifiziert und in Form von konkreten Massnahmen umsetzt. SCRUM beispielsweise sieht eine solche Feedbackrunde in der Sprint Retrospektive vor. Diese Möglichkeit zur Verbesserung sollte unbedingt genutzt werden.

Möchten Sie mehr über dieses spannende Thema erfahren oder wissen, wie die APP auch Sie bei einem herausfordernden Vorhaben unterstützen kann? Wir freuen uns auf Ihre Kontaktaufnahme.