

Feedback: Grundstein der agilen Softwareentwicklung

12 Januar 2018 | **Software Engineering** | [Markus Flückiger](#)

Lesezeit: 4 Minutes

Wendet man Prinzipien der agilen Softwareentwicklung als Leiter für einen Kurs an, entstehen neue Erkenntnisse. Beispielsweise ein essentielles Prinzip und vier Kerntechniken für den Umgang mit Anforderungen in der agilen Softwareentwicklung.

Im Kurs „Mastering Requirements in Agile Software Development“ der Zühlke Academy nutzen wir Techniken für die Moderation des Kurses, die auch die Teilnehmer selbst erlernen. Das spannende daran ist, dass wir bei der Gestaltung des Kurses dadurch zu ganz neuen Erkenntnisse über den Kursinhalt gelangen.

Ein Produkt strategisch entwickeln

Eine Technik, die wir für die Steuerung des Kurses verwenden, ist [User Story Mapping](#). Für Software Engineering bietet User Story Mapping eine ausgezeichnete Übersicht über das zu erstellende Produkt. Fast noch wichtiger ist aber die Strategie, die mit User Story Mapping verbunden ist und dem Team hilft, das zu erstellende Produkt stückweise aufzubauen. Es ergibt nämlich durchaus Sinn,

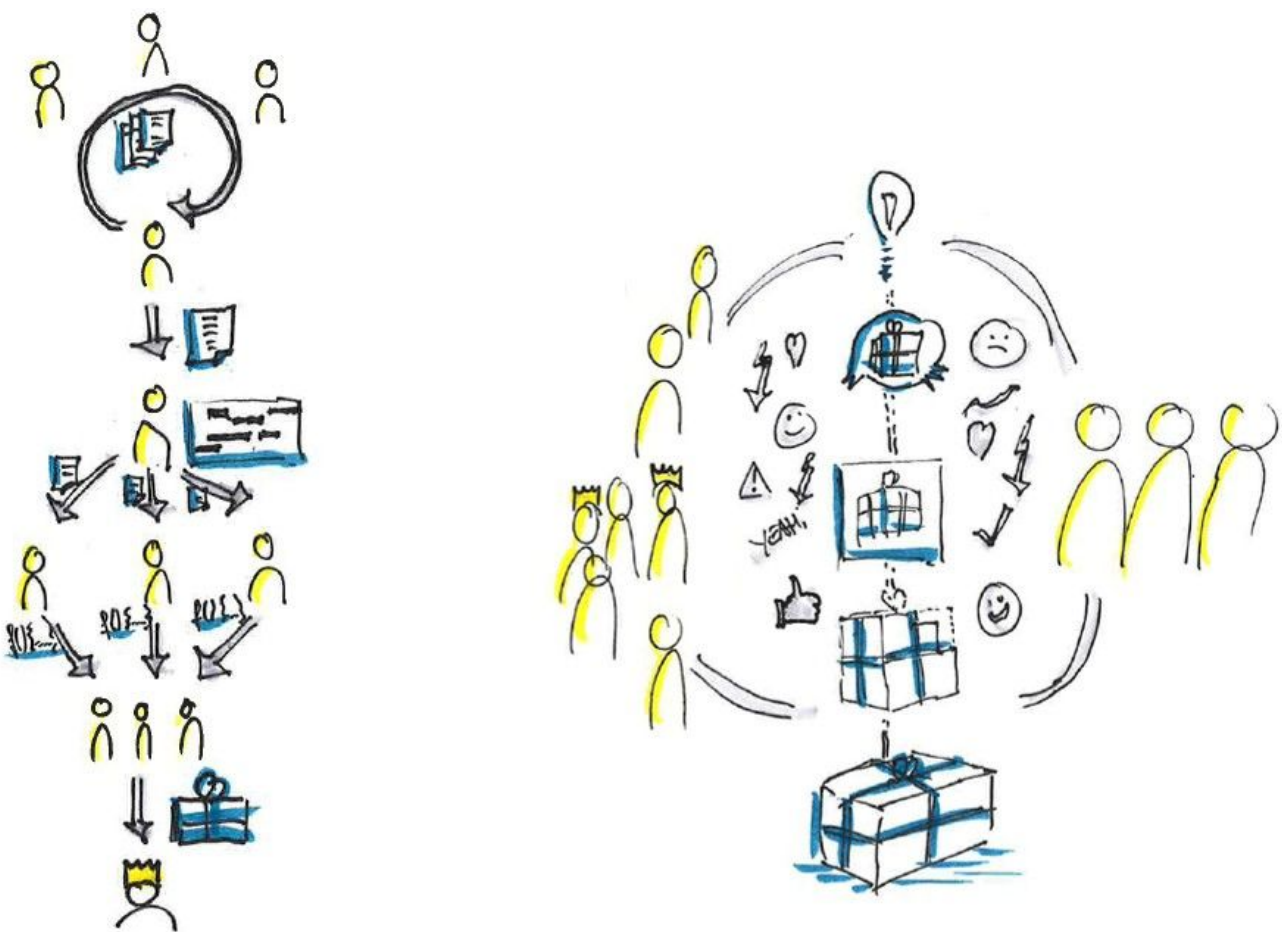
- zuerst mit dem Kern, dem wirklich Essentiellen der Lösung, zu beginnen. Das dabei entstehende klapprige Skelett lässt sich bereits anfassen und ausprobieren. Doch noch ist der Nutzen eher bescheiden und die Praxistauglichkeit nicht gegeben.
- Dieses klapprige Skelett lässt sich zu einem ersten Beta-Produkt erweitern, welches von einem ausgewählten Publikum bereits eingesetzt wird.
- Von nun an wächst das Produkt an den Erfahrungen aus der Benutzung. Es wird mit jedem weiteren Release für immer mehr Personen attraktiv und für weitere Zwecke einsetzbar. So weit so gut. Wie lässt sich dieses Prinzip auf einen Kurs anwenden? Wie ist das mit dem inkrementellen Aufbau und dem klapprigen Skelett eines Kurses? Was ist überhaupt das Produkt eines Kurses?

Diese Fragen mögen abgehoben klingen, doch so irrelevant sind sie gar nicht. Denn auf der Suche nach dem klapprigen Skelett und dem ersten brauchbaren Produkt für einen Kurs zum Thema Anforderungen stösst man unweigerlich auf wirklich wesentliche Erkenntnisse über den Umgang mit Anforderungen.

Wie wir kommunizieren

Also frisch ans Werk: Das Produkt des Kurses ist wohl die individuelle Veränderung, die der Kurs bei den Teilnehmern erzeugt. Im Kurs probieren wir hauptsächlich Techniken und Methodik aus, wie ein agiles Software-Team besonders effektiv mit Anforderungen umgehen kann. Die Veränderung beinhaltet also Konzept- und Methodenwissen, Fähigkeiten, Verhaltensweisen, persönliche Erkenntnisse und mehr.

Wir beginnen den Kurs mit zwei Mustern, wie in einem Team Informationen kommuniziert werden.



Informationen fließen in klassischen und agilen Projekten sehr unterschiedlich.

Das linke skizziert den klassischen Aufbau, wie er sich in der jahrelangen Praxis mit dokumentenzentrierten und stark arbeitsteiligen Entwicklungsprozessen etabliert hat. Business Analysten und Requirements Engineers holen Informationen von Stakeholdern ab und erstellen eine Anforderungsdokumentation. Hier schreiben sie fest, was die Software können soll. Die Dokumentation wird dann vom Projektleiter in Arbeitspakete unterteilt und stückweise an die Personen im Team zur Umsetzung verteilt. Das Ergebnis wird schliesslich

den Kunden übergeben.

Die Feedbackschleife: das Schwungrad in der agilen Entwicklung

Die rechte Seite ist eine Illustration eines Ansatzes, wie ihn Teams in einem agilen Umfeld anstreben. Im einfachsten Fall diskutieren hier Stakeholder, Analysten, Entwickler und Tester auf allen Detailstufen miteinander, was den eigentlich am gescheitesten erreicht werden soll und was die beste Lösung dafür ist. Die Lösung und das konkret zu lösende Problem entwickelt sich beim Bauen und Austesten von explorativen Prototypen, Produktinkrementen aber auch fertigen Releases.

Der Informationsfluss ist grundsätzlich anders. Links steht eine Art Softwarefabrikation. Hier müssen die richtigen Informationen eruiert und eingespielen werden und dann purzelt nach etlichen Produktionsschritten die gewünschte Software heraus. Rechts steht eine Art Problemlösungsschwungrad: Angetrieben durch die Interaktion mit den Stakeholder nähert sich das Team iterativ dem relevanten Problem und der dazu passenden Lösung an.

„Ich war zu Beginn sicher, dass wir wie auf der rechten Seite gezeigt aufgestellt sind. Aber je länger wir diskutieren, desto klarer wird mir: Wir sind noch fast vollständig auf der Linken! Das äussert sich schon darin, dass unsere Business Analysten die User Stories im Detail beschreiben und die Entwickler diese dann nur noch umsetzen.“ Kursteilnehmer

Das klapprige Skelett für den Umgang mit Anforderungen in der agilen Softwareentwicklung ist – so meine Vermutung – in diesem Grundmuster für den Informationsfluss zu sehen: Das Schwungrad bzw. eine Feedbackschleife zwischen Team und Stakeholder.

Kerntechniken

Wie funktioniert dieses Schwungrad konkret? Was sind die wichtigsten Techniken dazu und was wäre ein erstes nützliches Kursprodukt? Hier ist eine Auswahl von vier wirklich essentiellen Techniken:

1. **User Stories:** Die am häufigsten eingesetzte Technik sind die allseits bekannten User Stories. Tatsächlich sind die User Stories da, um das Schwungrad zu lenken und nicht etwa, um Anforderungen festzuhalten, wie das in der Praxis gelegentlich genutzt werden. Mit User Stories definiert das Team in erster Linie die Reihenfolge, mit denen die anstehenden Themen mit den Stakeholdern zusammen auf die Strasse gebracht werden sollen.
2. **User Story Mapping:** Wie Eingangs bereits erwähnt, richtet diese Technik die Produktentwicklung an einer Strategie aus.
3. **Inception Deck:** Das Inception Deck steht als Repräsentant für verschiedenen alternative Techniken, die den Kern des zu entwickelnden Produkts auf den Punkt bringen. Das Inception

Deck stellt das gemeinsame Ziel des Teams dar.

4. **10m Wand:** Schliesslich gilt es noch die 10m Wand zu nennen. Denn auf dieser wird der aktuelle Stand der Kommunikation für alle sichtbar, fassbar und somit gemeinsam diskutierbar.

So wäre meiner Meinung ein erstes für ausgewählte Personen nützliche Kursprodukt in einer Betaversion da: Das Prinzip der Feedbackschleife, Inception Deck, User Story Mapping, User Stories und 10m Wand. Wer diese 5 Dinge vernünftig anwenden kann, wird sehr viel effektiver mit Anforderungen in agilen Projekten umgehen können.

Auf einem solchen Betaprodukt lässt sich nun gut aufbauen. Techniken des User Centred Designs, der Business Analyse, des Design Thinkings, des Software Engineerings und aus vielen weiteren Quellen ergänzen nun situationsgerecht.

Ich freue mich schon jetzt auf die angeregten und spannenden Diskussionen, die wir in den kommenden Durchführungen dazu führen werden.