

Miteinander reden statt gegeneinander schreiben

Susanne Mühlbauer
Agile • Coaching • Beratung • Training
Muehlbauer@Susanne-Muehlbauer.de

Ich bin Requirements Engineer und leidenschaftliche Agilistin. Während ich in meiner Arbeit als agile Coach Requirements Engineering (RE) als hilfreiche und notwendige Disziplin im agilen Arbeiten fördere und vermittele, möchte ich heute ein Plädoyer für mehr Agilität im RE halten.

Requirements Engineering und Agilität

RE bedeutet für mich, zu *verstehen*, was der Kunde erreichen möchte, zu *vereinbaren*, wie eine Lösung für sein Ziel aussehen soll und *sicherzustellen*, dass er diese auch bekommt [1].

Diese Definition ist bewusst lösungsfrei formuliert, sie gibt weder einen konkreten Entwicklungsprozess vor, noch fordert sie den Einsatz bestimmter Methoden, wie das Erstellen einer schriftlichen Spezifikation.

Wozu eigentlich agil arbeiten?

Agilität bedeutet für mich, werte- und prinzipienbasiertes Arbeiten [2], mit dem Ziel in einem sich ständig ändernden Umfeld erfolgreich zu sein [3]. Im folgenden beziehe ich mich auf Scrum [4] als ein Framework, das dabei hilft agile Werte und Prinzipien in der Praxis umzusetzen.

Für Unternehmen gibt es 3 wesentliche Gründe für agiles Arbeiten, wie einer Umfrage von Version One zu entnehmen ist [5]:

1. Wir wollen schneller auf den Markt
2. IT und Fachseite sollen besser aufeinander abgestimmt sein
3. Wir wollen auf Änderungen besser reagieren können

Schneller auf den Markt

Agile bzw. Scrum Initiativen starten häufig in so genannten „Water-Scrum-Fall“-Implementierungen. Hierbei wird eine (vollständige) Spezifikation erstellt, die in der Entwicklungsphase in Sprints abgearbeitet wird.



Abb. 1: Water-Scrum-Fall

Die Erwartung der Unternehmen, durch dieses Vorgehen schneller marktreife Produkte zu erstellen wird dabei häufig enttäuscht.

Die Entwicklungszeit per se verkürzt sich durch das Entwickeln in Sprints nicht zwangsläufig. Vielmehr sind technische Investitionen notwendig (Stichwort Continuous Integration/ Deployment, Testautomatisierung, ...), um die Geschwindigkeiten in der Entwicklung zu erhöhen. Und auch das ist nur in bestimmten Grenzen möglich.

Wo kann also im Entwicklungsprozess Zeit eingespart werden? Aus meiner Sicht liegt ein Schlüssel darin, die Spezifikationsphase aufzulösen und RE-Aktivitäten konsequent und kontinuierlich in die Sprints zu integrieren. Nur wie kann ein Entwickeln „ohne Spezifikation“ gelingen?

IT und Fachseite sollen besser aufeinander abgestimmt sein

Ein Ziel von Spezifikationen (und von RE) ist, ein gemeinsames Verständnis zwischen IT und Fachseite über die zu entwickelnde Funktionalität herzustellen. In einem Scrum Team passiert das, indem die Fachseite vertreten durch den Product Owner und das Entwicklungsteam gemeinsam kontinuierlich an den Anforderungen im Product Backlog arbeitet (Backlog Refinement). Umso besser funktioniert das, wenn im Scrum Team RE-Kompetenzen vorhanden sind.

Ziel des Backlog Refinements (und von RE) ist, ein gemeinsames Verständnis über die zu entwickelnde Funktionalität und die dahinterliegenden Ziele des Kunden/ Anwenders herzustellen. Hierbei wird die Spezifikation (gegeneinander schreiben) durch ein Gespräch (miteinander reden) ersetzt

Das Gespräch ist eine zentrale Forderung des agilen Manifests [2]: „Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.“

Bewährt hat sich in diesem Kontext das Format der User Story und das Konzept der 3C [6]:

- C für Card - Die Karte enthält nur soviel Text, dass die Anforderung identifizierbar ist.
- C für Conversation - Die User Story wird zu einem „Versprechen für ein Gespräch“.
- C für Confirmation - Eine User Story benötigt Akzeptanzkriterien (am ehesten vergleichbar mit konventionellen textuellen Anforderungen)

Kommen nun RE-Methoden wie Erhebungstechniken, Visualisierungstechniken (z.B. Modellierung) oder Fragetechniken zum Einsatz, geben diese dem Gespräch eine Struktur und unterstützen das Scrum Team beim Ausarbeiten der User Stories. Im Gespräch treffen die Fachexperten und die Experten für die

Lösung aufeinander. Gemeinsam wird erarbeitet, welche weiteren Informationen das Entwicklungsteam benötigt, um die gewünschte Funktionalität zu entwickeln bzw. welche Auswirkungen eine Änderungsanforderung nach sich zieht.

Die Lösungsexperten erlangen dadurch Domänen- und Business Know How und verstehen, welche Ziele der Kunde/ Anwender durch das System erreichen will. Dadurch können sie verschiedenen Lösungsalternativen betrachten und eine Empfehlung für die Fachseite aussprechen. Die Fachseite erhält einen Einblick in die Konsequenzen der verschiedenen Lösungsmöglichkeiten und eine technische Beratung.

Auf Änderungen besser reagieren

Das Product Backlog und sein Refinement richtet sich konsequent am Thema „Wert“ (für den Kunden und das Unternehmen selbst) aus und folgt einer übergeordneten Produktvision.

Die dahinterliegenden Business Ziele und technischen Ziele sind dem Entwicklungsteam bekannt. Die Produkt-(weiter-)entwicklung soll beispielsweise zur Qualitätsverbesserung beitragen, Kosten senken oder möglichst viele innovative Funktionalität liefern. Diese Ziele orientieren sich u.a. an den Marktanforderungen und können sich im Laufe des Produktlebenszyklus ändern. Statt diese Entwicklung nun beim Schreiben einer Spezifikation voraussehen zu wollen, fließen beim agilen Arbeiten Änderungen laufend in das Product Backlog ein.

Die Ziele werden während des Backlog Refinements regelmäßig überprüft und die Entwicklungsrichtung daran angepasst. Dadurch lassen sich Zielkonflikte zwischen Fachseite und IT identifizieren (z.B. Qualitätsverbesserung versus neue Funktionalität).

Sind die Ziele bzw. Zielkonflikte erst einmal bekannt kann ein Gespräch darüber begonnen werden und über die künftigen Entwicklungsumfänge verhandelt werden.

Neben der Neuverhandlung von Zielen, werden auch neue Anforderungen/ Änderungen gemeinsam im Scrum Team auf ihre Konsequenzen untersucht.

Nur noch miteinander reden?

Das agile Manifest sagt, wir schätzen „funktionierende Software mehr als umfassende Dokumentation“. Diese Aussage führt immer wieder zu Missverständnissen.

Aufschreiben ist erlaubt! Die Konversation zwischen allen Beteiligten ermöglicht es, Missverständnisse sofort auszuräumen und genau soviel (schriftliche) Spezifikation zu erstellen, wie für die Entwicklung notwendig. Im Gegensatz zu umfangreichen und vermeintlich vollständigen Spezifikationen, die (mehr oder weniger kommentarlos) übergeben werden, steht hier das Erzielen des gemeinsamen Verständnisses im Vordergrund. Die leichtgewichtige schriftliche Spezifikation hält den gemeinsamen Stand fest und stellt die Vereinbarung über den zu entwickelnden

Umfang dar. Diese kann mehr oder weniger umfangreich sein, das ist beispielsweise davon abhängig, wie gut das Team sich untereinander, die Domäne, die Technologie etc. kennt. Fehlende Informationen werden gemeinsam identifiziert und entsprechend gemeinsam erarbeitet und ergänzt.

Und was ist mit Dokumentation? Angenommen, eine Spezifikation wird bedarfsabhängig erstellt und zugunsten von Konversation reduziert, gilt das nicht für notwendige Dokumentation des entwickelten Systems. Dokumentiert werden sollte sofort nutzbares Wissen (z.B. Coding Guidelines) und langfristig notwendiges Wissen (z.B. Architekturentscheidungen, fachliche Anforderungen).

Dokumentation in Form von automatisierten Regressionstests sind sowohl sofort als auch langfristig nutzbar und sollten viel stärker ins Gewicht fallen. Diese Dokumentation erfolgt im Rahmen einer Definition of „Done“ laufend und kontinuierlich während der Entwicklung als Teil des zu liefernden Produktes.

Miteinander reden statt gegeneinander schreiben

Was hat der Artikel nun mit der Überschrift zu tun und mit meinem Plädoyer für mehr Agilität im RE?

Nun, RE ist keine für sich allein stehende Disziplin und kein Selbstzweck. Unternehmen setzen auf Agilität, um Ihre Ziele zu erreichen. RE kann das unterstützen oder behindern. Wir, als Requirements Engineers oder Business Analysten sollten uns trauen, neue Wege einzuschlagen. Unsere Rolle als Requirements Engineer wandelt sich.

Wir helfen Unternehmen, ihre Entwicklung auf den Kunden und Unternehmensziele abzustimmen.

Wir unterstützen Product Owner und Entwicklungsteams in ihrer Arbeit, wir werden Teil des Scrum Teams. Wir nutzen unsere RE-Kompetenz, um fokussiert und zielgerichtet miteinander zu reden. Wir helfen IT und Fachbereichen beim Erlangen eines gemeinsamen Verständnisses, wir spezifizieren soviel, wie notwendig ist, wir nutzen unser Know How für das Erstellen langfristiger Artefakte. Aus meiner Sicht ist diese Interpretation der Rolle Requirements Engineer auch nicht neu – wir räumen nur ein Hindernis aus dem Weg: Statt über eine Spezifikation „gegeneinander zu schreiben“, erstellen wir über das „miteinander reden“ gemeinsam eine andere Art von Spezifikation.

Quellen:

- [1]: Arbeiten von Colin Hood, Robert Stevenson, Jörg Jungermann (HOOD bzw. Agile-by-HOOD)
- [2]: agilemanifesto.org
- [3]: J. Appelo, Management 3.0, Addison Wesley, 2012
- [4]: scrumguides.org
- [5]: <http://stateofagile.versionone.com/>
- [6]: ronjeffries.com