

1 Einleitung

1.1 Was ist Scrum?

1.1.1 Agiles Managementframework

Scrum [skrʌm] ist ein agiles Managementframework zur Entwicklung von Software, das aus wenigen klaren Regeln besteht. Diese beinhalten die Anwendung der drei Rollen Product Owner, Team und ScrumMaster, die Verwendung eines priorisierten Product Backlog sowie das Erstellen von Produktinkrementen innerhalb kurzer Arbeitszyklen, die Sprints genannt werden. Scrum lässt sich auf alle Arten der Softwareentwicklung anwenden: Software als eigenständiges Produkt und Software als Bestandteil eines Produkts, Software als unternehmensinterne Lösung oder Software, die im Auftrag eines Kunden entwickelt wird; Software, die neu entwickelt, und Software, die gewartet wird.

Als agiles Framework verkörpert Scrum die Werte des Agilen Manifests [Beck et al. 2001]. Dieses stellt den Menschen in den Mittelpunkt der Softwareentwicklung (*individuals and interactions, collaboration*). Schließlich entsteht Software nur durch die Interaktion und Kollaboration von Menschen. Scrum ist nicht technologie- oder toolorientiert, sondern fordert und fördert die enge Zusammenarbeit der Beteiligten. Das Agile Manifest formuliert außerdem die Optimierung von Kundenzufriedenheit und Wertschöpfung als Ziel der Softwareentwicklung (*working software, collaboration, responding to change*). Für kommerzielle Softwareprojekte zählt letztendlich, ob die wirtschaftlichen Ziele des Projekts erreicht wurden, ohne dabei Raubbau an den Mitarbeitern oder zukünftigen Softwareversionen und damit Kundenzufriedenheit und Wertschöpfung zu treiben. In Scrum ist der Product Owner für die Erreichung der wirtschaftlichen Ziele des Projekts verantwortlich und steuert dieses durch das priorisierte Product Backlog und den Releaseplan.

Scrum und Rugby

Der Begriff Scrum stammt aus dem Rugby und wird auf Deutsch als »Gedränge« übersetzt. Vereinfacht lässt sich der Spielzug folgendermaßen beschreiben: Jeweils acht Spieler der beiden Mannschaften formen das Gedränge. Beide Spielergruppen stehen sich eng umschlungen und nach vorne gebeugt gegenüber. Die vordersten drei Spieler verkeilen Kopf und Schultern. Alle Spieler drücken nun nach vorne. Ein Spieler außerhalb des Gedränges, der sog. Gedrängehalb der ballführenden Mannschaft, wirft den Ball seitlich in das Gedränge. Seine Mitspieler im Gedränge müssen den Ball mit den Füßen nach hinten schieben. Erst wenn der Ball das Gedränge verlassen hat, darf er wieder aufgenommen und ein Angriff eingeleitet werden. Das Gedränge ist ein komplizierter Spielzug, der sorgsam einstudiert und orchestriert werden muss. Er verlangt eine disziplinierte Teamarbeit.

1.1.2 Empirischer Prozess

Scrum ist ein empirischer Prozess. Arbeitsweise und Produkt werden regelmäßig begutachtet und angepasst (sog. *inspect and adapt*). Am Ende eines jeden Sprint beurteilt der Product Owner die Angemessenheit der erzielten Ergebnisse, und das Team reflektiert über seine Zusammenarbeit und die Anwendung des Prozesses. So lernt das Projekt von Sprint zu Sprint dazu und kann sich kontinuierlich verbessern. Scrum ist keine herkömmliche Methodologie und keine Komplettlösung. Scrum schreibt nicht detailliert vor, was wann zu tun ist, sondern fördert die Kreativität der Mitarbeiter. Daher beinhaltet Scrum auch keine Verfahrensanweisungen oder Templates: Soweit diese hilfreich sind, müssen Sie sie für Ihr Projekt und Ihre Organisation selbst erarbeiten.

1.1.3 Kein Wundermittel, sondern harte Arbeit

Scrum ist kein Wundermittel, das, einmal in eine Organisation eingeführt, quasi von selbst alles besser werden lässt. Im Gegenteil: Oft sind die ersten Sprints für Projektmitarbeiter und Management schwierig. Hindernisse und Probleme treten auf und müssen beseitigt werden, um zielgerichtet weiterarbeiten zu können. Dabei müssen alle Beteiligten nicht nur die neuen Spielregeln erlernen, sondern auch alte Angewohnheiten ablegen. Diese beinhalten das Zuweisen von Aufgaben an Mitarbeiter und das Erstellen qualitativ minderwertiger Software. Das erfolgreiche Anwenden von Scrum ist also ein Lernprozess, der Zeit und Geduld benötigt und neben den Teammitgliedern, dem ScrumMaster und dem Product Owner auch das Management betrifft. Seien Sie dabei auf der Hut: Oft ist es verlockend, nicht die eigenen Arbeitspraktiken, sondern Scrum zu ändern.

1.1.4 Scrum und schlankes Management

Die Geburtsstunde von Scrum fällt in das Jahr 1993: In diesem Jahr wurde das erste Scrum-Projekt durchgeführt [Sutherland 2004]. Beeinflusst wurde Scrum von Anfang an von neuen, innovativen Wegen in der Produktentwicklung, wie sie insbesondere von japanischen Unternehmen pilotiert wurden [Takeuchi&Nonaka 1986]. Diese neue Form des Managements und der Produktentwicklung wird heute als »schlank« (*lean*) bezeichnet [Womack&Jones 1996]. Besonders Toyota hat bei der Entwicklung schlanker Entwicklungsprozesse eine führende Rolle eingenommen [Liker 2003]. Ich verweise in diesem Buch auf erprobte Vorgehensweisen der schlanken Produktentwicklung [Morgan&Liker 2006, Ward 2007] und der schlanken Softwareentwicklung [Poppendieck 2003], wo diese Scrum-Praktiken erklären oder sinnvoll ergänzen. Scrum ist übrigens ein schlanker Prozess, der ein Ziehsystem (*pull*) einsetzt und Überlastungen systematisch vermeidet.

1.2 Warum Scrum?

1.2.1 Probleme frühzeitig erkennen, Handlungsspielraum sichern

Softwareentwicklung ist schwierig und herausfordernd. An dieser Tatsache ändert auch Scrum nichts. Denn das Wesen der Softwareentwicklung ist Innovation und Kreativität: Jedes Softwareentwicklungsprojekt befriedigt neue Kundenbedürfnisse. Um Bedürfnisse aufzudecken, zu verstehen und die richtige Lösung zu entwickeln, benötigen wir eine ordentliche Portion Kreativität. Dies fällt vielen Organisationen nicht leicht: Oft scheitern Softwareentwicklungsprojekte oder liefern Ergebnisse, die weder Kunden zufriedenstellen noch die angestrebten wirtschaftlichen Ziele erreichen. Organisationen und Projekte verfallen dabei in einen Teufelskreislauf:

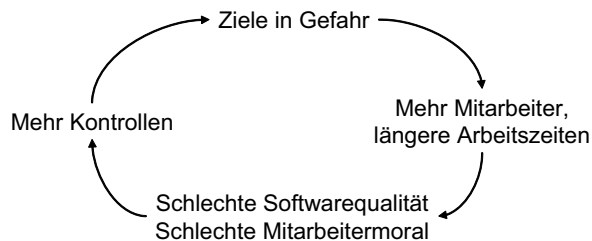


Abb. 1-1 Ein Teufelskreislauf, Quelle: [Ward 2007]

Sind die Ziele in Gefahr, so bitten wir häufig die Projektmitarbeiter länger zu arbeiten und fügen neue Mitarbeiter zum Projekt hinzu. Bedingt durch Hektik und Stress, längere Arbeitszeiten und schlecht eingearbeitete Projektmitglieder sinken

die Softwarequalität und die Moral der Mitarbeiter. Dies veranlasst das Management, mehr Kontrollen einzuführen, die die Entwicklung weiter verlangsamen.

Softwareentwicklungsprojekte weisen eine suboptimale Arbeitsorganisation nicht etwa deswegen auf, weil Management und Mitarbeiter nicht guten Willens sind. Das zentrale Problem traditioneller Vorgehensweisen besteht darin, dass wir frühzeitig versuchen, alle Eventualitäten und Arbeitsdetails zu antizipieren und einzuplanen, um anschließend unseren Plan auszuführen. Gleichzeitig erhalten wir erst spät im Projekt Rückmeldung über den tatsächlichen Fortschritt, meist erst dann, wenn die Software integriert und getestet wird. In Scrum führen wir alle Softwareentwicklungsaktivitäten innerhalb eines Sprint aus. So bekommen wir bereits nach wenigen Wochen Rückmeldung über den Fortschritt und etwaige Probleme und Hindernisse. Die Projektplanung fußt auf dem tatsächlichen Fortschritt des Projekts. Dabei gehen wir vor, wie in Abbildung 1–2 dargestellt.

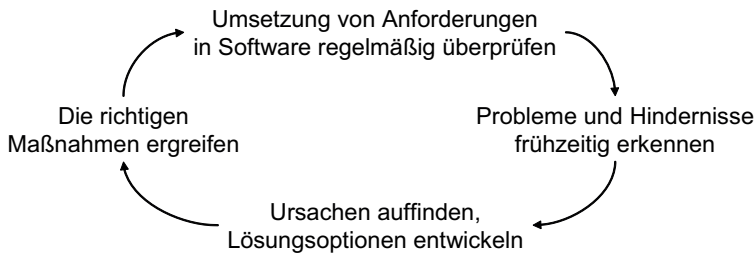


Abb. 1–2 Der Scrum-Kreislauf

Durch die Verwendung von kurzen Arbeitszyklen, an deren Ende ein Mehrwert entstanden sein muss, werden Probleme in Scrum frühzeitig offensichtlich. Wir haben so die Möglichkeit, rechtzeitig die Ursache des Übels aufzufinden, Lösungsoptionen zu entwickeln und die richtigen Maßnahmen zu ergreifen. Das frühe Auffinden von Problemen eröffnet uns einen größeren Handlungsspielraum und Flexibilität. Finden wir Probleme erst spät im Projekt, so sind viele Entscheidungen bereits gefällt und wir meist zur Schadensbegrenzung verdammt. Ursachenanalyse zu betreiben und die richtigen Verbesserungsmaßnahmen zu ergreifen, ist dabei kein Fingerschlecken, sondern harte Arbeit.

Die Mitarbeiter, Kunden und das liebe Geld

Wenn Sie Scrum konsequent einsetzen, so sollten Sie in den nachfolgend aufgeführten Bereichen positive Veränderungen erfahren.

Mitarbeiterzufriedenheit

Die Mitarbeiterzufriedenheit steigt bedingt durch Maßnahmen wie Bevollmächtigung und Selbstorganisation. Der Großteil der Scrum-Projektmitarbeiter bei Yahoo! beispielsweise beantwortete in einer Umfrage die Frage nach Verbesserung der Teammoral positiv [Deemer&Benefield 2006].

Kundenzufriedenheit

Durch die enge Zusammenarbeit mit Kunden und deren Einbeziehung beispielsweise in Anforderungsworkshops und Sprint-Reviews stellen wir sicher, dass die resultierende Software die Kundenbedürfnisse befriedigt.

Das liebe Geld

Softwareentwicklungsprojekte existieren, um einen wirtschaftlichen Nutzen zu erzielen. Scrum hilft, diesen durch folgende positive Auswirkungen zu erreichen oder zu übertreffen:

- *Time to market*
Scrum ermöglicht durch eine strikte Priorisierung der Anforderungen, durch die Vermeidung von Fehlleistungen und Überlastung, Software frühzeitig auszuliefern.
- *Qualität*
Scrum richtig angewandt führt zu einer Verbesserung der Softwarequalität. Ohne adäquate Qualität lassen sich auslieferbare Produktinkremente nicht innerhalb weniger Wochen erstellen.
- *Produktivität*
Durch enge Kollaboration, Selbstorganisation, durch Bevollmächtigung, durch das Vermeiden von Fehlleistungen und das Fokussieren auf die wichtigsten Anforderungen steigert Scrum die Produktivität. Yahoo! stellte beispielsweise eine deutliche Verbesserung der Produktivität beim Einsatz von Scrum fest [Deemer&Benefield 2006].

1.3 Warum dieses Buch?

Das vorliegende Buch bietet eine systematische und umfassende Beschreibung von Scrum und ergänzt die existierenden Bücher. Meine Intention ist nicht, dem Leser eine detaillierte Anleitung zur Anwendung von Scrum oder gar eine Scrum-Bibel zur Verfügung zu stellen. Dies widerspräche dem Wesen von Scrum als einem empirischen Prozess, der auf die spezifischen Bedürfnisse eines Projekts angewandt

werden muss und sich weiterentwickelt. Folgen Sie den Empfehlungen dieses Buchs daher nicht blind, sondern reflektieren Sie über Ihre Situation und passen Sie ggf. die Empfehlungen entsprechend an! Das Buch behandelt einführende und fortgeschrittene Themen. Stellen Sie sicher, dass Sie die Grundlagen verstanden haben, bevor Sie weiterführende Kapitel wie »Große und verteilte Projekte« lesen.

1.4 Mehr Informationen zu Scrum

Weitere Informationen zu Scrum sowie Scrum-Schulungen wie Certified Scrum Master (CSM) und Certified Scrum Product Owner (CSPO) finden Sie auf der Webseite der Scrum Alliance, www.scrumalliance.org.

1.5 Danke

Ohne die Unterstützung und Mitarbeit vieler Menschen wäre dieses Buch nie möglich gewesen. Besonders bedanken möchte ich mich bei Ken Schwaber, Mike Cohn, Jutta Eckstein, Preston Smith und Nancy van Schooenderwoert, ohne deren Unterstützung und Ermutigung ich dieses Buch wahrscheinlich nie geschrieben hätte. Jutta Eckstein ist quasi die Patentante des Buchs, da sie die Zusammenarbeit zwischen dem dpunkt.verlag und mir initiiert hat. Außerdem möchte ich mich ganz herzlich bei meinen Reviewern bedanken: Jutta Eckstein, Stefan Rook, Sabine Canditt, Jens Coldewey, Bernd Oestereich, Nicolas Arnold und Jiri Lundak.