Backlog Refinement – das Requirements Engineering (RE) der agilen Produktverantwortlichen

Andreas Becker



Agile Vorgehensweisen (z. B. Scrum) treffen keine expliziten Aussagen über Aktivitäten des Requirements Engineerings (RE), sondern fokussieren sich auf empirische Prozesssteuerung, agile Werte und eine minimalistische Erstellung von prozessbegleitenden Artefakten. Es kann der Eindruck erweckt werden, dass RE-Aktivitäten in diesen Umfeldern nicht notwendig sind. In der Praxis zeigt sich jedoch, dass viele agile Produktverantwortliche (z. B. Product Owner) bei ihrer täglichen Arbeit am Product Backlog überfordert sind – vor allem in agil skalierten Umfeldern.

3-3-5 + Empirische Prozesssteuerung

Scrum ist keine vollständige Methode, sondern ein Framework, um in komplexen Umfeldern, Produkte oder Systeme entwickeln zu können. Der Scrum-Guide [SS13] beschreibt in seiner aktuellen Fassung lediglich drei Rollen (PO, ScrumMaster und Entwickler), drei Artefakte (Product Backlog, Sprint Backlog und einem potentiell auslieferbarem Produktinkrement) sowie fünf Events (Sprint Planning, Sprint, Daily, Review, Retrospektive). Außerdem wird auf die empirische Prozesssteuerung deutlich hingewiesen, die in diesem Zusammenhang besagt, dass eine möglichst hohe Transparenz sicherzustellen ist, um die Arbeitsweisen und Ergebnisse des Teams beobachten und kontinuierlich anpassen zu können. Der Scrum-Guide trifft keine Aussagen über das WIE – also wie ein Produkt zu entwickeln ist. Es gibt demnach auch keinen vorgegeben Weg, wie das Requirements Engineering (RE) in Scrum zu integrieren ist. Es könnte sogar der Eindruck erweckt werden, dass ein Requirements Engineering in der agilen Entwicklung nicht notwendig ist, da es nicht explizit erwähnt wird.

Agile Projekte sind erfolgreicher

Die Standish Group, die seit 1985 weltweit Projekte auswertet, stellt in ihrem jährlich erscheinenden Chaos Manifesto fest, dass 2012 nur 39 % aller Entwicklungsprojekte erfolgreich abgeschlossen wurden. In den Jahren davor war die Anzahl der erfolgreichen Projekte noch geringer. Die Auswertung der erfolgreichen Projekte hat ergeben, dass direkte Anwenderbeteiligung, die Einführung von agilen Prozessen und die Nutzung von User Stories als Kommunikationsmittel und Anforderungs-Items einen maßgeblichen Anteil für den Erfolg hatten [StG13].

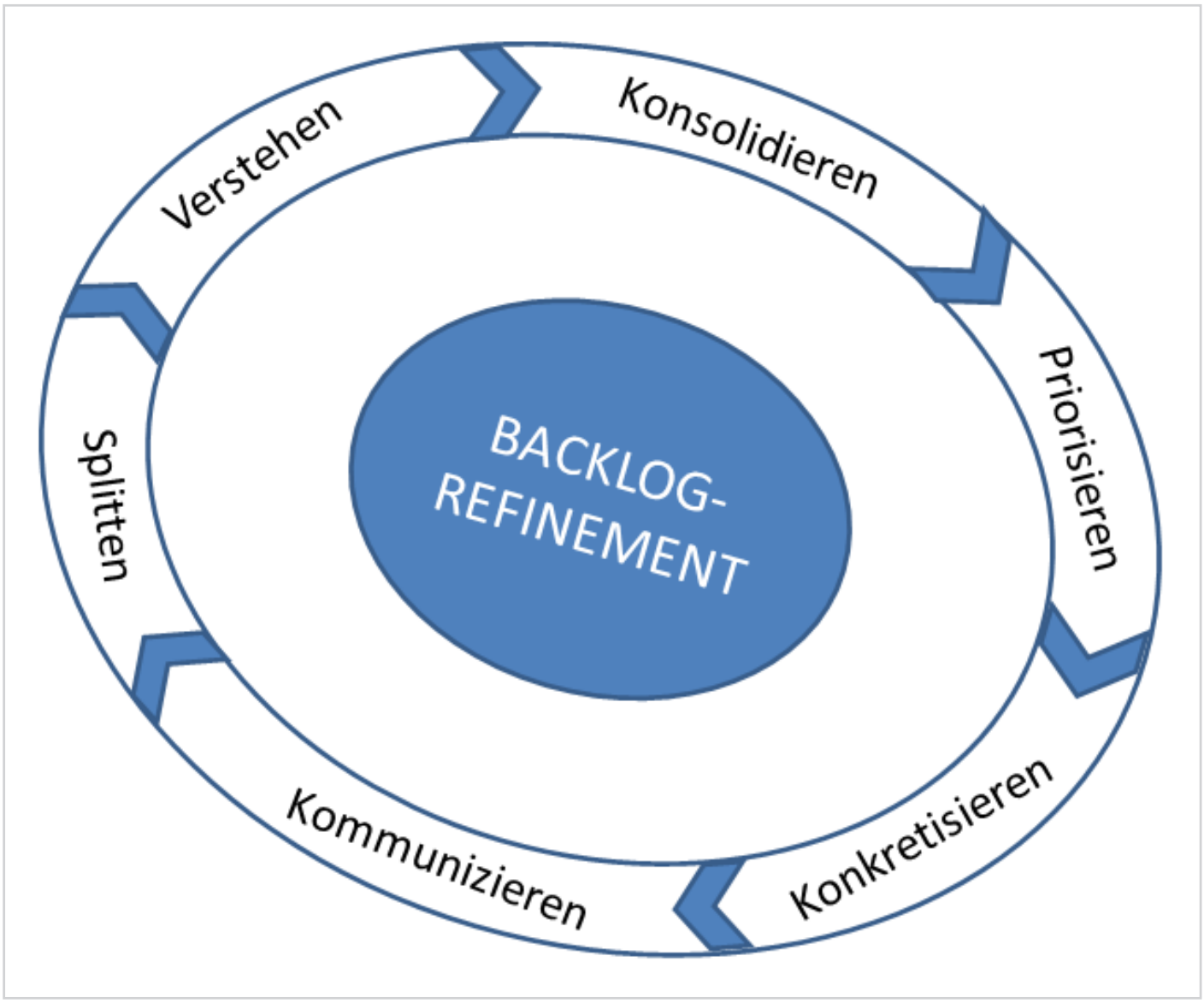
Agile Frameworks verändern das Requirements Engineering nachhaltig. Während in dokumentengetriebenen oder in phasenorientierten Vorgehensmodellen schriftlich verfasste Spezifikationen einen hohen Stellenwert aufweisen, konzentriert sich im agilen Umfeld die Arbeit an und mit den Anforderungen i. d. R auf User Stories nach dem Konzept von Mike Cohn [Coh10] oder Use Cases 2.0 nach Ivar Jacobson [JSB11]. Mit beiden Ansätzen kann den Prinzipien des Agile Manifest gerecht zu werden, das

eine direkte Kommunikation von Angesicht zu Angesicht als effizienteste und effektivste Art der Zusammenarbeit fordert [AMa01].

„Agile Waschtrommel“

Die kontinuierliche Tätigkeit des Product Owners „dreht“ sich rund um das Backlog-Refinement mit immer wiederkehrenden RE-Aktivitäten, die sich „im Rundlauf“ befinden. Aus diesem Grunde lassen sich diese Aktivitäten visuell darstellen und sind mit einer Waschtrommel (s. Abb. unten) vergleichbar. Die einzelnen dargestellten Schritte müssen aber nicht zwingend sequentiell durchlaufen werden.

Zunächst geht es um das Verstehen seitens des PO und des Entwicklungsteams, was genau die Problemstellungen der Kunden sind und welche Ziele mit einer zu implementierenden Lösung erreicht werden soll. Dies kann mit einem initialen Erhebungsworkshop erfolgen sowie kontinuierlich im weiteren Verlauf der Produktentwicklung. Das Review-Event am Ende eines Sprints stellt neben der Abnahme von



Agile „Waschtrommel“

umgesetzten User-Stories im ablaufenden Sprint das Feedback in den Mittelpunkt. Es sind neue User Stories von anwesenden Kunden, Anwendern und weiteren Stakeholdern zu erheben und das Verständnis des Product Owners für bereits vorhandene User Stories bzw. Epics zu schärfen und zu konkretisieren. Die Konkretisierung bedeutet, dass neue User Stories oder Epics entstehen oder Akzeptanzkriterien neue oder bereits im Product Backlog vorhandene User Stories ergänzen. Das Ziel des Sprint Review ist im Scrum Guide 2013 nochmals deutlicher hervorgehoben worden – es kann ein potentiell neu ausgerichtetes Product Backlog zur Folge haben [SS13].

In einem Sprint Review mit mehreren anwesenden Stakeholdern könnte es durchaus passieren, dass sich die Anwesenden in ihren Wünschen widersprechen. Der Product Owner hat nun die Aufgabe, die zuwiderlaufenden Wünsche zu konsolidieren und abzustimmen. Dies passiert im Einklang mit der Produktvision, die einen übergeordneten Rahmen für die Entwicklung vorgibt, dem bereitgestellten Budget und dem nachhaltigen Mehrwert des Produkts.

Die Besonderheit eines Backlogs liegt in der eindeutigen Rangfolge der Backlog-Einträge. Die Aufgabe des Product Owners ist es, im Rahmen einer Priorisierung diese Rangfolge zu bilden und bei Bedarf kontinuierlich anzupassen. Es sind neue Einträge hinzuzufügen, überflüssig gewordene Einträge zu entfernen, zu große Epics zu splitten und die in User Stories aufgeteilten Elemente im Product Backlog ggf. völlig unterschiedlich einzugruppieren.

Eine weitere Aktivität stellt das Konkretisieren von User Stories dar, die zeitnah umgesetzt werden sollen. Im Sprint Planning werden die geplanten und im Product Backlog hoch priorisierten User Stories für den anstehenden Sprint durch den Product Owner vorgestellt und durch Akzeptanzkriterien detailliert und konkretisiert. Es wird dadurch das WAS geklärt, dieser Event dient der Anforderungsanalyse. Eine sukzessive Detailierung der User Stories durch Akzeptanzkriterien kann in Satzform, Beispieldaten, Entscheidungstabellen, Skizzen oder Szenarien erfolgen. Der häufig vernommene Vorwurf, dass der Informationsgehalt einer User Story nicht ausreichend ist, kann durch die Ausarbeitung und Besprechung von Akzeptanzkriterien deutlich widerlegt werden. In diesem Zusammenhang können User Stories auch mit nichtfunktionalen Anforderungen in Verbindung gesetzt oder für das System allgemein gültige nichtfunktionale Anforderungen in die „Definition of Done“ aufgenommen werden, die den Qualitätsanspruch eines Entwicklungsteam darstellen.

Ein zusätzlicher Event namens Story Time

Für zahlreiche Entwicklungsteams ist es nicht ausreichend, wenn der PO seine User Stories erstmalig im Sprint Planning Event vorstellt und diese im unmittelbar anstehenden Sprint umgesetzt werden müssen. Häufig treten noch Fragestellungen auf, die weder der PO noch Mitglieder des Entwicklungsteams beantworten können. Aus diesem Grunde haben viele Scrum-Teams ihr Framework um ein weiteres Event erweitert und nennen es Story Time, Backlog Refinement oder Grooming Meeting. In diesem Event werden User Stories durch den Product Owner vorgestellt und Akzeptanzkriterien besprochen, um ein gemeinsames Verständnis sicherzustellen und gleichzeitig die Schätzung der Stories durchzuführen. Dabei werden nur User Stories besprochen, die in den unmittelbar anstehenden 1-2 folgenden Sprints umgesetzt werden sollen oder für die der Product Owner eine Einschätzung des Aufwands von seinem Entwicklungsteam für eine User Story oder ein Epic möchte, um eine weitere Entscheidungsgrundlage für die Eingruppierung im Backlog zu erhalten. Der Product Owner nutzt diesen Event um das Entwicklungsteam frühzeitig einzubinden und die vorhandenen Team-Kompetenzen zu nutzen.

„Und welches Tool haben Sie eingesetzt?“

Ich bin immer wieder überrascht, wie schnell die Frage nach Tools gestellt wird, wenn über Requirements Engineering, Produktentwicklung oder über erfolgreiche Projekte diskutiert wird. Der spezifische Einsatz von Tools hat nach Auswertung der Standish Group in keinem ihrer erfolgreich ausgewerteten Projekte entscheidend für den Projekterfolg beigetragen. Der Einsatz von Tools zum Projekterfolg wird häufig überbewertet, denn Know-how, direkte Kommunikation und Managementunterstützung sind für den Projekterfolg wesentlich ausschlaggebender [StG13].

Trotzdem ist ein Tooleinsatz für das Backlog-Refinement in vielen Entwicklungsprojekten sinnvoll, wenn dadurch die direkte Kommunikation nicht beeinträchtigt wird. So ist es beispielsweise in regulierten Umfeldern zwingend erforderlich, die Anforderungen nachverfolgbar zu halten. Es muss z. B. für eine Zertifizierungsorganisation nachvollzogen werden können, wer die Anforderungen gestellt hat und wie deren Umsetzungsweg war, um eine Marktzulassung (z. B. für Medizinprodukte) zu erhalten. Darüber hinaus lassen sich Anforderungen versionieren, gruppieren oder suchen und mit weiteren Attributen (z. B. Quelle der Anforderung) ergänzen und wenn nötig eine Traceability von der Anforderungsquelle bis zur Codeumsetzung sicherstellen.

Diese Maßnahmen haben aber keinen Selbstzweck, sondern sie sind nur durchzuführen, wenn sie einen tatsächlichen Mehrwert für die Entwicklungsorganisation darstellen. Für kleine Unternehmen, die

für einen Kunden ein Produkt mit einem Entwicklungsteam umsetzen, kann es durchaus ausreichend sein, ausschließlich Karteikarten für die Anforderungen zu verwenden und damit eine Backlog Refinement durchzuführen.

Zusammenfassung

Zusammenfassend kann man die Aufgabe eines Requirements Engineers oder Product Owners auch folgendermaßen ausdrücken: Anforderungen sind in einer angemessenen Detailierung, zum benötigten Zeitpunkt und für die zu erwartenden Beteiligten verständlich darzustellen. Um dies sicherzustellen lassen sich RE-Aktivitäten im Scrum-Framework an zahlreichen Stellen identifizieren. Die Anwendung findet im Gegensatz zu einem phasen- oder dokumentengetriebenen Entwicklungsansatz aber kontinuierlich während der gesamten Entwicklungszeit statt. Die Kommunikation zwischen Fachseite und Entwicklern muss im Mittelpunkt stehen und reine Übergaben von Dokumenten sind zu vermeiden. Zudem werden die zu erstellenden Artefakte nur so umfassend wie nötig und parallel zur Entwicklung oder unmittelbar vor der Umsetzung erstellt. Die Aktivitäten des Requirements Engineerings sollten niemals als Selbstzweck durchgeführt werden, sondern einen deutlichen Mehrwert bieten, um die Bedürfnisse und Ziele der Kunden erfolgreich im Produkte oder Systems umzusetzen.

Quellen:

[AMa01]
„Prinzipien hinter dem Agilen Manifest“, 2001

[Coh10]
„User Stories: für die agile Software-Entwicklung mit Scrum, XP”, Mike Cohn, mitp, 2010

[SS13]
„The Definitive Guide to Scrum: The Rules of the Game”, Ken Schwaber, Jeff Sutherland, Juli 2013, scrum.org

[StG13]
„CHAOS MANIFESTO 2013 – Think Big, Act Small”, The Standish Group International, 2013, Boston

[JSB11]
„Use-Case 2.0”, Ivar Jacobson, Ian Spence, Kurt Bittner, 2011

Autor



Andreas Becker

Andreas Becker ist Agile Coach, Senior Consultant und Trainer. Er coacht Product Owner, agile Entwicklungsteams sowie Scrum-Master und unterstützt Unternehmen auf ihrem Weg zu agilen Organisationen. Einen Schwerpunkt seiner Tätigkeit bildet die Einführung der agilen Entwicklung auf Basis des „Scaled Agile Framework“ SAFe™. Seit 2010 ist Herr Becker Lehrbeauftragter an der Hochschule Rosenheim für die Vorlesung „Requirements Engineering im agilen Umfeld“.

www.Agile-by-HOOD.de
andreas.becker@HOOD-Group.com