

SCRUM - die agile Projektmanagementmethode im Überblick

1. ALLGEMEIN

Scrum (engl. das Gedränge) ist eine Sammlung von Arbeitstechniken, Strukturen, Rollen und Methoden für das Projektmanagement im Rahmen agiler Softwareentwicklung. Es ist ein Vorgehensmodell, das wenige Festlegungen trifft. Teams bzw. Entwickler organisieren sich weitgehend selbst und wählen auch die eingesetzten Methoden. Das Vorgehen und die Methoden werden fortlaufend aktuellen Erfordernissen angepasst. Ken Schwaber, Jeff Sutherland und Mike Beedle haben Scrum im Bereich Software-Entwicklung etabliert. Als Methode zur Software-Entwicklung wurde Scrum das erste Mal in dem Buch "Wicked Problems, Righteous Solutions" als Software-Entwicklungsmethode beschrieben. Scrum in Produktionsumgebungen wird zum ersten Mal in dem Artikel "The New New Product Development Game" erläutert und später in "The Knowledge Creating Company" weiter ausgeführt von Ikujiro Nonaka und Hirotaka Takeuchi.

2003 legte Ken Schwaber ein Zertifizierungsprogramm für Scrum Master auf. Das Ziel, heute wie damals ist es, die Software-Entwicklung durch das Nutzen von Scrum zu professionalisieren. Inzwischen wird das Training "Certified Scrum Master" unter der Schirmherrschaft der Scrum Alliance durchgeführt.

Mittlerweile ist Scrum als agile Methode für Projektmanagement in anderen Branchen etabliert wo einerseits Grundlagenentwicklung betrieben wird oder andererseits eine hohe Volatilität in den Projektanforderungen und im Projektumfeld vorhanden sind. Als Beispiele aus anderen Branchen sind dabei zB im Medienbereich die Produktion von animierten Spielfilmen, in der Biotechnologie die Entwicklung und Produktion von Geruchs- und Geschmacksstoffen sowie im Bereich Automotive die Grundlagenforschung für neue Oberflächenbehandlungen (zB Farbbeschichtung von Blechen zur Vermeidung von Lackierprozessen).

2. GRUNDANNAHMEN

Scrum knüpft an viele Grundannahmen einer schlanken Produktion (engl. lean production) an und überträgt Erfahrungen aus der Automobilbranche auf die Softwareentwicklung. In der Automobilbranche gilt die Firma Toyota als ein Vorreiter der "schlanken" Produktion. Das sich ständig in der Weiterentwicklung befindende Toyota Production System (TPS) umfasst dabei Methoden und Arbeitsmittel der Produktion und steht im selben Verhältnis zum Toyota Way, der Philosophie hinter dem TPS, wie die Scrum-Methodik zur agilen Softwareentwicklung. Im Mittelpunkt steht bei beiden Systemen die ständige Weiterentwicklung der Mitarbeiter, der Herstellungsprozesse, der Arbeitsmittel und Methoden, mit gleichzeitig konstantem Beibehalten der Grundannahmen, die dahinter stehen. Eine Gemeinsamkeit in den meisten agilen Vorgehensmodellen ist dabei die gleichzeitige Weiterentwicklung und Professionalisierung aller an dem Prozess Beteiligten, daher auch der Kunden und Partner. Ziel der Grundannahmen ist es, die Produktion ständig zu verbessern, um höchste Qualität bei niedrigstem Aufwand zu erreichen.

Bei Scrum wird grundsätzlich angenommen, dass Produktfertigungs- und Entwicklungsprozesse so komplex sind, dass sie sich im Voraus weder in große abgeschlossene Phasen, noch in einzelne Arbeitsschritte mit der Granularität von Tagen oder Stunden pro Mitarbeiter vorherplanen lassen. Somit ist es produktiver, wenn sich ein Team in einem festen äußeren Rahmen mit sehr grober Granularität selbst organisiert. Dieses selbstorganisierte Team übernimmt in diesem, mit dem Product-Owner abgestimmten, Rahmen die gemeinsame Verantwortung für die Fertigstellung der selbstgewählten Aufgabenpakete. Dabei werden traditionelle Werkzeuge, z. B. zur Kommunikation oder Projektsteuerung sowie durch das Management „von oben“, für die Teamstrukturierung festgelegter Prozesse, die die Zusammenarbeit im Team kontrollieren und regulieren, abgelehnt.

Scrum basiert als agile Methode auf 'Werten', die 2001 im Agilen Manifest u.a. von Kent Beck, Alistair Cockburn, Ward Cunningham und Martin Fowler formuliert wurden:

1. Individuen und Interaktionen gelten mehr als Prozesse und Tools.
2. Funktionierende Programme gelten mehr als ausführliche Dokumentation.

3. Die stetige Zusammenarbeit mit dem Kunden steht über Verträgen.
4. Der Mut und die Offenheit für Änderungen steht über dem Befolgen eines festgelegten Plans.

Im Folgenden werden Kernpunkte der Scrum-Arbeitsmittel wiedergegeben:

3. SCRUM-ROLLEN

3.1. Allgemein

Bei Scrum gibt es drei klar getrennte Rollen, die von Mitarbeitern ausgeführt werden, die am gleichen Projekt zusammen arbeiten und damit auch das gleiche Ziel haben. Damit jeder für das, was er kann, zuständig und verantwortlich ist, werden die Zuständigkeiten wie folgt aufgeteilt:

3.2. Product Owner

Der Product Owner legt das gemeinsame Ziel fest, dass das Team erreichen muss. Er stellt das Budget zur Verfügung. Er setzt regelmäßig die Prioritäten der einzelnen Product-Backlog-Elemente. Dadurch legt er fest, welche die wichtigsten Features sind, aus denen das Entwicklungsteam eine Auswahl für den nächsten Sprint trifft.

3.3. Team

Das Team schätzt die Aufwände der einzelnen Backlog Elemente ab, und beginnt mit der Implementierung der für den nächsten Sprint machbaren Elemente. Dazu wird vor dem Beginn des Sprints ein weiteres Planungstreffen durchgeführt, bei dem die höchst priorisierten Elemente des Backlogs und konkrete Aufgaben aufgeteilt werden. Das Team arbeitet selbstorganisiert im Rahmen einer Time Box (dem Sprint), und hat das Recht (und die Pflicht), selbst zu entscheiden, wieviele Elemente des Backlogs nach dem nächsten Sprint erreicht werden müssen, man spricht dabei von commitments.

3.4. Scrum Master

Der Scrum Master hat die Aufgabe, die Prozesse der Entwicklung und Planung durchzuführen und die Aufteilung der Rollen und Rechte zu überwachen. Er hält die Transparenz während der gesamten Entwicklung aufrecht, und fördert das Zu-Tage-Treten der bestehenden Verbesserungspotentiale. Er ist keinesfalls für die Kommunikation zwischen Team und Product Owner verantwortlich, da diese direkt miteinander kommunizieren. Er steht dem Team zur Seite, ist aber weder Product Owner noch Teil des Teams. Der Scrum Master sorgt mit allen Mitteln dafür, dass das Team produktiv ist, also die Arbeitsbedingungen stimmen und die Teammitglieder zufrieden sind.

Bei der Rollenaufteilung wurde berücksichtigt, dass das Team sich selbst organisiert. Der Product Owner gibt nicht vor, welches Teammitglied wann was macht und wem zuarbeitet. Bei Scrum wird von der Annahme ausgegangen, dass das Team sich intuitiv selbst organisiert, und zu jeder Aufgabe dynamisch eine best angepasste innere Organisationsstruktur bildet, die sich relativ schnell an die sich wandelnden komplexen Aufgaben anpasst. Der Scrum Master hat die Pflicht, darauf zu achten, dass der Product Owner nicht in diesen adaptiven Selbstorganisationsprozess eingreift und das Team stört oder Verantwortlichkeiten an sich nimmt, die ihm nicht zustehen.

4. ARTEFAKTE & MEETINGS

4.1. Product Backlog

Das Product Backlog enthält funktionale und nicht funktionale Anforderungen des zu entwickelnden Produkts. Es beinhaltet alle Funktionalitäten, die der Kunde wünscht, zuzüglich technischer Abhängigkeiten. Vor jedem Sprint werden die Elemente des Product Backlogs neu bewertet und priorisiert, dabei können bestehende Elemente entfernt sowie neue hinzugefügt werden. Hoch priorisierte Features werden von den Entwicklern im Aufwand geschätzt und in den Sprint Backlog übernommen.

Priorität	Backlog Item	User Story	Akzeptanzkriterien	Story points
1	Passagierliste	Als Personal am Check-in Schalter möchte ich die Fluggäste namentlich für das weiterführende Boarding erfassen können.	Alle erfassten Passagiere müssen am Boarding-Schalter mit Namen, Klasse, und Sitzplatz identifizierbar sein	8
1	Boarding	Als Boarding-Personal muss ich eine ziffernmässige und namentliche Kontrolle der Passagiere innerhalb von längstens 15 Minuten durchführen können.	Jede mögliche Ausnahme darf keine Verzögerung der Maximalzeit (15 Min.) nach sich ziehen.	13
2	Unterweisungszyklus	Als Board-Personal möchte ich eine Checkliste der Unterweisungspunkte haben (z.B. Anlegen der Schwimmwesten) die jedoch situativ änderbar sein muss.	Geänderte Aspekte (z.B. Verhalten bei Terrorangriffen) müssen innerhalb von 48 Std. in den Unterweisungszyklus einzubinden sein.	3

Abb. 1.: Beispiel eines Product Backlogs

4.2. Sprint Backlog

Das Sprint Backlog enthält alle Aufgaben, die notwendig sind, um das Ziel des Sprints zu erfüllen. Eine Aufgabe sollte dabei nicht länger als 2 Arbeitstage dauern. Längere Aufgaben sollten in kurze Teilaufgaben zerlegt werden. Bei der Planung des Sprint werden nur so viele Aufgaben eingeplant, wie das Team an Kapazität aufweisen kann. Die Kapazität berechnet sich gemäss folgender grober Formel: Kapazität (in Stunden) = Arbeitstage × Anzahl Personen × 7 h. Die Planung des jeweiligen Sprints wird in Form eines zweistufigen "Sprintplanung Meetings" durchgeführt. Im Teil 1 gibt der Product Owner das Sprintziel vor. Daran orientiert sich das Team und sucht sich eigenständig jene User Stories aus dem Product Backlog (zumeist die am höchsten priorisierten) heraus, die dem Ziel des kommenden Sprints entsprechen und nimmt sie damit im Sprint Backlog auf. Anschliessend zieht sich im Teil 2 des "Sprintplanung Meetings" das Team zurück und plant eigenständig die Tasks. Das Team bricht sozusagen die ausgewählten User Stories in kleinere Arbeitseinheiten herunter.

Das Sprint Backlog, als Visualisierungs Board ausgeführt besteht aus den Spalten: Backlog Item (User Story), Tasks open, in Arbeit, erledigt, abgeschlossen und abgenommen. Zusätzlich werden die Tasks nochmals neu geschätzt. Die daraus resultierende Summe pro User Story muss nicht unbedingt der geschätzten Anzahl von Story Points im Product Backlog entsprechen. Die Differenz entsteht durch die höhere Genauigkeit der Tasks im Sprint Backlog und dient als weitere Kennzahl für die Ermittlung der Geschwindigkeit.

Backlog Item	Story Points	Offen	In Arbeit	Erledigt	Abgenommen
Passagier-liste	1				Bildschirmmaske (AK)
	3		Datenbank-Abfrage Routine (MB)		
	2			Login-Prozedur (AK)	
	2	Inbound-Dokumentation			
Boarding	

Abb. 2: Beispiel eines Sprint Backlogs, abgeleitet vom Product Backlog (siehe Abb. 1)

4.3. Impediment Liste

In die Impediment Liste werden alle Hindernisse des Projekts eingetragen. Der Scrum-Master ist dafür zuständig, diese gemeinsam mit dem Team auszuräumen.

Datum des Auftretens	Thema	Lösungsbeschreibung	Datum der Lösung
....
12.03.2009	Eine Änderung der bestehenden „Check-In“ Software ist teurer als geplant.	Eine Umprogrammierung der Datenbank ist lt. Aussage des Herstellers doch nicht notwendig	18.03.2009
25.03.2009	Kathy fällt für 3 Wochen als Notersatz für ein dringendes Kundenprojekt überraschend aus.	Eine dadurch ausgelöste Verlängerung des Prozessprojektes wurde durch den Vorstand genehmigt	25.03.2009
29.03.2009	Das angemietete Areal für die Simulation der Boarding Prozesse steht nicht mehr zur Verfügung	Ein Ersatz wurde 2 km- ausserhalb der Stadt gefunden. Die längere Fahrtzeit verzögert in Summe um 1 Tag	01.04.2009
....

Abb. 3: Beispiel einer Impediment Liste

4.4. Sprint

Zentrales Element von Scrum ist der Sprint. Ein Sprint bezeichnet die Umsetzung einer Iteration, Scrum schlägt ca. 30 Tage als Iterationslänge vor. Vor dem Sprint werden die Produkt-Anforderungen des Kunden in einem Product Backlog gesammelt. Auch technische und administrative Aufgaben (funktionale und nichtfunktionale Anforderungen) werden dort aufgenommen. Das Product-Backlog muss nicht vollständig sein; es wird laufend fortgeführt. Die Anforderungen für den ersten Sprint sind meistens rasch aufgestellt. Die Anforderungen werden informell skizziert. Für einen Sprint wird ein Sprint-Backlog erstellt, in diesen werden Anforderungen übernommen, die während des Sprints umgesetzt werden sollen. Die Entscheidung, welche Anforderungen umgesetzt werden, wird vom Kunden nach von ihm festgelegten Prioritäten getroffen. Zum Sprint organisiert sich das Entwicklungsteam selbst, braucht also keine detaillierten methodischen Vorschriften.

4.5. Daily Scrum-Meeting

An jedem Tag findet ein kurzes (maximal 15-Minuten dauerndes Meeting) Scrum-Meeting statt.

Scrum definiert keine konkrete Uhrzeit für das Meeting, das Meeting sollte jedoch täglich zur gleichen Zeit stattfinden. Empfohlener Zeitpunkt für das Scrum-Meeting ist die Zeit nach dem Mittagessen, da morgendliche Scrum-Meetings oft mit flexiblen Gleitzeitregelungen kollidieren und der müde Punkt nach dem Mittagessen bei einem Scrum-Meeting, welches durchaus im Stehen abgehalten werden kann, nicht so stark ins Gewicht fällt wie bei anderen Tätigkeiten. Die Meetings sind kürzer als am Morgen, weil allgemeine Dinge und Neuigkeiten schon vorher diskutiert wurden und die Mitarbeiter mit dem Kopf ganz bei der Arbeit sind.

Im Scrum-Meeting werden vom Team Mitglied an folgende Fragen beantwortet:

- "Bist Du gestern mit dem fertig geworden, was Du Dir vorgenommen hast?"
- "Welche Aufgaben wirst Du bis zum nächsten Meeting bearbeiten?"
- "Gibt es ein Problem, das dich blockiert?"

Die Sitzung dient dem Informationsaustausch im Team. Erledigte Aufgaben und Features werden anschließend im Burndown Graph aktualisiert. Falls neue Hindernisse erkannt wurden, müssen diese vom Scrum Master bearbeitet werden. Dazu werden sie in das Impediment Backlog eingetragen.

Ein weiterer darin erstelltes Artefakt ist das Burndown Chart (siehe Abb. 4). Damit wird die aktuelle Situation des Projektes visualisiert, indem die noch verbleibenden Tätigkeiten der noch offenen Tasks abgetragen werden.

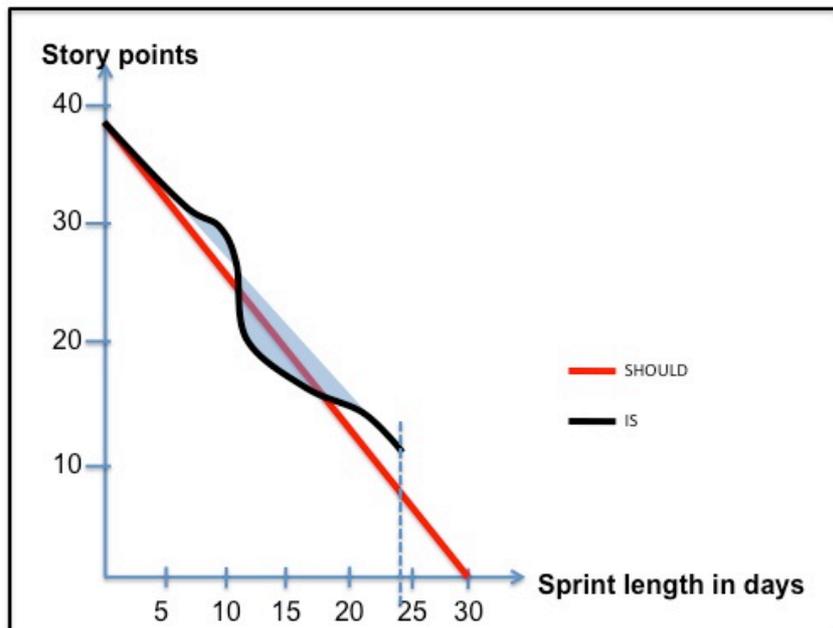


Abb. 4: Beispiel eines Burndown Charts

Größere Projekte werden können mittels Organisation des Projektes in Form mehrerer Teams modularisiert werden. Dann wird von den jeweiligen Vertretern der Teilprojekte ein „Scrum-of-Scrum-Meeting“ durchgeführt, bei dem die Ergebnisse der einzelnen Gruppen zusammengefasst werden.

4.6. Review

Nach einem Sprint wird das Sprint-Ergebnis einem informellen Review durch Team und Kunden unterzogen. Dazu wird das Ergebnis des Sprints (die laufende Software) vorgeführt, eventuell werden technische Eigenschaften präsentiert. Der Kunde prüft, ob das Sprint-Ergebnis seinen Anforderungen entspricht, eventuelle Änderungen werden im Product Backlog dokumentiert.

4.7. Retrospektive

In der Retrospektive wird die zurückliegende Sprint-Phase betrachtet. Es handelt sich dabei nicht um Lessons Learned, sondern um einen zunächst wertfreien Rückblick auf die Ereignisse des Sprints. Alle Teilnehmer notieren dazu die für sie wichtigen Ereignisse auf Zetteln und ordnen sie dem Zeitstrahl des Sprints zu. Anschließend schreiben die Teilnehmer alle Punkte auf, welche ihnen zu den Fragen "Was war gut?" (Best practice) bzw. "Was könnte verbessert werden?" (Verbesserungspotential) einfallen. Jedes Verbesserungspotential wird priorisiert und einem Verantwortungsbereich (Team oder Organisation) zugeordnet. Alle der Organisation zugeordneten Themen werden vom Scrum Master aufgenommen und in das Impediment Backlog eingetragen. Alle teambezogenen Punkte werden in das Product Backlog aufgenommen.

Wird für die Retrospektiven und deren Vorbereitung nicht genug Zeit eingeräumt, bleiben die Erkenntnisse und Ergebnisse oberflächlich und die Resultate nach jedem Sprint ähneln sich. Dann läuft man Gefahr, dass die Retrospektiven an Stellenwert verlieren oder ganz gestrichen werden, weil die Ergebnisse der Retrospektiven vorhersehbar sind.

5. WEITERFÜHENDE LITERATUR

Cohn, M. (2004): User Stories Applied. For Agile Software Development, Boston.

Gloger, B. (2008): Scrum. Produkte zuverlässig und schnell entwickeln, München.

Nonaka, I./Takeuchi, H. (1998): The Knowledge-Creating Company. How Japanese Companies Create The Dynamics Of Innovation, New York.

Ohno, T. (1988): Toyota Production System. Beyond Large Scale Production, Cambridge.

Pichler, R. (2008): Scrum. Agiles Projektmanagement erfolgreich einsetzen, Heidelberg.

Poppendieck, M./ Poppendieck, T. (2006): Implementing Lean Software Development. From Concept to Cash, New York.

Schwaber, K./Beedle, M. (2001): Agile Software Development with Scrum, New York.

Schwaber, K. (2004): Agile Project Management with Scrum, Redmond.

Schwaber, K. (2007): The Enterprise and Scrum, Redmond.

Takeuchi, H./Nonaka, I. (1986): The New Product Development Game. Stop Running The Race And Start Rugby. In: Harvard Business Review. Jan-Feb 1986, S. 137-146, Boston.

Womack, J., P./Jones, D., T. (1996): Lean Thinking, Auf dem Weg zum perfekten Unternehmen, 1. Aufl., München.