



□ Joscha Jenni

(joscha.jenni@mimacom.com)

leitet den Bereich Projekt Management für die mimacom ag und arbeitet als Projektleiter und Berater im Bereich Softwareentwicklung. Seit mehreren Jahren beschäftigt er sich mit agilen Ansätzen und deren Implementierung.



□ Silvio Wandfluh

(silvio.wandfluh@mimacom.com)

ist als Scrum Master bei der mimacom ag tätig. Neben der Leitung von Projekten ist er in den Bereichen Continuous Process Improvement und Project Management Innovation tätig. Er hat den Requirements Engineering Prozess der mimacom aufgebaut.

## Ein Interview mit dem Product Backlog

Das Product Backlog ist eines der zentralen – wenn nicht das zentralste – Artefakt in Scrum. In diesem werden die funktionalen und nicht-funktionalen Requirements definiert und verwaltet. Viele Projekte scheitern aus unserer Sicht unter anderem daran, dass das Product Backlog nicht in ausreichender Güte erstellt und nicht professionell verwaltet, weiterentwickelt und adaptiert wird. Wir hatten die einzigartige Gelegenheit mit dem Product Backlog aus einem unserer Entwicklungsprojekte ein Interview durchzuführen. Dabei gehen wir von folgenden Rahmenbedingungen aus: Das neu zu entwickelnde Produkt wird in einem Auftraggeber/Auftragnehmer-Verhältnis entwickelt. Das Product Backlog wird auf der Grundlage eines Festpreises für die Entwicklung einer individuellen Web-Applikation im Enterprise-Umfeld erstellt. Die Grundlagen dazu sind in einem Lasten- und/oder Pflichtenheft festgehalten. Das Projektteam besteht aus einem Product Owner (Auftraggeberseite), einem Scrum Master sowie einem Development-Team. Letztere beiden Rollen werden von dem Auftragnehmer wahrgenommen.

*Das Interview führten Joscha Jenni und Silvio Wandfluh (nachfolgend kurz JW) mit dem Product Backlog (nachfolgend kurz PB) am Freitag, 23. August 2013*

**JW:** Vor etwas mehr als zehn Jahren wurde das agile Manifest [AgM] publiziert und seitdem sind Sie nun auf dem Markt tätig. Wie haben Sie diese Zeit erlebt und was hat sich Ihrer Ansicht nach geändert?

**PB:** Insbesondere hat sich die Flexibilität und Transparenz innerhalb von Projekten erhöht. Ob dies nur auf mich zurückzuführen ist, ist schwierig zu sagen. Vieles hat sich leider auch nicht maßgeblich verändert. Nach einer Studie von SwissQ und der Universität St. Gallen [SQC] sehen 45,1 % der Befragten den Reifegrad ihres Requirements Engineering als mittelmäßig bis schwach an. Missverständnisse in der Kommunikation sowie wachsende oder sich ändernde Anforderungen gelten als Hauptgründe für ungenügende Requirements.

**JW:** Vor nicht allzu langer Zeit wurden Softwareprojekte mittels abschließender Spezifikation gestartet. Wo sehen Sie die Vorteile eines Product Backlogs gegenüber einer klassischen Spezifikation?

**PB:** Die Flexibilität für alle Beteiligten ist mit Sicherheit der größte Vorteil eines Product Backlogs im agilen Sinne. Der Product Owner hat die Möglichkeit, neue Requirements ins Product Backlog aufzunehmen und solche mit einem geringeren Nutzen niedriger zu priorisieren oder ganz zu entfernen. Gerade bei volatilen Marktbedürfnissen kann die Verwendung eines flexiblen Ansatzes mittels Product Backlog ein entscheidender positiver Faktor sein.

Bei einem Festpreisprojekt gelten allerdings einige Regeln, die unbedingt zu beachten sind. Bereits bei Erstellung der Offerte sind Requirements grob granular bekannt, um überhaupt einen Festpreis anbieten zu können. In diesem Fall gilt es, möglichst schnell den Scope (Umfang) für

das Product Backlog zu definieren. Die Anforderungen werden soweit analysiert und verfeinert, bis es möglich ist, das Produkt in einzelne Teile zu zerlegen und diesen einen Wert zu geben.

Zusammengezählt ergeben die Werte der einzelnen Teile den finanziellen Rahmen für den Product Scope. Dieser wird vom Kunden abgenommen. Ist der Product Scope einmal abgenommen, ist eine der wichtigsten Grundlagen für den Umgang mit dem Product Backlog geschaffen, die Grundlage für das Change Management. Der Kunde erhält damit zwei Möglichkeiten, neue oder geänderte Requirements in das Product Backlog einzubringen.

- Ein neues Requirement wird mit einem bereits bestehenden Requirement gleichen Wertes ausgetauscht. Der Preis für das Produkt bleibt derselbe.
- Das Volumen des Scopes wird durch das neue Requirement erweitert, was dazu führt, dass auch neues Budget

für die Umsetzung des neuen Requirements zur Verfügung gestellt werden muss.

Grundsätzlich nichts Neues. Ich kann aus Erfahrung sagen, dass ein professionelles Change Management im Umgang mit dem Product Backlog mitunter eine der wichtigsten Disziplinen ist, die in einem Softwareprojekt beherrscht werden muss. Dies gilt, ob nun agil oder klassisch vorgegangen wird.

**JW:** Wie stehen Sie zur Disziplin „Requirements Engineering & Management“?

**PB:** Requirements Engineering & Management ist ein sehr dehnbarer Begriff, der in der Literatur unterschiedlich erläutert wird. Ich verstehe unter Requirements Engineering einen Prozess, der zum Ziel hat, die Requirements so zu beschreiben, dass deren Umsetzung dem Kunden den größtmöglichen Nutzen bringt. Das Ergebnis dieses Prozesses ist das Product Backlog.

Mit dem **Requirements Engineering Process** wird ebenfalls definiert, wie die Requirements während dem Prozess verwaltet werden. Somit ist im Requirements Engineering Process auch das Requirements Management enthalten. Der Requirements Engineering Process kann grob in drei Phasen unterteilt werden, wobei im Sinne der Agilität die Phasen in mehreren Iterationen durchlaufen werden können.

■ **Business Case erarbeiten – Das Business verstehen.** Damit die Requirements am Ende das beschreiben, was dem Kunden den größtmöglichen Nutzen bringt, muss das Business, in dem sich der Kunde bewegt, genau verstanden werden. Damit wird auch die Antwort auf die Frage gefunden, was das neue Produkt oder der neue Service verbessern soll und wo eine neue Lösung zum Einsatz kommen muss. Ein gemeinsames Verständnis und eine Vision, die von allen Stakeholder getragen wird, ist von zentraler Bedeutung.

■ **Lösung erarbeiten – Wie soll das Produkt oder der Service aussehen?** Um diese Frage mit ersten Modellen beantworten zu können, muss das Problem, welches der Grund dafür ist, ein neues oder angepasstes Produkt zu implementieren, erkannt werden. Dafür wird ein gemeinsames Verständnis

geschaffen, was essenziell verbessert werden muss, um das Problem zu lösen und damit das Business effizienter zu gestalten.

■ **Produkt oder Service definieren – Definition verstandener Requirements.** In der letzten Phase werden die Requirements definiert. In welcher Form dies geschieht, ist stark von den Erkenntnissen und dem geschaffenen gemeinsamen Verständnis der vorangegangenen Phasen abhängig.

Um diesen Prozess besser zu verstehen, möchte ich eine kurze Analogie zu einem Fußballtrainer geben, der den Auftrag hat, ein Team zum Erfolg zu führen.

Stellen Sie sich einen neuen Trainer eines Vereins vor. Er kommt in ein neues Umfeld, bestehend aus Geschäftsleitung, Sportdirektor, Spielern, Supportern, usw., die als seine Stakeholder bezeichnet werden können. Das Business ist schnell klar: Das Ziel ist der Gewinn der Meisterschaft am Ende der Saison und die Vision für die Teilnahme an der Champions League. Zusammen mit den Spielern schafft sich der Trainer das gemeinsame Verständnis, was diese Vorgaben bedeuten. Sein Produkt ist, das Team so zu formen, dass alle Stakeholder mit Erfolgen verwöhnt werden. Das Business hat er damit verstanden.

Nun gilt es, in der Vorbereitungsphase zu entdecken, wo in der vergangenen Spielzeit das Problem lag und was als oberste Priorität verbessert werden muss, um eine Lösung für ein effizienteres und damit erfolgreicherer Spielsystem zu finden. Zusammen mit den Spielern wird an Spielzügen, Taktiken und einzelnen Skills gearbeitet, bis alle ein gemeinsames Verständnis darüber haben, wie das Team zum Erfolg kommen soll.

Dies ist durchaus mit der Modellierung und somit der Produktdefinition vergleichbar. Damit ist definiert, wie das Produkt aussehen soll. Vor dem ersten Spiel treffen sich Trainer und Spieler in der Garderobe, um sich auf das bevorstehende Spiel einzustellen. Im Detail wird nun zusammen besprochen, wie die Aufgabe angegangen wird. Der Gegner wurde im Vorfeld analysiert. Zusammen wird eine detaillierte Taktik, die den Anforderungen an das Team und die Spieler entspricht, zurechtgelegt. Damit ist der Vorbereitungsprozess abgeschlossen und das Spiel – die Implementierung – kann angepfiffen werden.

**JW:** Welche Strukturierung würden Sie im Sinne eines Spielzuges definieren?

**PB:** Meine Erfahrung zeigt, dass sich im agilen Umfeld eine Strukturierung in zwei bis drei, maximal vier Hierarchiestufen bewährt. Zentrales Element der Spezifikation von Requirements bilden User Stories. Diese werden durch Akzeptanz-Kriterien verfeinert und in Epics zusammengefasst.

Die Granularität von User Stories kann je nach Produkt und Projekt variieren. Diesbezüglich sollte den Verantwortlichen für das Requirements Engineering auch die nötige Freiheit gewährt werden. Unnötige Einschränkungen wirken sich negativ auf die Qualität der Requirements aus.

Epics können beispielsweise Prozesse, Use Cases, Schnittstellen oder Funktionen sein. Ich schlage vor, dass zwischen Architektur- und Business-Epics unterschieden wird, wobei die Architektur-Epics durch die nicht-funktionalen Requirements definiert werden und die Business-Epics ihrerseits durch die funktionalen.

**JW:** Wie sieht das Spielfeld des Requirements Engineering Process aus? Wo spielen Sie am liebsten, beziehungsweise, wo werden Sie am liebsten verwaltet?

**PB:** Das kommt ganz auf die Anforderungen an, die eingehalten werden müssen. Hier spielen beispielsweise Fragen bezüglich der Transparenz, Nachvollziehbarkeit, Datensicherheit oder Flexibilität eine wichtige Rolle. Gemäss SwissQ [SQC] gilt Microsoft Office mit 67,6 % (Word und Excel) immer noch als das meist genutzte Werkzeug im Bereich Requirements Engineering, gefolgt von moderneren Tools wie Atlassian JIRA, die ihren Marktanteil von 1,8 % auf 27,2 % erhöhen konnten.

Diese Ergebnisse lassen sich natürlich nicht 1:1 auf die Verwaltung des Product Backlog zurückschliessen, aber sie decken sich ziemlich gut mit meinen Marktbeobachtungen. Aber um auf Ihre Frage zurückzukommen: Unter den eingangs erwähnten Rahmenbedingungen werde ich tatsächlich am liebsten in JIRA verwaltet.

**JW:** Bleiben wir beim Vergleich mit dem Fußballspiel. Hier werden von den Sportvereinen Saisonziele festgelegt. Gibt es etwas Vergleichbares in Softwareprojekten? Oder anders ausgedrückt, was steht am Anfang des Life Cycles eines Product Backlog?

**PB:** Diese Parallelen bestehen durchgängig. Es ist sehr wichtig, dass bereits zu Beginn der Erstellung eine klar definierte Vision vorhanden ist, die dem Projektteam bereits eine bestimmte Richtung vorgibt. Diese Vision sollte nicht nur festgehalten sein. Jeder involvierte Stakeholder sollte sich zur Vision bekennen und seine Handlungen daran ausrichten.

Genauso wichtig wie die Vision sind gemeinsame Ziele, die während des Vorhabens erreicht werden sollen. Die Vision beschreibt in der Softwareentwicklung einen Sollzustand, der längerfristig (je nach Branche und Software zwischen zwei und zehn Jahren) erreicht werden soll. Ziele sind kurzfristiger, da sie für jedes Release oder noch besser für jede Iteration neu definiert werden sollten.

Diesbezüglich habe ich die Erfahrung gemacht, dass Ziele nicht nur vorgegeben werden, sondern von allen beteiligten Personen mitgetragen werden sollten. Die Ziele sind ein wichtiger Bestandteil eines Projektteams. Ist die Vision und sind die Ziele einmal klar und verstanden, startet die Definition der Requirements an das neu zu entwickelnde Produkt.

**JW:** Wie stellen Sie sicher, dass die Vision vom Projektteam verstanden und getragen und im zu entwickelnden Produkt integriert wird?

**PB:** Dieses oft fehlende Verständnis ist meiner Erfahrung nach einer der häufigsten Gründe, warum Projekte scheitern und ein Hauptgrund dafür, dass das Endprodukt nicht den Bedürfnissen der Kunden entspricht. Egal wie genau das Product Backlog spezifiziert wird, wenn die beschriebenen Lösungen keine Probleme des Kunden lösen, wird dieser mit dem Produkt nicht zufrieden sein. Daher müssen die Verantwortlichen die Branche, in der sie sich bewegen, und die Bedürfnisse, welche vom Produkt erfüllt werden müssen, sehr genau kennen.

**JW:** Damit sind wir bei den Verantwortlichkeiten. Wer ist grundsätzlich verantwortlich für das Product Backlog?

**PB:** Eine Person muss abschließend verantwortlich sein. Im agilen Umfeld wird diese Rolle in den meisten Fällen als Product Owner bezeichnet. Die drei wichtigsten Aufgaben des Product Owner sind die Definition der Requirements (was soll umgesetzt werden), die Priorisierung der Re-

quirements sowie die Abnahme der umgesetzten Backlog Items.

Falls es in einem Projekt ein Auftraggeber/Auftragnehmer-Verhältnis gibt, ist die Rolle des Product Owner in der Regel auf der Seite des Auftraggebers zu besetzen. Der Auftragnehmer, welcher häufig den Product Owner bei der Spezifikation der Anforderungen unterstützt, sollte in solchen Fällen einen Verantwortlichen für das Requirements Management ernennen.

Diese Rolle wird häufig auch als Proxy-PO bezeichnet. Der Proxy-PO nimmt die Verantwortung des Single Point of Contact zwischen Auftragnehmer und Auftraggeber wahr. Bei Projekten, bei denen die Softwareentwicklung schneller ist als die Definition der Requirements, werden zusätzliche Requirements Engineers eingesetzt, die durch den Product Owner (PO) oder Proxy-PO geführt werden. Es besteht durchaus die Möglichkeit, dass Entwickler zusätzlich die Rolle eines Requirements Engineers übernehmen.

**JW:** In der Praxis stellt sich oft die Frage, ab wann mit der Implementierung begonnen werden kann. Wie gehen Sie diese Herausforderung an?

**PB:** Versetzen wir uns wieder in die Situation eines Fußballteams. Es muss definiert werden, was in der Vorbereitung bis zum ersten Meisterschaftsspiel getan werden muss. Zusammen werden beispielsweise Spielzüge definiert, die im Wettkampf umgesetzt werden. Was heißt dies nun für das Softwareprojekt?

Grundsätzlich verfolge ich die Daumenregel, dass für eine Produktentwicklung, je nach Projektlänge, 2-3 Sprints im Detail spezifiziert sein sollten, damit ein gewisser Arbeitsvorrat vorhanden ist. Damit wird sichergestellt, dass sich das Business und das Entwicklungsteam mit dem zu entwickelnden Produkt auseinandergesetzt haben. Dazu werden die Epics benötigt, die im geplanten Release umgesetzt werden sollen.

Als Mindestanforderung sehe ich ebenfalls einen ersten Entwurf der Softwarearchitektur sowie des Softwaredesigns. Um eine erste Iteration zu starten, müssen vom Business und dessen Vertretern (Product Owner) die entsprechenden Requirements abgenommen sein, respektive der Definition of Ready (DoR) entsprechen.

In einer sogenannten DoR wird spezifiziert, welche Punkte erfüllt werden müs-

sen, damit ein Requirement für die nächste Iteration eingeplant und umgesetzt werden kann. Andererseits hat auch das Entwicklungsteam ein Mitspracherecht. Es bestimmt mit, wann ein Requirement bereit ist für die Umsetzung.

**JW:** Sie bringen mich direkt zur nächsten Frage. Was halten Sie von „Definition of Done“ und „Definition of Ready“, die ja im neuen Scrum Guide [ScG13] eine Aufwertung erhalten haben? Begegnen Sie den beiden Hilfsmitteln oft?

**PB:** Als Product Backlog bin ich natürlich viel enger mit der Definition of Ready (DoR) als mit der Definition of Done (DoD) verbunden. Die DoR erfüllt für das ganze Projektteam eine zentrale Aufgabe, da sie als Bindeglied, man kann sogar sagen als eine Art Vertrag zwischen Product Owner und Entwicklungsteam fungiert. Wie ich schon erwähnt habe, hält das Entwicklungsteam in der DoR die Anforderungen an ein Requirement fest, die erfüllt werden müssen, damit ein Requirement überhaupt für eine Iteration eingeplant wird. Dass sich das Entwicklungsteam zusammen mit dem Business während des Projekt-Setups zur DoR bekennt, ist Grundlage für deren erfolgreichen Einsatz.

Das gleiche gilt auch für die DoD. Sie beschreibt in Form einer Checkliste die zu erfüllenden Punkte bei der Implementierung, damit ein umgesetztes Requirement dem Kunden ausgeliefert wird. Die DoD wird vom Product Owner zusammen mit dem Entwicklungsteam definiert und hat zum Ziel, dass ein umgesetztes Requirement bei der Auslieferung einer gewissen Qualitätsgüte entspricht.

Somit ist die DoD ein wichtiges Instrument für die Sicherstellung der geforderten Qualität. Leider begegne ich den beiden Hilfsmitteln im Markt viel zu selten. Und wenn sie einmal beschrieben und somit transparent vorliegen, werden sie selten konsequent angewendet oder angepasst. Hier sehe ich ein großes Potenzial.

**JW:** Apropos großes Potenzial. Woran liegt es, dass viele Projekte nicht nach dem Business Value priorisiert werden?

**PB:** An fehlendem Know-how sowie Mut und der fehlenden Erkenntnis, dass diese Herangehensweise dazu führt, dass die Kundenanforderungen mit einem Produkt befriedigt werden, das ihren Ansprüchen

entspricht. Nach Business Value zu priorisieren bedeutet, dass entsprechende Werthaltungen in einer Unternehmung oder einem Projekt aufzubauen sind und das Produkt gleichzeitig aus vielen verschiedenen Gesichtspunkten und Interessen zu beleuchten.

Eine Anforderung hat aus der Sicht des Anwenders und des Auftraggebers selbstverständlich einen unterschiedlichen Business Value. Der Product Owner ist hier die zentrale Figur. Am Ende des Tages sagt er/sie, welche Faktoren in welcher Gewichtung den Business Value beeinflussen. In diesem Zusammenhang bilden die Priority Guidelines eine wichtige Rolle.

Für die Priorisierung von Requirements gibt es eine Vielzahl von Methoden. Eine davon sind sogenannte Schiebenleger, durch die Requirements auf der Basis der nicht-funktionalen Requirements bewertet werden. Eine weitere, etwas aufwendigere Methode zur ökonomischen Priorisierung stellt der „Weighted Shortest Job First“ (WSJF) [SaF] dar.

Hierbei wird der Ratio aus der Summe von Business Value, Time Value sowie Risikoreduktion und „Jobgröße“ ermittelt (User Value+Time Value+Risk Reduction / Job Size). Eine eher einfachere und pragmatischere Methode stellt die MoSCoW-Priorisierung dar.

**JW:** Welche Trends sehen Sie für die kommenden Jahre im Bereich Requirements Engineering und Product Backlog?

**PB:** Ich gehe davon aus, dass sich Methoden zur Qualifizierung von Requirements

und zur Quantifizierung von Qualität im Bereich Requirements Engineering vermehrt durchsetzen werden und somit zu einer höheren Qualität des Product Backlog beitragen.

Viel Potenzial sehe ich beim agilen Requirements Engineering und insbesondere bei der Priorisierung der Requirements durch den Business Value. Einen weiteren Trend beobachte ich im Acceptance Test Driven Development (ATDD). Ob sich diese Trends durchsetzen werden, wird uns die Zukunft zeigen. ■

## Referenzen:

**[AgM]** Agile Manifesto, <http://agilemanifesto.org/>

**[SQC]** SwissQ Consulting AG, Trends & Benchmarks Report Schweiz, Requirements 2013, [http://www.swissq.it/wp-content/uploads/2012/11/Requirements-Trends-und-Benchmarks-2013\\_Web.pdf](http://www.swissq.it/wp-content/uploads/2012/11/Requirements-Trends-und-Benchmarks-2013_Web.pdf)

**[ScG13]** Scrum Guide,

<https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf#zoom=100>

**[SAF]** Scaled Agile Framework, <http://www.scaledagileframework.com>