

In dieser Ausgabe:

- ⇒ Wintersteiger, Mathis: Agile Developers Skills: Effektives Arbeiten in einem Scrum-Team
- ⇒ Osheroove: The Art of Unit Testing
- ⇒ Rupp: Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis
- ⇒ Entin, V.; Winder, M.; Bo Zhang; Christmann, S; Software Quality Assurance, Omicron Electron. GmbH, Klaus, Austria; Introducing Model-Based Testing in an Industrial Scrum Project; ICSE 2012, 7th International Workshop on Automation of Software Test

**Notwendigkeit des Testens in agilen Prozessen
(SCRUM)**

Agile Prozesse und im Besonderen SCRUM haben in den letzten Jahren eine große Verbreitung und Beliebtheit gefunden. Leider verleitet die Eingängigkeit des Ansatzes und die Offensichtlichkeit seiner Vorteile immer wieder zu der Annahme, dass Qualität des Endproduktes implizit und „out of the box“ automatisch entsteht, wobei dies auch zusätzlich dadurch motiviert wird, sich auf diese Weise lästigen Nebenaufgaben, wie z.B. eine Dokumentation, über das im agilen Manifest beschriebene Maß hinaus zu entledigen. Aussagen wie: „Wir testen nicht systematisch, wir arbeiten nach SCRUM und sichern die Qualität gleich bei der Entwicklung.“ sind daher keine Seltenheit. So finden sich denn auch immer wieder Projekte, bei denen u.a.

- erst am Ende erkannt wird, dass Anforderungen nicht erfüllt wurden,
- eine umfassende Verifikation und Validierung nach dem letzten Sprint notwendig sind
- und Zeit und Budget deutlich überzogen wurden.

Letztlich kann man sagen, dass ein Prozessansatz nur den Rahmen vorgibt, der je nach Ausprägung entweder mehr oder eben weniger die Entstehung von Qualität unterstützt. SCRUM und andere agile Methoden gehören in diesem Sinne zwar zu qualitätsgetriebenen Ansätzen, die das Erreichen einer hohen und vorhersehbaren Endqualität durch explizite Methodik deutlich fördern.

Ob dies dann allerdings erreicht bzw. gelebt wird, ist eine Frage der richtigen Entscheidungen im Verlauf des Prozesses, die wiederum ausreichende Entscheidungsgrundlagen zu spezifischen Zeitpunkten benötigen. Hier setzt das Testen an, da es diese Informationen systematisch liefert. Die Frage, ob agile Prozesse Testen benötigen, kann also ganz klar mit „Ja!“ beantwortet werden. Mehr noch, sie fordern dies sogar. Denn nur durch eine zu den richtigen Zeitpunkten und in angemessener Frequenz verfügbare, verlässliche Aussage hinsichtlich der aktuellen Qualität des SUT (System under Test), kann eine Entscheidung bzgl. des Umfanges an neuer Funktionalität für einen Sprint getroffen und somit überhaupt erst das grundlegende Paradigma der Aufteilung der Entwicklung einer Software in im Umfang reduzierte aber bzgl. der Qualität fertiggestellte Zwischenprodukte realisiert werden. In den folgenden Abschnitten werden die Verfahren und Methoden zur Anpassung des Testens innerhalb von SCRUM gezeigt, damit es diese Anforderungen optimal erfüllen kann.

Der Quality-Knowledgeletter ist ein periodisches Informationsmedium von Software Quality Lab und dessen Partnern mit den Schwerpunkten IT-Qualitätsmanagement, Projekt- und Prozess-Management. Inhalt: fachliche Beiträge und Schwerpunktthemen, Vorstellung neuer Produkte und Leistungen, neue wissenschaftliche Erkenntnisse und andere Fachbeiträge aus unseren Themenbereichen. Aktuelle Fach- und Forschungsbeiträge sind willkommen. Einsendungen an info@software-quality-lab.com.

Weitere Infos zu diesem und anderen Themen finden Sie auf www.software-quality-lab.com.

Testen in agilen Prozessen am Beispiel SCRUM

Dipl.-Inf. Stephan Christmann - Software Quality Lab

Das Erreichen einer hohen Qualität bei der Entwicklung von Software muss auch bei agilen Prozessen und insbesondere bei SCRUM aktiv geplant und systematisch umgesetzt werden. Qualität entsteht auch in SCRUM nicht automatisch oder nur, weil die Entwicklung in Zwischenschritte aufgeteilt wird. In diesem Beitrag erhalten Sie einen Überblick über wichtige Aspekte der Qualitätssicherung, die bei der Ausrichtung auf SCRUM relevant und z.T. sogar absolut notwendig sind. Insbesondere wird ein Schwerpunkt auf die Einführungsphase von SCRUM gelegt.

Einleitung & Abgrenzung

Dieser Beitrag soll eine Hilfestellung bei der Einführung von SCRUM unter dem Aspekt der Qualitätssicherung und des Testens sein. Er soll ferner Argumentationshilfen geben, um dem Testen auch im SCRUM den richtigen Stellenwert zuzuordnen, Aufwände realistisch abzuschätzen und Ressourcen korrekt einzuplanen. Dazu werden zunächst die Potentiale in SCRUM hinsichtlich der Erreichung einer optimalen Qualität aufgezeigt. SCRUM bietet von seinem Aufbau her hohe Potentiale, eine Qualitätssicherung optimal umzusetzen und ist in einigen Punkten diesbezüglich sogar fordernd. Diese Potentiale zu zeigen, wird der Inhalt des ersten Abschnitts sein. In den folgenden Abschnitten wird auf die Potentiale näher eingegangen, um zu zeigen wie man diese auch nutzen kann. Der letzte Abschnitt wird das Bild abrunden, indem er sich von der anderen Seite der Thematik nähert und kurz z.T. einige Fallstricke aufzeigen wird, die es möglichst zu vermeiden gilt, wenn es um die Umsetzung von Qualitätssicherungsmaßnahmen und Testen im Besonderen innerhalb von SCRUM geht.

Eine schrittweise und erschöpfende Anleitung zur Umsetzung des Testens innerhalb von SCRUM würde den Rahmen dieses Artikels jedoch sprengen. Hier sei auf die umfangreiche Literatur am Markt verwiesen. Auch wird nicht auf die Unterschiede der verschiedenen agilen Ansätze eingegangen werden. Hier beschränkt sich der Artikel auf SCRUM als exemplarischen und am häufigsten verwendeten Ansatz im agilen Umfeld

SCRUM-Potentiale für Qualität

Wie eingangs erwähnt, ist SCRUM ein qualitätsgetriebener Prozess, der von sich aus und aufgrund seiner Methodik ein Umfeld bereitstellt, ja sogar fordert, in dem Qualität leichter entstehen kann als in anderen, nicht agilen Ansätzen. Warum ist das so? Dies soll im Folgenden anhand einiger wichtiger Aspekte der Qualitätssicherung kurz erläutert werden.

Kommunikation und optimale Wissensverteilung

Ein wesentlicher Erfolgsfaktor für eine hohe Endqualität der Software und eine gute Projektdurchführung ist ein hohes Maß an Kommunikation innerhalb des Projektteams. Nur durch entsprechend eingerichtete und gelebte Kommunikationsplattformen kann sichergestellt werden, dass Probleme frühzeitig erkannt und gemeinsam angegangen werden. Ein hohes Maß an Kommunikation verhindert Wissensinseln und fördert somit die gemeinsame Lösungsfindung innerhalb von

Teams. Dies ist für das Vorbereiten eines Umfelds, in dem gute Qualität gleich im Vorfeld entsteht, ganz wesentlich und gehört somit zu den Aspekten der konstruktiven Qualitätssicherung (Fehlervermeidung). Wo in anderen Projektmodellen entsprechende Plattformen erst eingerichtet und vor allem motiviert werden müssen („Kommt man vor lauter Meetings eigentlich auch zum Arbeiten?“), ist dies bei SCRUM bereits u.a. durch Daily-Meetings, Retrospektiven, Sprint-Reviews etc. vorstrukturiert und akzeptiert.

Integrierte Teams versus Abteilungsgrenzen

Qualität leidet immer dort, wo Zuständigkeiten und Abteilungsgrenzen zusätzliche Dokumentationen nötig machen und damit Spielraum für Interpretationen lassen. Im Umkehrschluss bedeutet dies, dass die direkte Kommunikation, das gemeinsame Planen und Umsetzen im Team mit einem gemeinsam aufgebauten Verständnis der Domäne sehr förderlich



für die Qualität sind. Grenzen zwischen Teams fördern zudem auch das Konfliktpotential, insbesondere wenn der Projektdruck steigt. Dies wiederum kann innerhalb des Projekts zu erheblichen Reibungsverlusten führen und die Qualität belasten. Man kennt dieses Phänomen bei Wasserfallmodellen, wenn „fertig gestellte“ Features erst einmal bewusst dem Test „rüber geworfen“ werden, damit sie aus den Augen sind, wohl wissend, dass noch erhebliche Mängel darin zu erwarten sind. Der Effekt ist, dass der Test sein Augenmerk viel zu stark auf offensichtliche Fehler lenken muss, anstatt tiefergehende Probleme systematisch untersuchen zu können. Eine mangelhafte Endqualität ist die Folge. In hochgradig integriert agierenden Teams, die gemeinsam auftreten, sind diese Gefahren deutlich reduziert. SCRUM bietet auch hier den entsprechenden Rahmen, dass Teams gemeinsam agieren. Tester und Entwickler gehören z.B. einem SCRUM-Team gemeinsam an und sehen sich gemeinsam verantwortlich für eine hohe Endqualität. Es gibt zwar auch unterschiedliche Rollen, die Erstellung der resultierenden Projektergebnisse wird aber als eine gemeinsame Aufgabe verstanden.

Vollständiger Knowledge Letter Zugang

Wir freuen uns, dass Sie an diesem Thema Interesse haben und den Knowledge Letter von Software Quality Lab bis hierher gelesen haben.



Dieser Knowledge Letter ist eine Vorschau (gekürzte Version des gesamten Artikels).

Wenn Sie den ungekürzten Knowledge Letter lesen möchten, registrieren Sie sich bitte unter <http://www.software-quality-lab.com/download/knowledge-letter/anfrage-knowledge-letter/>

Sie erhalten nach der Registrierung vollen Zugang zu allen bisherigen Knowledge Letters von Software Quality Lab und erhalten automatisch künftige Knowledge Letter per E-Mail.

Software Quality Days — Die größte Konferenz zum Thema „Software Qualität“ in Europa!



Besuchen Sie die Top-Konferenz mit allen Infos rund um Software Qualität.

Beste Qualität der Vorträge und Tutorials sowie eine Mischung aus praktischen und wissenschaftlichen Beiträgen machen die Software Quality Days zum Top-Event.

In den 3 praktischen Tracks werden anwendungsorientierte Vorträge präsentiert. Der wissenschaftliche Track zeigt Beiträge mit hohem Innovationsgrad und praktischer Anwendbarkeit, basierend auf Forschungsergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter

www.software-quality-days.com

