

# PQ4Agile – Steigerung der Produktqualität in agilen Projekten

Hartmut Schmitt<sup>1</sup>, Dominik Magin<sup>2</sup>, Dominik Rost<sup>2</sup>

HK Business Solutions GmbH<sup>1</sup>

Fraunhofer-Institut für Experimentelles Software Engineering IESE<sup>2</sup>

## Zusammenfassung

Agile Vorgehensweisen bieten hervorragende Unterstützung bei der Implementierung und beim Management von Softwareprojekten. Für die Erreichung nichtfunktionaler Produktqualitäten sind sie allerdings keine optimale Hilfestellung. Die Folge sind oft Mängel, die zu Lasten der Teamproduktivität und der Softwarequalität gehen. Zahlreiche Methoden aus dem Softwareengineering wären geeignet, um dieser Problematik zu begegnen, gelten unter Agilisten jedoch als zu schwergewichtig und zu starr. Das Projekt PQ4Agile schlägt eine Brücke zwischen beiden Ansätzen, indem es traditionelle Methoden und Praktiken des Softwareengineerings leichtgewichtiger gestaltet und gezielt für die Verwendung in agilen Projekten anpasst. Diese neu gestalteten Best Practices bieten agilen Entwicklern eine effiziente und systematische Unterstützung, um ein höheres Maß an Produktqualität zu erreichen.

## 1 Agilität & Softwarequalität

Agile Vorgehensweisen haben heute eine weite Verbreitung erreicht (Komus 2012), insbesondere bei kleinen und mittleren Softwareunternehmen, die sich durch die Einführung dieser Modelle mehr Flexibilität und Kundennähe erhoffen. Beliebte agile Techniken wie Pair Programming, Test Driven Development und Refactorings sorgen für eine effektivere Softwareentwicklung. Ein agiles Framework wie Scrum (Schwaber & Sutherland 2013) bietet mit Aktivitäten wie Sprint Planning, Daily Scrum und Sprint Review systematische Unterstützung beim Projektmanagement. Agile Vorgehensweisen leisten jedoch nur einen geringen Beitrag zur planvollen Erreichung nichtfunktionaler Produktqualitäten. Hierfür ist entweder sehr viel Teamerfahrung notwendig oder es bedarf einer zusätzlichen Hilfestellung. Fehlt beides, kann es bereits in frühen Entwicklungsphasen zu gravierenden Qualitätsmängeln kommen. Für deren Behebung sind zeitintensive Nacharbeiten notwendig, die zu Lasten der Produktivität gehen und die vor den Auftraggebern versteckt oder nachverhandelt werden müssen.

In verschiedenen Disziplinen des Softwareengineerings gibt es Methoden, Praktiken und Aktivitäten, die zur Steigerung der Softwarequalität entwickelt wurden. Diese Methoden sind wissenschaftlich validiert und bewähren sich seit vielen Jahren im Praxiseinsatz. Da sie vor

dem Hintergrund konventioneller Vorgehensmodelle entstanden sind, passen sie jedoch oft nicht zu den Paradigmen der Agilität, z. B. zur Entwicklung in kurzen Iterationszyklen, und werden daher von vielen Agilisten als zu schwergewichtig empfunden.

## 2 PQ4Agile – Projektidee und Vorarbeiten

Kernidee des vorgestellten Projekts „PQ4Agile – Produktqualität für Agile Softwareentwicklung“ ist es, etablierte Methoden aus den Bereichen Requirements Engineering, Usability Engineering und Softwarearchitektur gezielt für den Einsatz in agilen Vorhaben anzupassen, z. B. indem diese leichtgewichtiger gestaltet und bei Bedarf in Einzelteile zerlegt werden. Diese separat ausführbaren Best Practices können von den Mitgliedern agiler Entwicklungsteams nach Bedarf flexibel verwendet werden. Zusätzlich zu solchen adaptierten, vielfach erprobten Best Practices sollen bei Bedarf neue, innovative Best Practices entwickelt werden. Auf diese Weise entsteht ein umfassendes Kompendium von Praktiken, das zur Steigerung von nicht-funktionalen Produktqualitäten eingesetzt werden kann. Die Vorteile, von denen insbesondere kleine und mittlere Softwareunternehmen profitieren sollen, sind zum einen die planbare Erreichung von mehr Produktqualität, zum anderen eine Effizienzsteigerung, die sich z. B. aus der Vermeidung unnötiger Nacharbeiten ergibt. Beides soll zu einer höheren Erfolgswahrscheinlichkeit der realisierten Softwareprojekte beitragen.

Als Basis für die Arbeiten in PQ4Agile haben die Partner zunächst ein Qualitätsmodell entwickelt, das sämtliche Qualitätskriterien abdeckt, die für das Projekt relevant sind. Dieses Modell umfasst drei Qualitätsbereiche, bei denen jeweils ein bestehendes Qualitätsmodell als Grundlage diente: den Bereich Softwarequalität mit dem Modell der ISO 25010 (ISO/IEC 25010:2011-3), den Bereich Prozessqualität mit dem Modell „Gokyo Ri“ (Kneuper 2014) und den Bereich Strukturqualität mit dem Modell der ISO 9001 (DIN EN ISO 9001:2008). Im nächsten Schritt untersuchten die Partner, welche Integrationspunkte genutzt werden können, um die Best Practices in beliebige agile Entwicklungsprozesse einzubinden. Hierfür wurden die Entwicklungsprozesse der beteiligten Softwareunternehmen, verschiedene Prozesse aus der Software-Engineering-Literatur, das Scrum-Framework sowie das CMMI-DEV-Modell (CMMI Product Team 2011) untersucht. Ergebnis war ein agiler Referenzprozess, der in sechs Bereiche (Anforderungen, Planung und Design, Evaluation, Realisierung, Kontrolle sowie Projektplanung und -steuerung) unterteilt ist und insgesamt 22 Aktivitäten umfasst (Schmitt & Rost 2015).

## 3 Entwicklung eines Best-Practice-Kompendiums

Um agilen Entwicklern eine umfassende Unterstützung in allen relevanten Qualitätsbereichen und bei möglichst vielen Aktivitäten anbieten zu können, bedarf es einer Vielzahl von Best Practices aus unterschiedlichen Teilbereichen des Softwareengineerings. Ziel war es daher, ein Kompendium zu schaffen, das Methoden und Praktiken aus den Bereichen Requirements Engineering, Usability/User Experience und Softwarearchitektur umfasst. Um sicherzustellen,

dass diese Best Practices schnell und dynamisch anwendbar sind und mit agilen Vorgehensweisen kombiniert werden können, wurden für deren Auswahl sieben Prinzipien definiert: Minimalismus/Simplizität, Personennähe, Artefaktnähe, Timing, Skalierbarkeit, Änderungsaffinität und Integrierbarkeit (Schmitt & Rost 2015).

### 3.1 Auswahl existierender Best Practices

Zunächst untersuchten die Partner über 250 Methoden, Praktiken und Aktivitäten aus der Literatur. Nach der Bereinigung von Dubletten wurde eine Vorauswahl getroffen: Vorgehensweisen, die den definierten Prinzipien offensichtlich widersprechen und unter Agilisten eher als Anti-Patterns gelten, z. B. Big Design Up Front (detailliertes Festlegen der Architektur zu Beginn des Projekts), wurden von der weiteren Betrachtung ausgeschlossen. Für die Ausarbeitung verblieben circa 80 erfolgversprechende Kandidaten, die im Detail betrachtet wurden. Bei der Auswahl der Best Practices wurden konstruktive Methoden und Praktiken bevorzugt, die in möglichst frühen Phasen des Entwicklungsprozesses zur Anwendung kommen und die einen unmittelbaren und schnellen Beitrag zur Qualitätssteigerung leisten. Best Practices, die sich durch eine hohe Effizienz auszeichnen und redundante Arbeiten vermeiden, wurden ebenfalls priorisiert.

Beispielsweise wurde das kontextuelle Benutzerinterview (engl. Contextual inquiry, Holzblatt et al. 2004) als Best Practice ausgewählt, da es eine hohe Übereinstimmung mit dem Prinzip der Personennähe aufweist und besonders effizient ist. Die Best Practice verbindet ein halbstrukturiertes Interview mit einer teilnehmenden Beobachtung des Anwenders in dessen (Arbeits)kontext. Die so gewonnenen Informationen können in Form von Personas, funktionalen und nichtfunktionalen Anforderungen oder Beschreibungen des Ist-Zustands weiterverarbeitet werden. Mit etwas Vorbereitung und Übung kann diese Form des Interviews von einem beliebigen Mitglied des agilen Entwicklungsteams durchgeführt werden. Da sie außerhalb von Usability-Expertenkreisen nicht sehr bekannt ist (Ross 2012), stellt die Dokumentation im Rahmen des Projekts zudem einen Mehrwert für die anvisierte Zielgruppe dar.

### 3.2 Neuentwicklung von Best Practices

Neben bereits etablierten Methoden und Praktiken wurden auch erfolgversprechende neue Ansätze der Projektpartner untersucht, die geeignet sind, zu einer Steigerung der Produktqualität in agilen Prozessen beizutragen. Beschrieben wurden insbesondere Lösungsansätze für die Bereiche Usability/User Experience (z. B. Produktphilosophie erstellen, Template-basierte UI-Konzeption) und Softwarearchitektur (z. B. Kontinuierliche Architekturbewertung durchführen). Außerdem wurden zahlreiche aktuelle Vorschläge aus der wissenschaftlichen Community und der Praxis berücksichtigt. Beispielsweise wurde im Rahmen des UP14-Workshops „UX4Agile“ (Hess et al. 2014) erhoben, welche Herausforderungen bei agilen Entwicklungsprozessen in Bezug auf traditionelle Usability-Aktivitäten bestehen und welche neuartigen Usability-Praktiken die Teilnehmer bereits in ihren Projekten erfolgreich einsetzen.

### 3.3 Dokumentation und Veröffentlichung der Best Practices

Zielgruppe der Best Practices sind die Mitglieder agiler Entwicklungsteams bei kleinen oder mittelständischen Softwareherstellern, also keine Softwareengineering-Experten. Daher haben die Partner bei der Dokumentation der Best Practices darauf geachtet, dass die Beschreibungen einen Novizen nicht durch ihre Komplexität einschüchtern und dass die Best Practices auch ohne Expertenwissen möglichst einfach anwendbar sind (für Anfänger gegebenenfalls nach einer kurzen Einweisung durch einen erfahreneren Teamkollegen). Ferner wurde Wert darauf gelegt, dass die Anwendung der Best Practices keine aufwändigen Prozessanpassungen erfordert und möglichst ökonomisch ist (z. B. Anzahl der benötigten Probanden bei Tests).

Das Kompendium umfasst rund vierzig Best Practices und steht kostenlos im Projektportal [www.pq4agile.de](http://www.pq4agile.de) zur Verfügung. Die Partner wenden die Best Practices aktuell in Pilotprojekten an, so dass im weiteren Verlauf des Projekts überprüft werden kann, ob sich messbare Effekte, also Qualitätsverbesserungen im Entwicklungsprozess und bei den resultierenden Softwareprodukten, einstellen.

#### Danksagung

Das Projekt PQ4Agile wird vom Bundesministerium für Bildung und Forschung im Rahmen der Fördermaßnahme KMU-innovativ: IKT gefördert (Kennzeichen 01IS13032).

#### Literaturverzeichnis

- CMMI Product Team (2011): *CMMI für Entwicklung, Version 1.3*. Carnegie Mellon University, Pittsburgh
- DIN EN ISO 9001:2008 Qualitätsmanagementsysteme – Anforderungen. Beuth, Berlin.
- Hess, S., Rost, D. & Schmitt, H. (2014): UX4Agile – Integration von Best Practices des Usability- und User-Experience-Engineering in agile Entwicklungsprozesse. In: German UPA (Hrsg.): Tagungsband Usability Professionals 2014
- Holtzblatt, K., Burns Wendell, J., Wood, S. (2004): *Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design*. Morgan Kaufmann, San Francisco
- ISO/IEC 25010:2011-3 Software-Engineering - Qualitätskriterien und Bewertung von Softwareprodukten (SQuaRE) – Qualitätsmodell und Leitlinien. Beuth, Berlin.
- Kneuper, R. (2014): Gokyo Ri: Messung und Bewertung von Prozessqualität. Verfügbar unter: <http://www.kneuper.de/GokyoRi/>
- Komus, A. (2012): Studie: Status Quo Agile Verbreitung und Nutzen agiler Methoden. Ergebnisbericht (Langfassung). Hochschule Koblenz, Koblenz
- Ross, J. (2012): Why Are Contextual Inquiries So Difficult? Verfügbar unter: <http://www.uxmatters.com/mt/archives/2012/06/why-are-contextual-inquiries-so-difficult.php>
- Schmitt, H. & Rost, D. (2015). Qualitätssteigerung durch Best Practices. In: Software und Support Media GmbH (Hrsg.): *Entwickler Magazin Ausgabe 2015 (2)*, S. 82-86. Software & Support, Frankfurt am Main
- Schwaber, K. & Sutherland, J. (2013): *Scrum Guide - Der gültige Leitfaden für Scrum: Die Spielregeln*. Verfügbar unter: <http://www.scrumguides.org/download.html>