



Agile Verfahren – Scrum

Scrum (Gedränge) eignet sich für alle Arten von Teamwork, in denen idealerweise jeder weiß, um was es geht. Damit – das sei vorweggenommen - eignet es sich primär für interne Projekte. Wird es von Softwareentwicklern genutzt, dann z.B. für den Aufbau einer Software, mit der die eigene Arbeit erleichtert werden kann.

In diesem Zusammenhang wird Scrum gerne als agiles Projektmanagement-Framework bezeichnet. Dieses Framework umfasst Meetings und Rollen, die gemeinsam das Strukturieren und Managen der Teamarbeit unterstützen. Ebenso können dazu Tools eingesetzt werden.

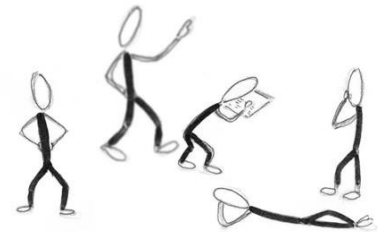
Denken wir uns folgendes Beispiel: eine Software-Firma kauft für seine 20 Mitarbeiter 5 Autos und 10 Fahrräder z.B. um Kundenbesuche durchführen oder um kleine Einkäufe zu machen. Um die Fahrzeuge zu verwalten und zu reservieren (und um das Führen eines Fahrtenbuches zu erleichtern) soll eine Webseite entwickelt werden, auf der sich die Mitarbeiter ein Fahrzeug oder Fahrrad reservieren können.

Ein anderes Szenario könnte auch die Organisation eines Betriebsausflugs sein.

In beiden Szenarien sind die Beteiligten sowohl in der Lage, sich vorzustellen, was herauskommen soll, als auch wie es durchgeführt und umgesetzt werden soll. Damit ist Scrum prima für die Planung und Umsetzung geeignet.

Scrum kennt hauptsächlich drei Rollen:

- Das Team ()
- Der Scrum-Master (führt durch das Projekt)
- Der Product-Owner (könnte als Primus inter Paris bezeichnet werden, wenn es um das Aufstellen des Backlog und der Story-Points geht)



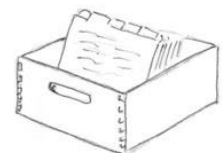
Im Folgenden soll die Umsetzung der Webseite als Leitfaden dienen.

Das Product-Backlog

Im Product-Backlog sind alle bis dahin bekannten Aufgaben/Eigenschaften schriftlich festgehalten. Ausgangspunkt sind sog. User-Stories, die im UML-Prozess als Use-Case ein Äquivalent haben.

Die Darstellung als Karteikärtchen ist bildhaft treffend. Die Stories können sein

- Ein Use-Case
- Ein funktionales Verhalten/ein Algorithmus
- Ein Oberflächenverhalten/Ansicht



Zu jeder Story (Karteikärtchen) werden zudem festgehalten

- Seine Priorität (Story-Points)
- Eine Maßnahme/eine Eigenschaft/ein Verfahren/Kriterien zur Abnahme. Dazu zählen auch relevante Testverfahren wie Unit-Tests, Integrationstests, aber auch sicherheitsrelevante Abnahmen wie statische Code-Analyse, Code Reviews usw.



Projektmanagement

Agile Verfahren - Scrum

- Einen Status (offen – angefangen – fertig)
- Ein geschätzter Aufwand / Kosten

Die Product-Roadmap

Nun wird jedoch auch bei agilen Verfahren nicht einfach wild drauflos entwickelt. Daher steht auch hinter agilen Verfahren das Gesamtziel genau wie bei traditionellen Verfahren wie dem Wasserfallmodell. Das Gesamtziel wird über die Produkt-Roadmap festgehalten. Das ist ein Projektplan, der vorgibt, wie sich ein Produkt oder eine Lösung über die Zeit entwickelt. Der Product-Owner verwendet die Roadmap zur Skizzierung der Produktfunktionen und der Release-Zeitpunkte. Die Roadmap kann und muss angepasst werden, wenn sich z.B. im Laufe der Zeit Rahmenbedingungen verändern. Das Team muss über die Roadmap laufend im Bilde sein.

Bezogen auf unser Beispiel könnte die Roadmap wie folgt aussehen:

- In 6 Monaten soll die Reservierung über eine Webseite möglich sein
- Nach weiteren 2 Monaten ist die Webseite für Mobilgeräte angepasst
- Nach weiteren 6 Monaten wird eine App fertig gestellt, die weitere Funktionen bereitstellt.

Diese übergeordneten, etwas allgemein formulierten Funktionen und Ziele werden oft auch als Epics bezeichnet.

Diese Epics werden in sogenannte User-Stories mit dem Ziel einer besseren Detaillierung zerlegt. Diese User-Stories sind Ausgangspunkt zur Entwicklung des Product-Backlogs.

Priorisierung (Story-Points)

Aufgaben sollten überschaubar sein. Dabei können Story-Points helfen, die als Schätzung für den Aufwand und/oder die Wichtigkeit einer Aufgabe dienen können.

Diese Schätzung sollte vom gesamten Team vorgenommen werden, damit jeder im Team seine Sicht auf die Aufgaben haben kann. Die Schätzung kann sich auf den Aufwand, auf die Komplexität oder die Wichtigkeit beziehen. Da jeder Auftraggeber (auch ein interner) vom Kunden wissen möchte, was das Produkt kostet, wird in aller Regel der Aufwand immer mit einbezogen. Ein abstrahiertes Punktesystem hat den Vorteil, dass tägliche oder wöchentliche Routineaufgaben keinen Einfluss nehmen können.

Schätzen mit Planning Poker (eine Variante davon ist Bucket Estimation, die besonders bei eingespielten Teams deutlich schneller ist).

Alle Aufgaben (Back-Log-Einträge) werden von allen Team-Mitgliedern geschätzt. Dabei kommt meist ein Punkte-System zum Einsatz, das einer Fibonacci-Folge entspricht, oft mit Zusatzwahlmöglichkeiten wie „Keine Ahnung/Schulungsbedarf“ oder „Kaffepause gewünscht“.

Für jede gibt es Aufgabe den gleichen Ablauf:

1. Der Back-Log-Eintrag (die Aufgabe) wird vorgestellt (vom Product Owner)
2. Alle Team-Mitglieder stellen solange Fragen, bis sie glauben, die Aufgabe verstanden zu haben
3. Jetzt schätzt jedes Team-Mitglied die Punkte, die dem Item entspricht.



Projektmanagement

Agile Verfahren - Scrum

4. Dann decken alle ihre Karte auf (Scrum Poker kann auch gut online mit Mobilgeräten durchgeführt werden)
5. Sind die Schätzungen dicht beieinander, kann z.B. das Mittel als Schätzung fixiert werden
6. Liegen die Schätzungen deutlich auseinander, diskutieren die Team-Mitglieder mit den Extremwerten (höchste/niedrigste) ihre Einschätzung aus. Die anderen beteiligen sich nicht.
7. Dann wird erneut geschätzt.
8. Nun liegt es auch am Project Owner, wie oft er den Zyklus durchlaufen will, oder ob er irgendwann einen Durchschnittswert festlegt.

Der große Vorteil dieser Methode liegt darin, dass alle Team-Mitglieder sich mit allen Aufgaben befassen müssen und damit auch mehr Gesamtverständnis für die einzelnen Teile untereinander entsteht. Zudem kann es passieren, dass vom Team eine Aufgabe als so umfangreich eingeschätzt wird, dass sie weiter zu unterteilen ist, also z.B. 2 oder 3 Aufgaben daraus zu machen sind.

Bis zu diesem Punkt unterscheidet sich Scrum nicht von Kanban. In Scrum haben die Sprints vor allem fest definierte Iterationszyklen.

Sprint-Planung und Sprints

Die eigentliche Entwicklung wird in sogenannten Sprints durchgeführt. Mit der Sprint-Planung startet der Sprint. Der Product-Owner ist für die Priorisierung von Aufgaben zuständig.

Ein Sprint ist ein festgelegter Zeitraum, in dem gearbeitet wird. Bei der Sprint-Planung wird entschieden, wie lang der Zeitraum dauern soll und evtl. welches Ziel der Sprint haben soll.

Die Aufgaben aus dem Product-Backlog können als Sprint-Backlog herausgezogen werden, damit sich jeder im Team orientieren kann, an was gerade gearbeitet wird. Eine gut gemachte Sprint-Planung kann das Team gerade so viel herausgefordert, dass es motiviert und erfolgreich arbeiten kann.

Der Product-Owner beschreibt das Ziel des Sprints aus seiner Sicht und einigt sich mit dem Team auf die Erreichbarkeit des Ziels unter Einbezug der vorgesehenen Aufgaben (dem product-Backlog). Die Einigung auf den Umfang des Sprint-Backlogs ist das wichtigste Ergebnis der Sprint-Planung.

Der Sprint

Sprints sind Zyklen mit fest vordefinierten kurzen Zeiträumen. Üblich sind Zeiträume von 1 bis zu 4 Wochen. Letztlich hängt viel an der Gesamtdauer des Projekts und der Verfügbarkeit von Ressourcen. Macht man Sprints länger, besteht die Gefahr, das Ziel aus den Augen zu verlieren. Werden sie kürzer, drohen die Meetings die Arbeit zu dominieren (man muss sich auf die Meetings ja vorbereiten).

Generell besteht ein Sprint, unabhängig von seiner Dauer, aus den folgenden Teilen:

- Sprint-Planung (siehe oben)
- Daily Scrum, eine tägliche kurze Zusammenkunft Teams
- Umsetzen der Sprint-Backlog Aufgaben



Projektmanagement

Agile Verfahren - Scrum

- Sprint Review, bzw. Sprint Retrospektive, in der das Erreichte analysiert und bewertet wird und in eine Veränderung der Aufgaben und/oder der Vorgehensweise mündet.

Zum letzten Punkt ist es essentiell, dass Änderungen der Aufgaben, der Bewertungen, der Priorisierung usw. dem Team jederzeit sichtbar gemacht werden. Dabei dient das Review eher der Vorstellung des Erreichten (Erfüllungsgrad der Aufgabe oder des Ziels), die Retrospektive eher einer qualitativen Bewertung des Vorgehens, der Planung, der Priorisierung.

Während der Sprints findet täglich ein Scrum (Daily-Scrum) statt, manchmal auch als Stand-Up bezeichnet. Das Treffen soll kurz sein, daher findet es oft im Stehen statt (kein Meeting Raum). Das Treffen dient vor allem dazu, das Team über den Fortlauf der Arbeit zu informieren. Typische Fragestellungen sind

- Was habe ich gestern getan?
- Behindert mich etwas?
- Was mache ich heute?

Der Scrum-Master ist dabei so etwas wie ein Coach des Teams. In einer ideal agilen Welt bräuchte es den Scrum-Master nicht, das Team würde sich selbst managen.

Der Scrum-Master kann z.B. die Meetings organisieren und durchführen.

In der Praxis obliegt ihm und dem Product-Owner, aus den Meetings die richtigen Schlüsse zu ziehen und ggfs. Maßnahmen zu ergreifen. Vor allem, das Team nicht zu überfordern und vor Scope Creeps zu schützen.

Diese sog. Scope Creeps sind Änderungen und Neuerungen durch den Auftraggeber. Es handelt sich eigentlich um neue Aufgaben, die zum Backlog hinzukommen. Werden sie falsch eingeschätzt und häufen sie sich, wird die geplante Roadmap schnell aus den Augen verloren.

Der Scrum-Master ist auch Ansprechpartner für Unstimmigkeiten im Team (z.B. wegen Arbeitsstilen) und er sorgt für die erwähnte Übersicht des Projekts (auch des Erledigungsgrades), um das Team auf dem Laufenden zu halten. Er kümmert sich auch um die Beseitigung externer Blockaden und die Beseitigung interner Hindernisse (z.B. defekte Infrastruktur).

So kann der Project Owner auch Maßgaben zur Umsetzung von Aufgaben vorgeben wie z.B. Pair Programming, um bei bestimmten Aufgaben den Vorteil dieses Zwiesgesprächs auszunutzen.

Nach der Beendigung eines Sprints folgt der nächste bis zur Fertigstellung des Produkts.

Sprint-Review, Sprint-Retrospektive

Wie oben bereits angedeutet, erfüllen Reviews den Zweck der Erfolgsdemonstration. Es wird gezeigt, was bislang erreicht wurde, das Team kann sich auch feiern.

Die Retrospektive hingegen dient dazu, schnell Feedback zu erhalten, um das Produkt und die Entwicklungskultur zu verbessern. Mithilfe von Retrospektiven wird vom Team dem Team aufgezeigt, was gut geklappt hat – und was verbessert werden muss.

Auch hier ist der Scrum-Master in der Pflicht, z.B. einen Aktionsplan zu entwickeln und Verbesserung zu suchen und zu fördern.