Doppelt hält besser

Mit UUX-Best-Practices agile Projekte zum Erfolg führen

Hartmut Schmitt

HK Business Solutions GmbH
Mellinweg 20
66280 Sulzbach
schmitt@hk-bs.de

Steffen Hess

Fraunhofer IESE
Fraunhofer-Platz 1
67663 Kaiserslautern
steffen.hess@iese.fraunhofer.de

Dominik Pascal Magin

Fraunhofer IESE
Fraunhofer-Platz 1
67663 Kaiserslautern
dominik.magin@iese.fraunhofer.de

Dominik Rost

Fraunhofer IESE
Fraunhofer-Platz 1
67663 Kaiserslautern
dominik.rost@iese.fraunhofer.de

Abstract

Gerade wegen ihrer Kundennähe, Prozesseffizienz und des flexiblen Reagierens auf Anforderungen erfreuen sich agile Entwicklungsvorgehen einer großen Beliebtheit. Agile Entwicklungsmethoden fokussieren meist auf funktionale Korrektheit. Daher bedarf es zur Erreichung nichtfunktionaler Produktqualitäten wie Gebrauchstauglichkeit oder Nutzungsqualität in agilen Projekten entweder sehr viel Erfahrung oder einer zusätzlichen Hilfestellung. Das Forschungsprojekt PQ4Agile erarbeitet eine effiziente und systematische Unterstützung agiler Teams bei der Durchführung von Usability- und User-Experience(UUX)-Aktivitäten (sog. Best Practices). Hierfür werden klassische UUX-Methoden an agile Entwicklungsvorgehen angepasst, aber auch neue, erfolgversprechende Ansätze entwickelt. Zwei neu erarbeitete UUX-Best-Practices stellen wir in diesem Beitrag im Detail vor.

Keywords

Best Practices, Agile Entwicklung, Scrum, Software-Engineering-Methoden

Motivation

Agile Softwareentwicklung ist zum führenden Entwicklungsansatz in vielen Domänen und Unternehmen geworden (Komus 2012). Angesichts der unbestreitbaren Vorteile agiler Vorgehen ist dies nicht verwunderlich: schnelles und stetiges Ausliefern nutzbarer (Teil-)Funktionalitäten durch kurze Entwicklungszyklen, Kundenzufriedenheit durch enge

Interaktion und kontinuierliche Anpassung der Anforderungen und Zufriedenheit der Entwickler durch eigenverantwortliches und selbstbestimmtes Handeln sind Ziele, die zahlreiche Unternehmen erreichen wollen.

In der Vergangenheit gab es viele Diskussionen darüber, ob konzeptionelle Tätigkeiten wie Usability- und User-Experience(UUX)-Engineering Platz in agilen Entwicklungsprozessen haben und ob es hier nicht eine grundsätzliche Inkompatibilität gibt. Diese Diskussionen sind mittlerweile mehrheitlich vorüber: Ein Framework wie Scrum (Schwaber & Sutherland 2013) bietet mit Aktivitäten wie Sprint Planning, Daily Scrum und Sprint Review systematische Unterstützung beim Projektmanagement, agile Techniken wie Pair Programming, Test Driven Development und Refactoring tragen zu einer effektiven Implementierungsarbeit bei. Eine Unterstützung zur systematischen Erreichung von Produktqualitäten wie Gebrauchstauglichkeit oder Nutzungsqualität fehlt jedoch gänzlich. In Projektsituationen mit in der Domäne und Technologie erfahrenen Entwicklern und einem System von übersichtlicher Größe und Lebenszyklus wird dieser Mangel durch die Erfahrung des Teams kompensiert. In allen anderen Situationen sind jedoch häufig frühe Qualitätsmängel die Folge, die durch aufwendige Nacharbeiten behoben werden müssen.

Folglich hat sich die Auffassung durchgesetzt, dass nicht-triviale Projektsituationen zusätzliche, dedizierte Engineering-Aktivitäten erfordern und dass es sich dabei keineswegs um einen Widerspruch handelt. Wie jedoch die Integration von UUX-Engineering in agiler Entwicklung zu gestalten ist, ist noch immer eine offene Frage.

Lösungsidee

Viele UUX-Techniken – z. B. die Durchführung von kontextuellen Benutzerinterviews, die Verwendung von Usability-Patterns oder Usability-Inspektionen – bieten genau diese erwünschte, zusätzliche Unterstützung zur systematischen Erreichung von Usability und User Experience. Solche Techniken wurden vielfach in der Praxis eingesetzt, sind empirisch validiert und werden kontinuierlich weiterentwickelt. Um diese gewinnbringend in agilen Entwicklungsprojekten einsetzen zu können, müssen sie jedoch auf die Spezifika von agilen Prozessen wie kurze Iterationszyklen oder enge Interaktion mit Stakeholdern angepasst werden.

Unsere Kernidee ist es, etablierte und zusammenhängende Methoden des UUX-Engineering in einzelne Bausteine zu zerlegen, die von Entwicklern in agilen Entwicklungsprojekten selbstbestimmt ausgewählt und effizient eingesetzt werden können – ohne die Notwendigkeit, den Entwicklungsprozess gänzlich umgestalten zu müssen. Um dies zu erreichen, entwickeln wir Best Practices, mit denen agile Entwicklungsprozesse angereichert werden können. Das Forschungsprojekt PQ4Agile bietet uns dafür den entsprechenden Rahmen. Dabei haben wir nicht nur Aktivitäten aus dem UUX-Bereich sondern auch aus dem Requirements Engineering und der Softwarearchitektur im Blick (Schmitt & Rost 2015). Abbildung 1 visualisiert diese Idee.

Abbildung 1: Kernidee – Erweiterung von agilen Entwicklungsprozessen mit Best Practices

Das Ergebnis des Projektes ist ein Kompendium mit Software-Engineering-Best-Practices aus den Bereichen UUX, Requirements Engineering und Softwarearchitektur. Aus diesem Kompendium können sich Entwickler nach Bedarf die vielversprechendsten Best Practices für ihr agiles Entwicklungsprojekt auswählen. Wir haben sieben Prinzipien definiert, denen die Best Practices folgen sollen, damit sie kompatibel zu agiler Entwicklung sind:

Minimalismus/Simplizität: Eine Software-Engineering-Aktivität soll nur so umfangreich ausgeführt und deren Ergebnisse nur so umfangreich dokumentiert werden, wie durch die gegebene Projektsituation notwendig ist, damit eine effiziente Weiterbearbeitung und Wartung möglich wird.

Personennähe: Die Verfügbarkeit und enge Zusammenarbeit mit Stakeholdern des Kunden und von Teammitgliedern untereinander soll durch Software-Engineering-Aktivitäten genutzt werden können.

Artefaktnähe: Software-Engineering-Artefakte sollen so produziert und verfügbar gemacht werden, dass sie möglichst effizient von Nachfolgetätigkeiten verwendet werden können und möglichst nahe am Zielprodukt sind.

Timing: Software-Engineering-Aktivitäten sollen immer genau bzw. spätestens dann ausgeführt werden, wenn es für den Projektfortschritt am nützlichsten und effizientesten ist.

Skalierbarkeit: Bei einer Zunahme von Projektfaktoren wie Stakeholdern, Systemkomplexität oder Teamgröße müssen Software-Engineering-Aktivitäten weiterhin so ausführbar sein, dass sie nicht den Grundprinzipien von agilen Vorgehen widersprechen und dennoch genügend Unterstützung zur Erreichung von Qualitätsattributen bieten.

Änderungsaffinität: Häufige Änderungen werden von agilen Vorgehen als essentieller Teil der Projektdurchführung gesehen. Software-Engineering-Aktivitäten sollen Änderungen ebenso begreifen und Änderungen, die die Bereiche User Experience Engineering, Requirements Engineering und Softwarearchitektur betreffen, effizient ermöglichen und Stakeholder dabei unterstützen.

Integrierbarkeit: Die Software-Engineering-Aktivitäten müssen so beschaffen sein, dass sie eine möglichst nahtlose Integration in bestehende agile Entwicklungsprozesse erlauben.

PQ4Agile – Produktqualität für Agile Softwareentwicklung

Seit November 2013 widmet sich das Forschungsprojekt „PQ4Agile – Produktqualität für Agile Softwareentwicklung“ der Aufgabe, die in der Lösungsidee beschrieben ist. Wichtigstes Ziel des Projekts ist die Erstellung von Software-Engineering-Best-Practices, die agile Entwickler bei der Erreichung nichtfunktionaler Qualitätsmerkmale unterstützen sollen – möglichst effektiv, mit wenig Aufwand und idealerweise ohne dass für die Anwendung der Best Practices Expertenkenntnisse notwendig sind. Als Basis für die Arbeiten in PQ4Agile haben wir zunächst ein Qualitätsmodell entwickelt, das ein gemeinsames Qualitätsverständnis für agile Ansätze und empirisches Software Engineering schafft. Dieses Qualitätsmodell deckt alle für PQ4Agile relevanten Qualitätskriterien ab: die funktionalen und nichtfunktionalen Anforderungen an das Softwareprodukt (Bereich Softwarequalität), die Aktivitäten des Entwicklungsprozesses (Bereich Prozessqualität) sowie strukturelle Aspekte im Unternehmen (Bereich Strukturqualität). Für jeden dieser drei Bereiche diente ein bestehendes Qualitätsmodell als Grundlage: das Modell der ISO 25010 (ISO/IEC 25010:2011-3) für den Bereich Softwarequalität, das Modell „Gokyo Ri“ (Kneuper 2014) für den Bereich Prozessqualität und das Modell der ISO 9001 (DIN EN ISO 9001:2008) für den Bereich Strukturqualität.

Im nächsten Schritt haben wir untersucht, welche Anknüpfungspunkte geeignet sind, um die Best Practices in beliebige agile Entwicklungsprozesse zu integrieren. Da Prozessanpassungen vermieden werden sollen, mussten möglichst allgemeingültige und generalisierbare Anknüpfungspunkte identifiziert werden. Hierzu wurden das Scrum-Framework, das CMMI-DEV-Modell (CMMI Product Team 2011), verschiedene Prozesse aus der Software-Engineering-Literatur sowie die Entwicklungsprozesse der an PQ4Agile beteiligten Softwareunternehmen analysiert. Ergebnis ist ein agiler Referenzprozess, dessen 22 Aktivitäten (z. B. „Lösungskonzept entwickeln“, „Oberflächenentwurf erstellen“, „Prototypen erstellen“ und „Reviews durchführen“) als Anknüpfungspunkte für die Best Practices geeignet sind. Dieser Referenzprozess ist in sechs Bereiche unterteilt: „Anforderungen“, „Planung und Design“, „Evaluation“, „Realisierung“, „Kontrolle“ sowie „Projektplanung und -steuerung“. Die Bereiche und die in ihnen enthaltenen Aktivitäten werden während des Entwicklungsprozesses in der Regel iterativ ausgeführt.

Entwicklung und Dokumentation der Best Practices

Ziel des Projekts PQ4Agile ist es, ein Kompendium von Best Practices zu schaffen, das für agile Entwickler möglichst effizient nutzbar ist und eine umfassende Unterstützung in allen relevanten Qualitätsbereichen bietet. Zudem sollen die Entwickler möglichst bei allen relevanten Aktivitäten, die im Referenzprozess beschrieben sind, eine Hilfestellung in Form von Best Practices erhalten. Um die erarbeiteten Best Practices in geeigneter Form dokumentieren zu können, wurde ein generisches Beschreibungstemplate entwickelt, das ein

Mapping der Best Practices mit den Qualitätsaspekten des Qualitätsmodell und den Aktivitäten des Referenzprozesses erlaubt. Alle Best Practices wurden so ausgestaltet und dokumentiert, dass für die Anwendung keine aufwändigen Prozessanpassungen notwendig sind; hierdurch soll die Hürde zur Anwendung der Best Practices bewusst niedrig gehalten werden.

Aus dem UUX-Bereich haben wir mehr als 130 etablierte Methoden und Praktiken untersucht. Nach der Bereinigung von Dubletten und einer ersten Vorauswahl verblieben circa 25 erfolgversprechende Kandidaten für die weitere Ausarbeitung (z. B. „Kontextuelles Benutzerinterview durchführen“, „Personas erstellen“, „Informationsarchitektur erstellen“, „Usability-Review durchführen“ und „Usability-Test durchführen“). Diese Praktiken haben wir bei Bedarf angepasst, so dass sie relativ leicht in agile Prozesse integriert werden können und auch von Usability-Novizen angewendet werden können. Berücksichtigt wurden aber auch erfolgversprechende neue Ansätze der Projektpartner, die zu einer Steigerung der Produktqualität in agilen Prozessen beitragen sollen.

Außerdem fanden aktuelle Vorschläge aus der Usability-Community Eingang. Beispielsweise haben wir im Rahmen der Konferenz Usability Professionals 2014 den Workshop „UX4Agile – Integration von Best Practices des Usability- und User-Experience-Engineering in agile Entwicklungsprozesse“ (Hess et al. 2014) durchgeführt. Hier haben wir gemeinsam mit den Teilnehmern untersucht, welche Best Practices des Usability-Engineerings heute bereits erfolgreich in agilen Projekten eingesetzt werden. Zudem haben wir in dem Workshop erarbeitet, welche Defizite und Herausforderungen in Bezug auf UUX-Aktivitäten in agilen Entwicklungsprozessen momentan bestehen und wie wir durch pragmatischere Vorgehensweisen und eine stärkere Einbindung der Entwickler Verbesserungen herbeiführen können (vgl. Abbildung 2). Auf diese Weise wurde wertvolles Feedback gewonnen, das sowohl bei der Anpassung etablierter UUX-Aktivitäten als auch der Beschreibung neuartiger Best Practices (z. B. „Usability-Patterns verwenden“) zum Tragen kam.

Abbildung 2: UX4Agile – Workshopergebnisse

Best-Practice-Beispiele

Wie oben dargestellt haben wir im Zuge von PQ4Agile existierende Best Practices angepasst und basierend auf den Erfahrungen der Projektpartner neue Best Practices entwickelt. Im Folgenden werden zwei neue UUX-Best-Practices vorgestellt.

„Templatebasierte UI-Gestaltung“

Das Aussehen der Benutzungsoberfläche (engl.: user interface, kurz: UI) einer Softwareanwendung unterliegt während der Entwicklung häufigen Änderungen, die prototypisch skizziert und als ausführbarer Code implementiert werden. Diese Änderungen haben nicht nur Einfluss auf einzelne Teile des UIs, sondern können Auswirkungen auf das komplette UI-Konzept haben. Konsistente Änderungen sind in der Regel sehr zeitintensiv und fehleranfällig, da diese Änderungen in allen betroffenen Artefakten nachgezogen werden müssen. Es muss nicht nur eine potenziell große Anzahl an Prototypen angepasst werden, sondern es müssen auch viele Änderungen am bereits implementierten Code vorgenommen werden. Eine Verfolgbarkeit von prototypisch skizzierten UI-Elementen zu deren Umsetzung im Code und die Verfolgung von Änderungen innerhalb von Prototypen über die Zeit hinweg sind dabei üblicherweise nicht möglich oder werden vernachlässigt. Eine schnelle und konsistente Umsetzung von Änderungswünschen bezüglich des UI-Designs ist dadurch nur schwer möglich.

Die Best Practice „Template-basierte UI-Gestaltung“ dient der Erstellung und Verwendung von Vorlagen, die bei der Gestaltung der Benutzeroberfläche zum Einsatz kommen. Diese Vorlagen definieren das Grundgerüst einer Anwendung durch instanziiierbare UI-Elemente, die innerhalb der Anwendung verwendet werden, z. B. Schaltflächen, Tabellen, Header und Navigationsmenüs. Hierbei wird eine hierarchische Struktur dieser Elemente festgelegt, zum einen um die Zuordnung einzelner Elemente zu übergeordneten Elementen zu kennen, zum anderen um die Einflüsse von Änderungen bestimmter Elemente auf untergeordnete, von ihnen abhängige Elemente von vornherein festzulegen und zu kennen. Weiterhin werden sämtliche Attribute (vgl. Abbildung 3: Content Area) jedes einzelnen Elements spezifiziert, damit bei dessen Implementierung keine un spezifizierten oder nichtspezifizierten Aspekte auftreten. Abbildung 3 zeigt ein Template vom Aufbau einer Webanwendung.

Abbildung 3: Template einer Webanwendung

Sämtliche Dialoge der entwickelten Anwendung werden unter ausschließlicher Verwendung dieser Vorlagen gestaltet. Ähnlich einem Styleguide können somit die Elemente für ein einzelnes Projekt, für eine bestimmte Produktlinie oder für das gesamte Unternehmen spezifiziert werden. Dadurch wird nicht nur die Erstellung konsistenter Prototypen erleichtert, sondern es wird zugleich die Nachverfolgbarkeit von prototypisch instanziierten UI-Elementen zu ausführbarem Code ermöglicht. Abbildung 4 zeigt eine konkrete Instanz des in Abbildung 3 dargestellten Templates. Da die Erstellung einer Hierarchie der UI-Elemente zunächst zeitaufwändig ist, eignet sich diese Best Practice vor allem für größere Entwicklungsprojekte mit mehr als zwanzig verschiedenen Templates. Auch wenn die Vorteile der Best Practice bereits in kleinen Entwicklungsprojekten greifen, könnte der Aufwand für die Erstellung der UI-Element-Hierarchie hier in einem ungünstigen Verhältnis zu dem dadurch erzeugten Nutzen stehen, da ein beträchtlicher Teil des Gesamtbudgets in die Erstellung der Hierarchie fließen könnte. Das Nachziehen von Änderungen und die damit verbundene konsistente Gestaltung von Prototypen sind in kleinen Entwicklungsprojekten noch relativ einfach manuell durchzuführen. In großen Entwicklungsprojekten hingegen ist der Einsatz der Best Practice sehr viel gewinnbringender, da hier nicht nur mit einer wesentlich höheren Anzahl verwendeter UI-Elemente zu rechnen ist, sondern durch die

größere Anzahl erstellter Prototypen und die größere Menge implementierten Codes eine Nachverfolgung von Änderungen sowie eine konsistente Erstellung von Prototypen schwerer möglich ist.

Abbildung 4: konkrete Instanz eines Templates

„Produktphilosophie erstellen“

Die Best Practice „Produktphilosophie erstellen“ unterstützt die Definition und Bewertung der User Experience einer Softwareanwendung. Sie wird im Rahmen agiler Projekte hauptsächlich zur Entscheidungsunterstützung bzw. Priorisierung durch das Team bzw. den Product Owner verwendet. Dabei wird insbesondere der Trade-off zwischen funktionalen Features und User-Experience-Features aufgelöst. Abbildung 5 zeigt das Mobile Template der Produktphilosophie, bei dem es vordefinierte Attribute gibt, auf die man sich in den ersten Sprints einigt. Dahinter steckt jeweils eine Erklärung der Bewertung bzw. die Unterstützung durch Beispiele. Die Wertepaare und die Bewertung geben hierbei an, welchen Eindruck die Software vermittelt bzw. vermitteln soll. Beispielsweise besagt die Produktphilosophie in Abbildung 5, dass das zu entwickelnde Produkt einem individuellen Look & Feel folgen soll. Dies bedeutet, dass es bei Entscheidungen hinsichtlich des

Interaktionskonzepts und der Benutzeroberfläche eher ein individuelles Look & Feel (z.B. Company CI/Guidelines) zu verfolgen gilt. Der Entwickler erhält so eine Idee, wie die genannten Aspekte zu berücksichtigen sind.

Die Best Practice „Produktphilosophie erstellen“ verbessert die Kommunikation zwischen UI-Verantwortlichen, Vertrieb und Entwicklern. Insbesondere unterstützt sie Entwicklerteams dabei, systematisch eine spezifische User Experience in das fertige Produkt zu „injizieren“. Um dies zu erreichen, können z. B. einzelne User Stories auf die Attribute gemappt werden. Einen besonderen Nutzen erzielen Unternehmen, wenn sie die vorhandenen Templates vor der Verwendung auf ihre konkreten Bedürfnisse anpassen. Im Rahmen von PQ4Agile sollen verschiedene weitere Templates erstellt werden, die den unterschiedlichen Anwendungsarten (z. B. Mobile, Security, Software Renovation) gerecht werden.

Produktphilosophie Mobile Template

Verbindend						Gekapselt
Plattform Look & Feel						Individuelles Look & Feel
Fancy						Klassisch
Fokussiert						Spielerisch
Generisch						Persönlich
Nutzerzentriert						Technologisch
Bezaubernd						Sicher
Informierend						Einbeziehend

Abbildung 5: Produktphilosophie Mobile Template

Ausblick

Aktuell werden die entwickelten Best Practices in Pilotprojekten der PQ4Agile-Partner angewendet. Im weiteren Verlauf des Vorhabens wird empirisch überprüft, ob sich durch die Verwendung der Best Practices messbare Effekte, also Verbesserungen im Entwicklungsprozess und bei den resultierenden Softwareprodukten, einstellen. Im Zuge dieser Evaluation haben wir einen Fragebogen entwickelt, um die Qualität des Produkts und des Entwicklungsprozesses zu ermitteln. Es handelt sich bei den Evaluationsfragen um Items

auf Basis von Standards (z. B. ISO 25010) und etablierten Theorien (z. B. Hassenzahl et al. 2003), die nach den drei Qualitätsbereichen (Softwarequalität, Prozessqualität, Strukturqualität) klassifiziert sind. Die Bewertung durch den Fragebogen erfolgt zu drei Zeitpunkten: vor Einführung der Best Practices, nach Einführung und Anwendung eines ersten Sets von Best Practices und zu Projektende, nach Einführung eines zweiten Sets von Best Practices. Somit können Einflüsse durch Anwendung der Best Practices auf die Produkt- und Prozessqualität festgestellt und der Wert der Best Practices bestimmt werden.

Um die dokumentierten Best Practices einem möglichst großen Entwicklerkreis zugänglich zu machen, werden diese kostenlos und diskriminierungsfrei im Projektportal zur Verfügung gestellt. Unterstützend wird ein Wiki aufgebaut, in dem Softwareentwickler verschiedene Einstiegspunkte finden, um nach geeigneten Best Practices zu suchen (z. B. die Aktivitäten und Rollen des agilen Referenzprozesses oder die Qualitätsmerkmale und -teilmerkmale des Qualitätsmodells). Die Entwickler werden damit in die Lage versetzt, verschiedene Best Practices für ihre eigenen Entwicklungsprojekte auszuwählen und ohne großen Aufwand auszuprobieren. Das Wiki soll zudem die Möglichkeit zum Erfahrungsaustausch und zur Diskussion der Best Practices bieten, so dass Erfahrungen, die Entwickler mit der Anwendung der Best Practices gemacht haben, in die Best Practices zurückgespiegelt werden können.

Danksagung

Das Projekt „PQ4Agile – Produktqualität für Agile Softwareentwicklung“ wird vom Bundesministerium für Bildung und Forschung im Rahmen der Maßnahme KMU-innovativ: IKT – Softwaresysteme und Wissenstechnologien gefördert. Mehr Informationen: www.pq4agile.de

Literatur

- CMMI Product Team (2011): *CMMI für Entwicklung, Version 1.3*. Carnegie Mellon University, Pittsburgh
- DIN EN ISO 9001:2008 Qualitätsmanagementsysteme – Anforderungen. Beuth, Berlin.
- Hassenzahl, M., Burmester, M. & Koller, F. (2003): *AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität*. In J. Ziegler & G. Szwillus (Hrsg.): *Mensch & Computer 2003: Interaktion in Bewegung*, S. 187-196. Teubner, Stuttgart
- 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*
- 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

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>