

Project Management

Alle Anforderungen im Griff?

Mit agilem Requirements Engineering Projekte erfolgreich machen

Ulrike Umkehrer-Neudeck, Senior Manager
Braincourt GmbH

Inhaltsverzeichnis

1	Einleitung.....	3
2	Klassisches und agiles Anforderungsmanagement im Vergleich	3
2.1	Klassisches Anforderungsmanagement – Hellsehen für Fortgeschrittene?.....	3
2.2	Agiles Anforderungsmanagement – ergibt die Optimierung aller Teilauslieferungen auch die Optimierung des Gesamtproduktes?	4
3	Die Notwendigkeit zur Nachvollziehbarkeit verbindet Klassik und Agilität.....	5
3.1	Nachvollziehbarkeit von Anforderungen durch die Verbindung von Use Cases mit User Stories.....	5
3.2	Nachvollziehbarkeit von Anforderungen durch Story Mapping	6
4	Klassisch oder agil, erfolgreiches Requirements Engineering benötigt Dialog	7
5	Ansprechpartner.....	8

1 Einleitung

Agile Projektmanagementmethoden boomen schon seit einigen Jahren. Gemäß der aktuellen Studie „State of Agile™ Survey“¹ sind mittlerweile bereits 94 Prozent der weltweiten Projekte agil. Folgt man dem „The State of Scrum Report“² haben 81% aller IT Projektteilnehmer schon mindestens einmal an einem agilen Projekt gearbeitet. Agilität ist inzwischen Mainstream geworden. Von allen agilen Frameworks ist Scrum das am meisten verbreitete. Viele DAX 30 Firmen haben in den letzten Jahren eigene, agile Frameworks konzipiert und implementiert. Dabei bieten die wenigsten dieser agilen Frameworks jedoch konkrete Hilfestellung zur systematischen, nachhaltigen und effizienten Ermittlung, Analyse, Beschreibung, Konsensbildung und Verabschiedung von Anforderungen, die in agilen Projekten umgesetzt werden sollen. Dies führt erfahrungsgemäß zu großen Problemen.

Agilität ist Mainstream geworden.

Früher wurden alle Anforderungen an ein Projekt in dem Fachkonzept festgehalten, das über einen längeren Zeitraum erstellt, geprüft, revidiert und verabschiedet wurde. Dieses setzte man danach über einen längeren Zeitraum um. Sich ändernde Anforderungen (z.B. veränderte Geschäftsprozesse, Mergers & Acquisitions) waren hier fehl am Platze. Auch die fehlende Anwenderintegration konnte dazu führen, dass eine Anwendung nach der vollständigen Umsetzung und Implementierung nicht akzeptiert wurde, weil entscheidende Features (z.B. Eingabemasken, fachbereichsspezifische Modelle, individuelle Auswertungsmöglichkeiten) nicht realisiert waren.

Die erhöhte Flexibilität bei agilen Projekten kann zu Lasten der Nachvollziehbarkeit, vor allem beim Anforderungsmanagement gehen.

Beim agilen Vorgehen werden Anforderungen kurz und prägnant in *UserStories* ‚Anwendergeschichten‘ festgehalten und in sogenannten *Sprints* ‚kurzen Zeitfenstern‘ nacheinander abgearbeitet. Nach jedem Sprint werden die neuen Funktionalitäten durch Anwender getestet und nutzbare Funktionalitäten bereits an den Betrieb übergeben. Sinnvolle Änderungen können so schneller umgesetzt werden. Doch wie verliert man dabei das *Big Picture*, das ‚Gesamtbild‘, die Nachhaltigkeit und Rückverfolgbarkeit der Anforderungen nicht aus den Augen? Gibt es Möglichkeiten das Beste aus beiden Welten miteinander zu verbinden?

2 Klassisches und agiles Anforderungsmanagement im Vergleich

2.1 Klassisches Anforderungsmanagement – Hellsehen für Fortgeschrittene?

Klassische Projektvorgehensmodelle angelehnt an PMBOK®, PRINCE2® oder das V-Modell werden in den IT Organisationen deutscher Unternehmen nun schon seit sehr langer Zeit geschult und angewendet. Das Zusammenspiel mit Fachabteilungen, Kunden und externen Providern ist gut erprobt und mittlerweile ausreichend akzeptiert.

¹ State of Agile™ Survey VERSIONONE.COM ©2015 VersionOne, Inc. All rights reserved (4.6.2015)

² THE STATE OF SCRUM: BENCHMARKS AND GUIDELINES, Scrum Alliance.org (4.6.2015)

Das *Requirements Engineering* oder auch ‚Anforderungsmanagement‘ steht dabei immer am Beginn des Entwicklungsprozesses. Den Abschluss der Anforderungsaufnahme bildet im Regelfall ein detailliertes Fachkonzept oder ein ausführliches Lasten- bzw. Pflichtenheft. Im Idealfall sind anspruchsvolle Anzeige Masken (UI), komplexe Rechenoperationen oder Workflows ausführlich und anschaulich beschrieben. Ist das Dokument formell abgenommen, bildet es die inhaltliche und formelle Grundlage für den Entwicklungsauftrag an die IT und vertragliche Vereinbarungen mit externen Providern.

Die Abnahme der erstellten Lieferobjekte kann in diesem Modell relativ einfach über einen Abnahme- oder Akzeptanztest durchgeführt werden, der feststellt, ob die Liste der beschriebenen Anforderungen erfüllt wurde. Sowohl das IT Konzept als auch das spätere Betriebskonzept können auf dem Detaillierungsgrad des Pflichtenhefts aufsetzen und ausgearbeitet werden.

Dieses Vorgehen ist sinnvoll und effizient und die Anforderungen sind nach allen Seiten zu jeder Zeit nachvollziehbar, es gibt allerdings einen entscheidenden Nachteil: In der Zeit ab Abnahme des Pflichtenheftes bis zur Abnahme des fertig entwickelten Systems liegt in den meisten Fällen ein Zeitraum von mindestens 6-12 Monaten. In dieser Zeit können sich heute sowohl grundlegende Anforderungen an die Funktionalität als auch entscheidende Prioritäten der Stakeholder ändern, so dass das fertige Produkt nicht mehr den aktuellen Anforderungen entspricht oder bereits wieder geändert werden muss, bevor es überhaupt eingeführt wurde.

Um Änderungen in der Zukunft bereits in der Projektanfangsphase vorauszu- sehen, wären hellseherische Fähigkeiten notwendig. Diese sind in Unternehmen jedoch nur in den seltensten Fällen anzutreffen. Gerade weil sich die Änderungs- frequenz in der heutigen Geschäftswelt immer noch weiter erhöht, entscheiden sich mehr und mehr Unternehmen für ein agiles Projektvorgehen.

2.2 Agiles Anforderungsmanagement – ergibt die Optimierung aller Teilauslieferungen auch die Optimierung des Gesamtproduk- tes?

In agilen Projekten sehen sich die anfordernden Fachbereiche in regelmäßigen Abständen an einem Tisch mit der IT sitzend, bunte Moderationskarten vor sich liegend, und mit der Aufgabe konfrontiert, ihre Anforderungen als User Stories möglichst einfach zu formulieren, zu diskutieren³ und anschließend zu priori- sieren. Vereinfacht gesagt, beschreiben User Stories in kurzen prägnanten Aus- sagen einzelne funktionelle oder nicht-funktionelle Verhaltensweisen des zu ent- wickeln Systems. Dabei wird vor allem das „Wer?“ (Rolle) und das „Was?“ (Ziel/Wunsch) in einfacher Sprache festgehalten, bspw. „Als Controller benötige ich ein Kommentarfeld beim Umsatzbericht X“.

Klassisches Anforderungs- management ist ausführlich, detailliert und nachvollzieh- bar.

Dabei lassen Ergebnisse länger auf sich warten und für Flexi- bilität bleibt wenig Spielraum.

³ Ron Jeffries; Essential XP: Card, Conversation, Confirmation (30.08.2001)

Die User Stories werden dann im sogenannten Backlog in Listenform dokumentiert, in Reihe gebracht und je nach Priorität in kurzen Iterationszyklen zur Umsetzung durch die Entwicklungsteams gegeben. Am Ende jedes Zyklus werden sowohl das entstandene bzw. weiterentwickelte Produkt als auch der Herstellungsprozess überprüft und verbessert, neue Informationen bei der Priorisierung der User Stories im Backlog berücksichtigt und ein neuer Entwicklungszyklus beginnt mit einem ausgewählten Set an höchstpriorisierten Anforderungen.

In agilen Projekten werden Anforderungen in kleine Häppchen verpackt und sinnvoll portioniert.

Dieses Vorgehen ist äußerst wirksam, wenn es darum geht unnötige Komplexität aus Funktionsbeschreibungen, Architektur und Programmierung herauszunehmen und die Entwicklung an den aktuellen Informationsstand auszurichten.

Ergebnisse werden schnell sichtbar, Komplexität wird reduziert und Änderungen sind jederzeit möglich.

Können damit aber auch sehr umfangreiche Projektvorhaben mit komplexen Anforderungen erfolgreich, gesamthaft und nachhaltig umgesetzt werden? Ist es möglich mit User Stories Event gesteuerte Workflows zu beschreiben oder komplexe Zusammenhänge oder Rechenoperationen darzustellen, die sich über verschiedenen Systemebenen oder Komponenten erstrecken ohne dass der übergeordnete Zusammenhang verloren geht? Wird das Produkt am Ende die übergeordnete Geschäftsstrategie und die Produktvision erfüllen?

3 Die Notwendigkeit zur Nachvollziehbarkeit verbindet Klassik und Agilität

3.1 Nachvollziehbarkeit von Anforderungen durch die Verbindung von Use Cases mit User Stories

Ivar Jacobson, Ian Spence und Kurt Bittner veröffentlichten im Dezember 2011 ein Konzept für den Einsatz von Use Cases⁴ im agilen Projektvorgehen, genannt Use Case 2.0⁵.

Use Cases ‚Anwendungsfälle‘ basieren kurz gesagt, auf einer Vorgehensweise zur Entwicklung von Anforderungen, die das Verhalten von Hauptakteuren und unterschiedlich möglichen Handlungsabläufen auf verschiedenen Aggregations-ebenen beschreiben. Diese lassen sich in agilen Projekten anwenden, indem User Stories aus den Handlungsabläufen als sogenannte „Slices“ herausgeschnitten und je nach Priorität umgesetzt werden.

Anstatt User Stories relativ zusammenhanglos im *Backlog* ‚Aufgabenliste‘ zu erfassen, wird durch das Einziehen einer übergeordneten Use-Case-Ebene, ggf. auch mit Visualisierung durch Use-Case-Diagrammen, ein besserer Überblick über das System und mögliche Abhängigkeiten generiert. Durch die Ableitung von User Stories aus den Handlungsabläufen der Use Cases hat das Entwicklerteam ein besseres Bild von den übergeordneten Zusammenhängen zur Verwendung der Software und den Hauptakteuren, die diese Software bedienen sollen.

Ein Use Case stellt den gesamten Fahrplan dar, während User Stories die Haltestellen auf dem Weg zum Ziel beschreiben.

⁴ Ivar Jacobson: Object Oriented Development in an Industrial Environment, OOPSLA – Proceedings, October 4-8, 1987, ACM 0-89791-247-0/87/0010-0183

⁵ Ivar Jacobson, Ian Spence, Kurt Bittner; Use Case 2.0 (E-Book), Ivar Jacobsen International 2011

Der verhaltensbezogene Ansatz dieser agilen Requirements Engineering Technik stellt die notwendige Nachvollziehbarkeit und Integration agiler User Stories mit komplexen Anforderungen aus dem Business in sehr praxisnaher Weise her.

3.2 Nachvollziehbarkeit von Anforderungen durch Story Mapping

Story Mapping wurde erstmals von Jeff Patton im Artikel „It’s all in how you slice it“⁶ beschrieben.

Dieses Vorgehen liefert größtmögliche Transparenz und Einbindung für alle Stakeholder und ermöglicht, die Komplexität eines Projektes visuell abzubilden.

Zuerst werden die User Stories gesammelt, die aus der Perspektive eines Anwenders geschrieben sind. Es werden Attribute hinzugefügt wie „Wer nutzt das Feature?“, „Wie oft wird das Feature genutzt?“ und „Welchen Geschäftsnutzen besitzt dieses Feature?“.

Die User Stories werden dann in einer Gitterform (*Grid*) geordnet. Das Ordnungskriterium auf der x-Achse ist die Reihenfolge der Verwendung (z.B. auch die Schritte im Ablauf eines Use Cases) und auf der y-Achse die Häufigkeit der Verwendung durch den User. Es entstehen dabei keine Arbeitslisten, wie man sie vom Backlog her kennt, sondern eher Arbeitsschichten, sogenannte *System Spans*.

Durch die grafische Anordnung der User Stories in einer Story Map werden Zusammenhänge, aber auch Ungereimtheiten schneller sichtbar.

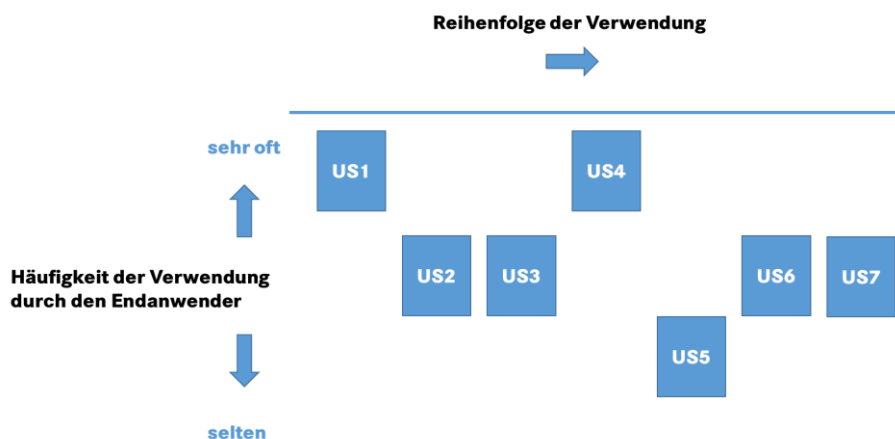


Abbildung 1: Erste Anordnung der User Stories - exemplarisch -

Durch die Anordnung der User Stories im Grid werden sowohl komplexe Arbeitsprozesse nachvollziehbar dargestellt als auch Logikbrüche oder Ungereimtheiten recht schnell sichtbar.

Durch das Einziehen von vertikalen Linien vor und hinter zusammengehörigen Geschäftsprozessen und horizontalen Linien nach dem geschätzten Funktionsumfang für einen Sprint hat man die Verbindung und Nachvollziehbarkeit der agilen User Stories zum übergeordneten Geschäftszusammenhang hergestellt.

⁶ Jeff Patton in BETTER SOFTWARE Magazine JANUARY 2005 http://agileproductdesign.com/writing/how_you_slice_it.pdf (4.6.2015)

Gleichzeitig werden so System Spans ermittelt, mit Hilfe derer man die Anzahl der notwendigen Releases zur Erfüllung der Produktvision abschätzen kann.

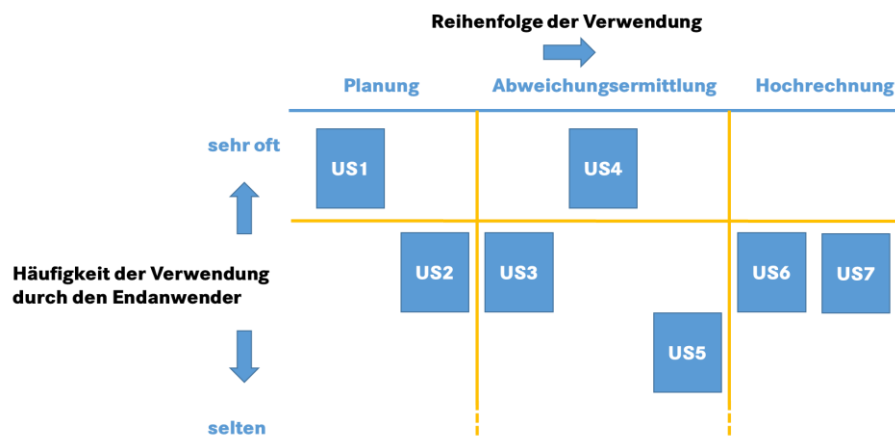


Abbildung 2: Bildung von System Spans - exemplarisch

Auch diese agile Requirements Engineering Technik ermöglicht die Nachvollziehbarkeit der Business Anforderungen in Kombination mit der agilen Entwicklung und Auslieferung kleinteiliger User Stories.

4 Klassisch oder agil, erfolgreiches Requirements Engineering benötigt Dialog

Es gibt also Möglichkeiten den Umstieg vom gewohnten Fachkonzept zur agilen User Story zu schaffen und dabei das „Big Picture“ im Auge zu behalten. Die Verbindung von Use Cases mit User Stories oder das Story Mapping bieten hierfür gute Ansätze.

Dennoch, es gibt keine einfache Lösung für gutes Requirements Engineering. Denn nichts ist missverständlicher als geschriebene Sprache. „Kommunikation ist zugleich eine der komplexesten und wichtigsten Fähigkeiten des Menschen und besteht eben nicht allein in der Weitergabe von sachbezogenen Informationen. Vielmehr laufen etwa zwei Drittel des Austausches in einem Gespräch über den visuellen oder akustischen Kanal in Form von Gesten, Körperhaltung, Mimik, Betonung oder Sprachmelodie.“⁷

Eine gute Kommunikation zwischen den Projektbeteiligten ist bei allen Methoden eine wesentliche Voraussetzung für den Erfolg.

Es ist in beiden Welten, klassisch oder agil notwendig, dass sowohl einfache als auch insbesondere komplexe Anforderungen mit den IT-Entwicklern persönlich durchgesprochen werden und ein Dialog stattfindet. Nur eine direkte Interaktion aller Beteiligten und wiederholte Revision der Anforderungen können Schwachstellen in Logik und Beschreibung aufdecken, doppeldeutige Inhalte eindeutig machen und das notwendige Verständnis sowohl für die Einzelanforderung als auch deren Einordnung in die übergeordnete Geschäftsstrategie und die Produktvision schaffen.

⁷ Stangl, W. (2015). Kommunikation. <http://arbeitsblaetter.stangl-taller.at/KOMMUNIKATION/> (4.6.2015)

5 Ansprechpartner

Ulrike Umkehrer
Senior Manager
Braincourt GmbH
Fasanenweg 11
70771 Leinfelden-Echterdingen
ulrike.umkehrer@braincourt.com
Telefon: +49 711 75 85 80 -47