

---

Scrum und agile Softwareentwicklung

---

Drei Rollen in Scrum

---

Zwei Listen und vier Meetings

---

## Kapitel 1

# Das ist Scrum und so funktioniert es

**S**crum ist eine Methode, sehr effektiv im Team zu arbeiten und Dinge zu erledigen, die zu komplex sind, um sie auf Autopilot abzuwickeln. Das trifft auf sehr viele unserer Projekte zu. Scrum zielt konsequent auf effektive Arbeit. Scrum ist bei der Softwareentwicklung entstanden, ist aber auch in anderen Bereichen sehr nützlich.

## Scrum und agile Softwareentwicklung

Scrum wurde 1995 von Jeff Sutherland und Ken Schwaber entwickelt. Sie untersuchten, welche Prinzipien äußerst erfolgreichen Projekten zugrunde liegen. Diese Prinzipien sind letztlich die Basis für Scrum. Scrum liefert eine geringe Anzahl wirksamer Regeln, die in Situationen, in denen es keine einfache Lösung gibt, für Stabilität sorgen. Dann sind Kreativität, Engagement und Zusammenarbeit im Team nötig, um schnell zu immer besseren Lösungen zu gelangen. Also eigentlich immer.

Scrum betont die Tatsache, dass neue Erkenntnisse sich immer dann einstellen, wenn man mit der Arbeit begonnen hat, und dass ein Team so flexibel sein sollte, die besten Erkenntnisse zu seinem Vorteil zu nutzen. Scrum ist die beliebteste agile Methode. Agil ist die Philosophie, in der Flexibilität eine herausragende Rolle spielt, festgelegt im »Agilen Manifest«.

Scrum hat nur wenige Regeln, die alle auf ein paar Seiten Platz haben. (Warum dann ein ganzes Buch?) Scrum enthält zwei Listen, drei Rollen und vier Meetings. Das ist alles. Scrum ist einfach. Fast so einfach wie Schach: leicht zu erklären, schwierig zu gewinnen.

Scrum wurde bewusst klein gehalten. Dadurch ist es breit anwendbar und leicht zu memorisieren. Das Wichtigste dabei aber ist vielleicht, dass Sie immer selbst nachdenken müssen. Scrum schreibt so wenig vor, dass Sie jederzeit selbst entscheiden, wie Sie Scrum in einer bestimmten Situation anwenden. Gleichzeitig ist das wohl auch der größte Nachteil: Scrum ist nicht die Antwort auf alle Probleme, es schreibt nicht vor, was in jeder denkbaren Situation zu tun ist. Scrum ist also eigentlich keine »Methode«, die Schritt für Schritt konkrete Anweisungen gibt. Scrum ist ein Rahmen, der Ihnen hilft, schnell herauszufinden, was Sie tun müssen.

Wenn Sie die Prinzipien oder Regeln von Scrum genau einhalten, gewinnen Sie sehr schnell sehr viele Erkenntnisse, und Sie werden mit der Nase auf die Fakten gestoßen. Scrum ist der ultimative Prozess, um alle Möglichkeiten und Unmöglichkeiten sichtbar zu machen und damit das Beste aus dem Team herauszuholen. Die Scrum-Regeln schreiben vor, dass alles transparent und sichtbar gemacht wird, vom Arbeitsanfall bis zu Vereinbarungen, von Ergebnissen bis zur Zusammenarbeit, von der Planung bis zum Fortschritt.

Scrum »zwingt« Teams, die wichtigsten Dinge zuerst anzugehen, und das auf der Basis konkret sichtbarer Ziele, von denen Menschen profitieren. Scrum sorgt außerdem dafür, dass Sie über einen langen Zeitraum so weiterarbeiten können, durch einen festen Rhythmus und starke Betonung der Qualität.

## Wie funktioniert Scrum?

Scrum funktioniert in Iterationen. Eine Iteration ist ein kurzer Zeitraum von wenigen Wochen, der zu einem Ergebnis (meist zu einem Produkt) führt, das tatsächlich fertig ist. Diese Iterationen heißen *Sprints*. Jeder Sprint liefert ein echtes Ergebnis. In Scrum-Begriffen sagen wir, das Ergebnis einer Iteration sei »a potentially shippable product increment«. Mit anderen Worten: ein Teil der Lösung, der (so klein er auch sein mag) schon gut genug ist, um geliefert und verwendet zu werden. Ja, Sie lesen richtig: an Kunden und Nutzer geliefert. An echte Menschen. Das Ergebnis ist also dokumentiert, getestet, integriert und so weiter. Es muss nichts mehr daran getan werden. Vielleicht enthält es noch nicht genug Funktionalität, um wirklich verkäuflich zu sein, aber funktionieren würde es.



Ein Sprint liefert ein echtes Ergebnis. Wenn Sie also Software entwickeln, heißt das, dass Sie produktionsreife Software liefern. Wenn Sie mit Scrum ein neues Fahrrad bauen, dann führt jeder Sprint zu einem Fahrrad oder Fahrradteil, nicht nur zu Entwürfen und Dokumentationen. Scrum zwingt Sie dazu, Dinge fertigzustellen, sodass Sie Feedback zu echten Produkten bekommen. So sieht echtes Risikomanagement aus.



Arbeiten Sie lieber in kurzen als in langen Sprints. Je kürzer der Sprint, desto eher wird ein Wert geliefert, desto schneller lernen Sie und können Feedback umsetzen. In der ursprünglichen Scrum-Literatur ist von Sprints von maximal einem Monat die Rede. Zurzeit arbeiten die meisten Teams in Sprints von zwei bis drei Wochen, aber auch eine Woche ist nicht ungewöhnlich. Kurze Sprints geben Ihnen die Möglichkeit, öfter etwas zu verändern, früher auf Feedback zu reagieren und schneller besser zu werden.

## Drei Rollen

Der *Product Owner* ist die erste der drei Rollen in Scrum. Der *Product Owner* ist verantwortlich für das »Was« und vertritt alle interessierten Parteien. Was wollen wir entwickeln? Wie viel darf es kosten? Wann ist es fertig? Er legt fest, ob etwas wirklich genug Funktionalität enthält, um in Produktion zu gehen.

Um das alles richtig definieren zu können, ist es wichtig, dass der *Product Owner* auch das »Warum« erklären kann. Schließlich hat das »Was« allein selten einen Wert. Der *Product Owner* vertritt alle interessierten Parteien, die wir in Scrum *Stakeholder* nennen. Mit seinem Wissen um alle *Stakeholder* und ihre Interessen sortiert der *Product Owner* das *Product Backlog*. Das ist die vorrangige Aufgabe des *Product Owners*: dafür zu sorgen, dass es ein gutes *Product Backlog* gibt, und es nach Wert zu priorisieren.

Die eigentliche Arbeit wird vom *Team* ausgeführt, auch *Development-Team* genannt. Diese Menschen erledigen die anfallende Arbeit. Ein *Development-Team*, und also auch jedes Teammitglied, ist vollkommen engagiert. Das *Team* arbeitet zusammen, um die Ziele zu erreichen, die der *Product Owner* gesetzt hat, Sprint um Sprint. Das *Team* legt selbst fest, wie es das tut, also auch, wer was tut, und gibt selbst Schätzungen ab. Niemand sagt dem *Team*, wie es die gesetzten Ziele erreichen soll, also auch nicht, wie lange es für eine bestimmte Aufgabe brauchen darf. Es übernimmt Verantwortung für die eigene Arbeit. Diese Art Selbstbestimmung ist eine wichtige Motivation, die bei allen im *Team* Begeisterung für Scrum auslöst.

Der *Scrum Master* hat die Aufgabe, dafür zu sorgen, dass Scrum richtig angewendet wird. Anders ausgedrückt: Der *Scrum Master* achtet darauf, dass die Spielregeln eingehalten werden. Der *Scrum Master* ist wie das Öl in der Maschine, er sorgt dafür, dass alles so glatt wie möglich läuft. Störungen, die das *Team* erlebt, räumt er aus dem Weg. Außerdem erklärt der *Scrum Master* allen Beteiligten Scrum und die Konsequenzen seiner Anwendung – auch dem Management. Er tritt nach außen proaktiv auf und fördernd gegenüber dem *Team* und dem *Product Owner*. So kann der *Scrum Master* helfen, die Meetings zu organisieren, für das *Team* Dinge regeln oder den *Product Owner* bei der Formulierung der Anforderungen und Wünsche unterstützen.



Kann das Team selbst bestimmen, wie lange es dauert, bis das Produkt fertig ist? Nein, natürlich nicht. Das legt der Product Owner fest, der schließlich auch entscheidet, welche Features in welcher Reihenfolge bearbeitet werden. Wenn die Schätzungen des Teams damit nicht übereinstimmen, findet immer eine Diskussion statt, um zu sehen, ob etwas anders gemacht werden kann, und alle suchen nach einer Lösung. Nach jedem Sprint ist allerdings ein Product Increment fertig. Der Product Owner kann also nach jedem Sprint entscheiden, ob es ausgeliefert beziehungsweise releast werden soll oder nicht. Dadurch wird das Release zu einer taktischen oder strategischen Überlegung, nicht zu einer operativen. Das Produkt ist nämlich jederzeit »releaseable«.

Bei Scrum ist jeder total engagiert, nicht nur in Gedanken, sondern mit der ganzen Person, fünf Tage pro Woche. Bei Scrum arbeitet man an nur einem Projekt gleichzeitig, über ein Product Backlog, das vom Product Owner verwaltet wird. Das sorgt für viel Klarheit: Alle arbeiten gemeinsam und gleichzeitig an der Sache, die den höchsten Wert hat, und bringen sie außerdem tatsächlich zum Abschluss.

Bei Scrum arbeiten Sie mit einem festen Team, dem sogenannten *Scrum-Team*. Dieses Scrum-Team besteht aus dem Product Owner, dem Scrum Master und dem Development-Team. Alle zusammen arbeiten auf dasselbe Ziel hin: so schnell wie möglich das (Teil-)Produkt entwickeln, das den größten Mehrwert liefert und Menschen Freude bereitet. Es liegt auf der Hand, dass man ein solches Team, das gut zusammenarbeitet und immer besser wird, möglichst nicht trennen sollte.



Die Terminologie bei Scrum ist manchmal verwirrend. Mit dem Scrum-Team sind alle gemeint: Product Owner, Scrum Master und Development-Team. Dieses Development-Team wird oft auch verkürzt Team genannt. Wenn wir in diesem Buch irgendwo nur »Team« schreiben, meinen wir das Development-Team. Wenn wir wirklich alle meinen, verwenden wir den eindeutigen Ausdruck Scrum-Team.



Ist das alles? Gibt es keinen Projektmanager mehr? Nein. Scrum ist grundsätzlich anders in dem Sinn, dass wir lieber jemanden haben, der Eigentümer des Produkts wird und nicht nur Manager des Projekts. Ein feiner Unterschied vielleicht, aber ein grundlegender. Alle traditionellen Aufgaben des Projektmanagers sind noch da, aber auf die drei Rollen verteilt.

## Zwei Listen

Alle Wünsche und Anforderungen, die das Produkt beschreiben, werden in einer Liste festgehalten: dem *Product Backlog*. Diese Liste wird vom Product Owner verwaltet und ist immer nach dem Geschäftswert geordnet und mit Schätzungen des Teams versehen. Der Fortschritt bei der Realisierung des Produkts oder des Release wird vom Product Owner in einem Release Burndown Chart festgehalten.

Bei jedem Sprint wird der obere Teil des Product Backlogs auf das Sprint Backlog übertragen, das vom Team in kleine Abschnitte unterteilt wird, um die Arbeit während des Sprints aufteilen und den Fortschritt verfolgen zu können. Dabei führt das Team einen Sprint Burndown Chart.



Gibt es also zwei Burndown Charts? Genau, eine pro Sprint für das Team, eine pro Release für den Product Owner.

## Vier Meetings

Es gibt pro Sprint vier Meetings. Das sind alles »Timeboxen«. Das heißt, es wird vorher eine maximale Zeitmenge für das Meeting reserviert, die nicht überschritten wird. Früher fertig werden ist gut. Als Faustregel dauert ein Meeting nie länger als zwei Stunden für einen Sprint von zwei Wochen.

Das erste Meeting zu Beginn eines Sprints ist das *Sprint Planning Meeting*. Es besteht aus zwei Teilen: Der erste Teil dient dazu, festzulegen, welche Items in diesem Sprint angegangen werden und was geliefert werden kann. Im zweiten Teil wird geschaut, wie die Arbeit verteilt wird, und festgelegt, was das Team am Ende des Sprints liefern kann.

Zu Beginn jedes Tages hält das Team eine kurze Arbeitsbesprechung im Stehen ab. Bei diesem *Daily Scrum* spricht das Team über die Planung für diesen Tag und über die Zusammenarbeit. Dies ist kein Bericht an wen auch immer. Das Meeting ist allerdings öffentlich, sodass jeder zuhören kann.

Am Ende des Sprints wird die Arbeit durch das Team dem Product Owner und den Stakeholdern gezeigt, um deren Feedback zu erhalten. Dieses Meeting wird oft Demo genannt, aber damit ist es nicht ausreichend beschrieben. Die Demo ist das Mittel, das Ziel ist Feedback. Das Meeting heißt demnach auch *Sprint Review*.

Schließlich zieht sich das Team zu seinem letzten Meeting zurück: der *Sprint-Retrospektive*. Hier blickt das Team auf den vergangenen Zeitraum zurück und überlegt, wie es sich verbessern kann. Es benennt die Dinge, die gut laufen, und bespricht, was besser werden könnte. Das Team untersucht die Ursachen und Lösungen und trifft Vereinbarungen über durchzuführende Aktionen.

So, das war's! Fragen?