

Benutzeranforderung in Agilen mittels User Stories

Mohamed Sohil
Sajed Guedich

21. Januar 2019

Zusammenfassung

Die Anforderungen in Agilen Methoden sind unvollständig und grob bekannt nicht wie die Anforderungen in Klassischen Methoden. Allerdings enthält Scrum (als Agile Methode) keine Konkreten Vorgaben, wie die Anforderungen an die zu entwickelnde Software erfasst und spezifiziert werden sollen, dies wird mit Verwendung der User Stories erfüllt. Trotzdem enthält die Anforderungsformulierung mittels User Stories viele Probleme. In dieser Arbeit wird gezeigt, wie gute User Stories geschrieben werden und welche Probleme während des Projektablaufs auftreten können. Die Umsetzung einer guten formulierten User Stories in lieferbares Produkt führt zur erhöhen die Kundenzufriedenheit und dadaruch könnten Softwareentwickler schnell und zuverlässig gute Software bereitstellen.

1 Einleitung

Die Anforderungen in Agilen Methoden sind unvollständig und grob bekannt nicht wie die Anforderungen in Klassischen Methoden. Ein klassisches Projektmanagement ist klar aufgebaut. Zu Beginn dem Projektablauf werden Ziele formuliert, der Verlauf wird in einzelne Phasen mit Zwischenschritten gegliedert. Anforderungen und Umsetzungen der gesetzten Ziele werden bereits frühzeitig sehr klar ausgearbeitet und festgehalten, deswegen werden die Änderungen als Ausnahme betrachtet. Im Gegensatz wirkt Agiles Projektmanagement auf den ersten Blick geradezu konfus und planlos. Ansätze wie Scrum gehen davon aus, dass gerade IT-Projekte so komplex und schnell veränderlich sind, dass sie nicht bereits am Tag des Projektstarts komplett linear festgehalten werden können. Es gibt immer neue Anforderungen, die oftmals erst während des Projektablaufs auftreten, erfordern kurzfristige Planungskorrekturen. Während des Projektablaufs mit Agilen Methoden werden User Stories häufig verwendet, um Anforderungen an das Softwareprodukt in schriftlicher Form festzuhalten. Es ist lange bekannt, dass Fehler im klassischen Vorgehen zu Problemen während und nach einem Projekt führen können. Trifft dies auch für agile Methoden und User Stories zu? In dieser Arbeit wird gezeigt, wie eine gute User Story aufgebaut wird, der Unterschied zwischen einer guten und einer schlechten User Story und als Resultat wird gezeigt, ob die Verwendung von User Stories eine Garantie für Problemvermeidung ist.

2 User Story

Eine User Story ist das wichtigste Werkzeug, um agile Projekte inhaltlich zu steuern. Sie transportiert die Wünsche des Kunden an ein Softwareprodukt. Der Product Owner beschreibt, wie sich Anwender eine Software wünschen oder Kunden einen Liefergegenstand eines agilen Projekts, um ihre Geschäftsziele zu erreichen. User Stories sind nicht zu verwechseln mit Use Cases. Letztere kommen aus der klassischen Softwareentwicklung und stellen ebenfalls Anforderungen in natürlicher Sprache und im Kontext der Nutzer dar. Der Unterschied ist jedoch, dass Use Cases alle Erfolgs- und Misserfolgsszenarien des formulierten fachlichen Problems darstellen. Das Use Case bildet somit den gesamten Rahmen und beinhaltet mehrere Szenarien, die zu einer bestimmten fachlichen Anforderung passen. Eine User Story hingegen beschreibt ein ganz konkretes Szenario.

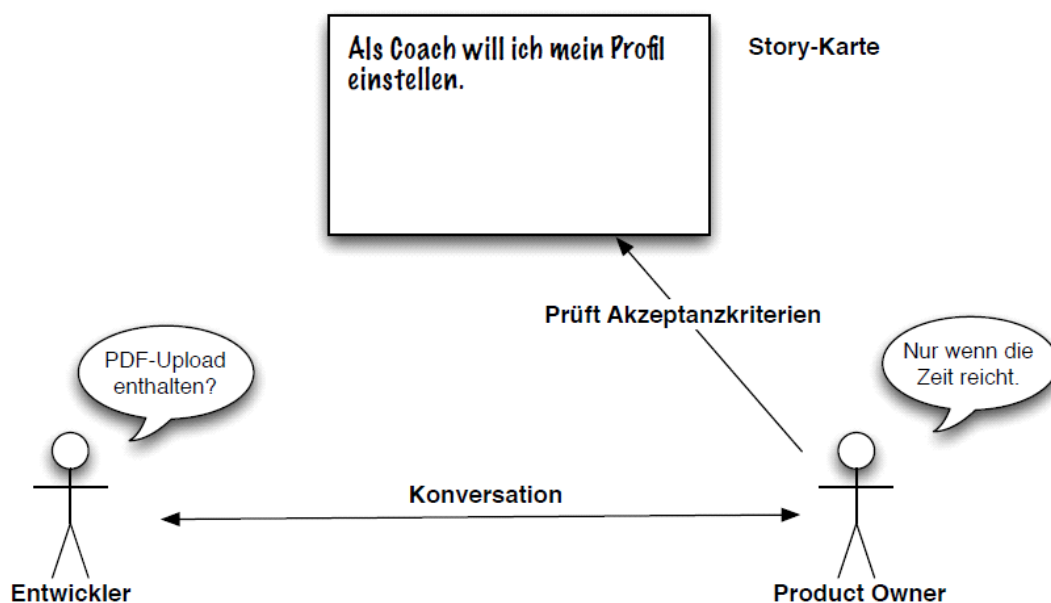


Abbildung 4.1 Karte, Konversation, Akzeptanzkriterien

Bei Erstellung von User stories sind die drei Cs zu beachten (Card, Conversation, Confirmation).

Card

User Stories sollen kurz und prägnant sein. Sie sollen in erster Linie an den Kundenwunsch erinnern und die wichtigsten Punkte, die mit dem Softwareentwicklungsteam vereinbart wurden, festhalten. Viele agile Software-Entwicklungsteams verwalten Stories und Tasks offline, z.B. in Form eines Taskboards. Die Kundenwünsche werden dann auf Karteikarten festgehalten.

Conversation

Die Kunden und das Team sollten mindestens einmal über jede Story diskutieren. Der Austausch über die Story ist erfahrungsgemäß ein sehr wichtiger Prozess. Es reicht nicht, auf einer Karte die Anforderungen festzuhalten. Stories werden in der Regel sogar mehrmals besprochen.

Confirmation

Für jede Story werden Akzeptanzkriterien vereinbart, um nachzuweisen, dass die Stories in der gewünschten Form implementiert worden sind: Der Kunde definiert vor Beginn der Umsetzung einer Story die zentralen Kriterien, nach denen die Abnahme der Story später erfolgen soll. Hierbei bietet sich die Implementierung von Akzeptanztests an, um die Erfüllung der Akzeptanzkriterien sicherzustellen.

2.0.1 Eigenschaften eine gute User Stories:

Im besten Fall sollte jede User Story nach dem INVEST-Prinzip geschrieben werden:

Independent (unabhängig): Das bedeutet, dass die User Story unabhängig von anderen sein soll.

Negotiable (Verhandelbar): Das bedeutet, dass User Stories keine Verträge sind, sondern dienen als Gesprächsgrundlage und können gemeinsam weiterentwickelt werden.

Valuable (Nützlich): Das bedeutet, dass die User Story immer einen Vorteil für den Nutzer oder den Auftraggeber darstellt.

Estimable (Schätzbar): Das bedeutet, dass die User Story nicht unkonkret ist und das Umsetzungsteam hat die Erfahrung diese zu schätzen.

Small (Klein): Das bedeutet, dass die User Story die richtige Größe besitzen muss. Sie passt also in einen Sprint. Alles was nicht in einen Sprint passt, ist erstmal ein Epic und muss dann weiter verteilt werden. Das kann im Backlog Refinement durchgeführt werden.

Testable (Testbar): User Stories sollen testbar sein. Testbare User Stories haben klar definierte Akzeptanzkriterien, die bestimmen, wann die Story fertig ist. JE geringer die Testbarkeit einer Story, desto größer ihr Testaufwand, und desto schwieriger ist zu bestimmen, wann die Story fertig ist.

3 User Story Smalls:

3.1 Zu große User Story

Eine zu große User Story bedeutet, dass die umzusetzende Funktionalität und Komplexität einer User Story direkt ihre Größe beeinflussen. Eine zu große User Story weist Komplexitäten und Abhängigkeiten auf. Da zu große Story die Simplizität einer idealen User Story überschreitet, bereitet sie somit Schwierigkeiten beim Planen. Als Indikatoren für dieses Problem nennen wir die Überdurchschnittliche Anzahl an Tasks, überdurchschnittliche Anzahl an Story Points, fehlende Priorisierung, fehlende Schätzung und Unklarheit über die Umsetzung.

3.1.1 Unterschied zwischen kleine und große User Story:

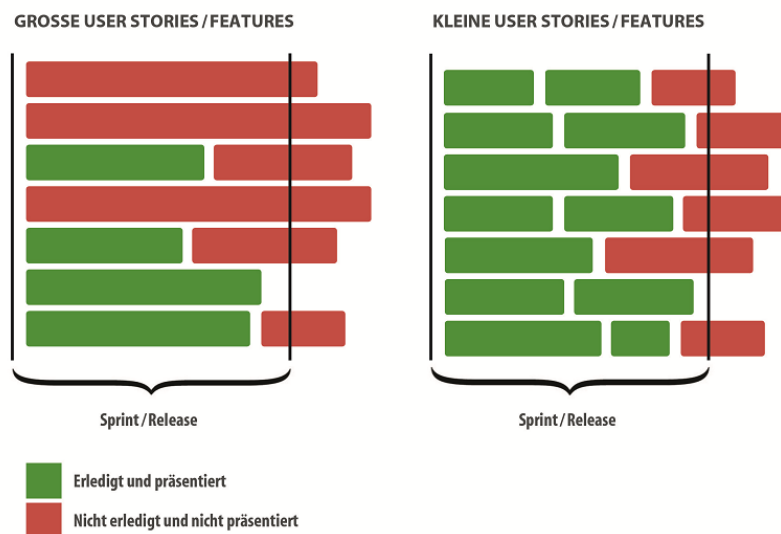


Abbildung 1: der Unterschied zwischen große und klein formulierte User Stories im gleichen Sprint

Wir haben eine gleiche Menge von User Stories bei linken Abbildung wie auch bei der rechten Abbildung in eine gleiche eingeplante Sprint umgesetzt. Die große formulierte User Stories werden nach dem Sprint nicht vollständig und konkret umgesetzt, und führt zu späterem Anwenderfeedback. Im Gegensatz sind die in kleinen User stories in eingeplante Sprint konkret und werden Problemlos umgesetzt und führt zu besserer Qualität durch regelmässige ausgelastete Tester und hat eine motivierende Wirkung auf die Entwickler.

3.1.2 Lösung für zu große User Story:

„INVEST“-Ansatz beachten Das heißt prüfen, ob die User Stories dem „INVEST“ Ansatz entsprechen.

User Story aufteilen

Zu große User Stories sollten nach Funktionen, Aktionen oder Operationen aufgeteilt werden, die der Benutzer separat ausführen kann.

User Story-Splitting wird nicht nur von Produkt Owner formuliert, sondern soll das Team auch miteinbeziehen.

Es gibt viele Möglichkeiten, um eine große User Stories aufzuteilen, wie zum Beispiel die Aufteilung nach Komplexität, Performance, Aufwände oder Datenstrukturen.

Komplexität verringern

Optionen und Komplexität sollten verringert werden. Es ist möglich, mit einem schlichten und einfachen Design zu starten.

3.2 Sprachliche Schwächen:

Die User Story enthält sprachliche Schwächen. Dies sind einzelne Wörter oder Formulierungen, die Mehrdeutigkeit und Fehlinterpretation beim Leser verursachen. Sprachliche Schwächen werden durch den Schreibstil des Autors und dessen sprachliche Qualifikationen eingeführt, was zu den verschiedensten Ausprägungen sprachlicher Qualität in User Stories führen kann.

4 Resultat:

Es gibt viele Probleme, die während des Projektsablaufs auftreten können wie zum Beispiel Sprache schwächen bei der Schreiben von User Story , kein Wert von User Story oder zu große User Story. Es werden die Lösung für die erwähnte Probleme vorgeschlagen zum Beispiel die User Story dem „INVEST“ Ansatz entsprechen und wie werden die Komplexität von User Story verringert. Doch die Verwendung von User Story ist keine Garantie für Problemvermeidung in Softwareentwicklung.

5 Zusammenfassung:

In diesem Papier werden die Nutzeranforderungen in Agilen mittels User Stories erläutert. Es ist schon bekannt, dass Fehler in Klassische Vorgehen zu Problemen führen können. Agile Methoden mittels User Stories sind auch sicher kein Allheilmittel zur Lösung aller Probleme bei der Formulierung der Nutzeranforderungen. Es kommt ganz wesentlich auf die Gegebenheiten des einzelnen Softwareentwicklungsprojekts an. Denn es gibt nicht die eine optimale Methode für alle Projekte, sondern immer nur eine passende Methode oder Methodenkombination für ein einzelnes Projekt.

Literatur

- [1] scrum mit user stories :Ralf Wirdemann,Johannes Mainusch
- [2] scrum verstehen und erfolgreich einsetzen :stefan roock, henning wolf
- [3] www.business-wissen.de/artikel/scrum-so-erstellen-sie-gute-user-stories/
- [4] www.think-pi.de/Scrum-User-Stories
- [5] www.estategy-magazin.de/einfluesse-der-user-story-qualitaet-auf-den-scrumprozess.html
- [6] swissq.it/wp-content/uploads/2018/02/Sprint.jpeg