



Ein Plädoyer für agile Softwareentwicklung mit Scrum

Geregelte Selbstbestimmung

Marion Eickmann

Auch wenn Scrum mittlerweile auf eine fast 15-jährige Erfolgsgeschichte zurückblicken kann, betrachten viele Entwickler und Manager diese agile Projektmanagement-Methode immer noch mit großer Skepsis. Eine Protagonistin dieses Verfahrens argumentiert vehement dafür, Scrum auf jeden Fall einmal auszuprobieren.

Scrum ist ein iterativer, inkrementeller Prozess für die Produktentwicklung und die Organisation von Teams, den Mitte der 90er-Jahre Ken Schwaber und Jeff Sutherland entwarfen. Dreh- und Angelpunkt ist das Setzen auf eine hohe Eigenmotivation: Mit Scrum legt ein Team selbst fest, welche Aufgaben es wie erledigen kann. Kundenanforderungen werden iterativ priorisiert und sukzessive realisiert. Eine prägnante Definition ist beispielsweise diese: „Scrum is a simple ‚inspect and adapt‘ framework that has three roles, three ceremonies, and three artifacts designed to deliver working software in Sprints, usually 30-day iterations.“ [1] Sport-

interessierten dürfte das Wort Scrum übrigens aus einem ganz anderen Zusammenhang bekannt sein: Es ist im Rugby die Bezeichnung für das „geordnete Gedränge“ zum Wiederbeginn des Spiels nach einem Regelverstoß.

Die drei Scrum-Rollen

Im Gegensatz zu klassischen Projektmanagement-Methoden kennt Scrum weder Produktmanager noch Projekt- oder Teamleiter – jedenfalls nicht dem Namen nach. Stattdessen arbeitet es mit drei wichtigen Rollen: Product Owner, Scrum Master und Team. (Zur Problematik der eng-

lischen Begrifflichkeiten siehe Textkasten „Scrum auf Deutsch“ auf Seite 11.)

Alle drei sind gleichgestellte Managementrollen und haben bestimmte Verantwortlichkeiten, im Wesentlichen die im Folgenden genannten.

Der Product Owner ist verantwortlich für die „Produkt-

vision“, die umfassende Vorstellung vom Endergebnis, die Erhebung und Priorisierung der Anforderungen, die Budgetkontrolle und den Return on Investment (Kosten-Nutzen-Verhältnis). Der Scrum Master räumt Probleme aus dem Weg, sorgt dafür, dass alle die Scrum-Regeln einhalten, und schafft die nötigen Rahmenbedingungen für das Team. Das Team ist in Scrum eine selbstorganisierte Einheit, die für die Erstellung und die Qualität des Produkts verantwortlich ist. Neben diesen drei Rollen gibt es noch weitere Parteien („Stakeholder“), die Anforderungen stellen oder als Beobachter und Ratgeber dienen.

Scrum-Projekte werden in Iterationen, sogenannte Sprints, aufgeteilt, und haben in der Regel eine Länge (Time Box) von zwei oder vier Wochen.

Zu Beginn eines Sprint stellt der Product Owner dem Team im Rahmen des Sprint Planning Meeting das priorisierte Product Backlog, eine Liste aller Anforderungen („User Stories“), vor. Das Team findet durch gezielte Fragen heraus, was genau sich hinter den fachlichen Anforderungen verbirgt und schätzt den groben Aufwand. Anders als in klassischen Projektmanagement-Verfahren schreibt man dem Scrum Team nicht vor, wie viele Features es in einer bestimmten Zeit realisieren soll. Stattdessen wählt das Team selbst aus, die Umsetzung welcher Funktionen es in einem Sprint für realistisch hält und „committed“ sich dazu, diese zu realisieren.



- Scrum ist eine agile Projektmanagement-Methode zur Softwareentwicklung.
- Scrum setzt stark auf die Selbstorganisation und Selbstbestimmung von Teams.
- Scrum versucht Transparenz zu erreichen durch frühzeitige und umfassende Einbeziehung des Kunden und regelmäßige Kommunikation aller Beteiligten.

Im zweiten Teil des Sprint Planning Meeting, an dem nur noch der Scrum Master und das Team teilnehmen, brechen diese die User Stories in einzelne Aufgaben herunter und legen sie im Sprint Backlog schriftlich fest. Diese Aufzeichnungen werden täglich mit der Angabe über den jeweiligen Restaufwand versehen. So entsteht der immer aktuelle Burn-down Chart, eine grafische Darstellung des Restaufwands.

Fertig ist wirklich fertig

Eine Erweiterung des Sprint Backlog während der laufenden Iteration ist nicht zulässig, um die Fertigstellung der vereinbarten User Stories nicht zu gefährden. Denn nur was am Ende fertig ist, und zwar fertig im Sinne von „getestet und dokumentiert“, kann der Product Owner im sogenannten Review Meeting abnehmen. Aussagen wie man sei fast fertig gibt es somit nicht mehr.

Während des Sprints arbeitet das Team konzentriert und ungestört die Tasks aus dem Sprint Backlog ab. Es trifft sich täglich zu einem „timeboxed“ Meeting von 15 Minuten, dem Daily Stand-up Meeting, und bespricht in der Regel folgende Dinge:

- Was habe ich gestern geschafft?
- Was werde ich heute tun?
- Gibt es Probleme?

Am Ende des Sprint präsentiert das Team dem Product Owner im Sprint Review Meeting die implementierten Funktionen. Der Product Owner entscheidet dann, ob er die Lieferung abnimmt und ob alle Akzeptanzkriterien erfüllt sind.

Der Scrum Master sorgt während des gesamten Prozesses für die Einhaltung der Regeln und wehrt Störungen von außen ab. Das Retrospektiv Meeting am Ende eines Sprint dient der Reflektion über gut und schlecht Gelaufenes sowie darüber, wie der Prozess angepasst und was im

nächsten Sprint geändert werden sollte.

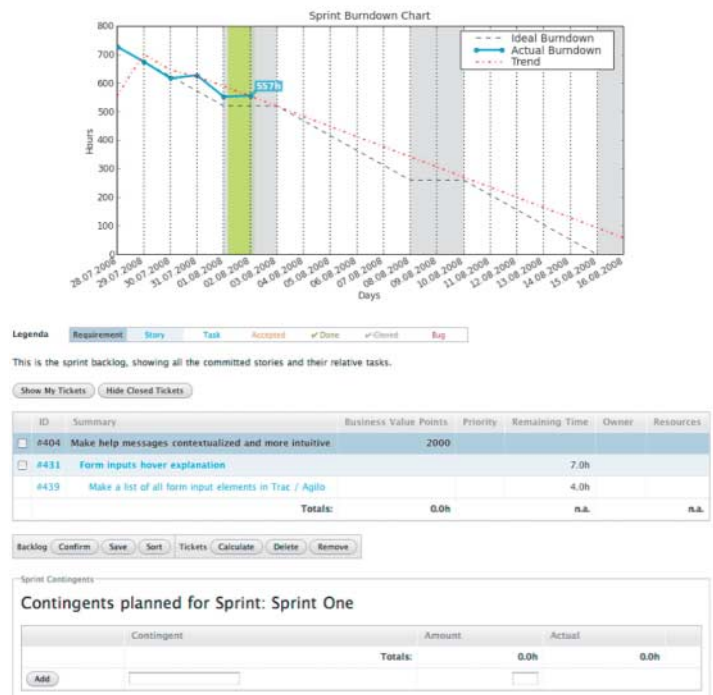
Eine Einführung von Scrum geht, wie der Einsatz jeder Vorgehensmethode, immer einher mit Veränderungen. Diese betreffen nicht nur den Softwareentwicklungsprozess selbst, sondern sie beeinflussen die Mitarbeiter, das Management, Beurteilungssysteme und Stellenbeschreibungen, sprich die gesamte Firmenstruktur und -kultur.

In seinem Buch „The Enterprise and Scrum“ stellt Ken Schwaber einige provokante und auf den ersten Blick negativ klingende Thesen auf. Diese machen deutlich, warum Scrum so oft auf Bedenken stößt. Schwaber sagt unter anderem zwei Dinge voraus [2]:

Scrum und seine Folgen

„Scrum wird Mitarbeiter veranlassen, das Unternehmen zu verlassen.“ Scrum schafft Transparenz und sorgt dafür, dass man sich nicht mehr hinter den (vermeintlichen) Fehlern der anderen verstecken kann. Durch die Einhaltung der „Time Box“, die klaren Aufgaben innerhalb des Teams und die Transparenz, wann was zu liefern ist, ziehen Argumente wie „Ich habe ja noch auf dieses und jenes gewartet“ oder „Ich musste so viele andere Sachen für den Kunden erledigen und bin einfach nicht dazu gekommen“ nicht mehr. So kann Scrum in der Tat dafür sorgen, dass sich einige Mitarbeiter oder Manager nicht mehr wohl fühlen und das Unternehmen verlassen. Transparenz macht Schwächen sichtbar, und Scrum hat ausdrücklich den Anspruch, Transparenz zu vertiefen.

„Scrum führt zu Konflikten.“ Die einfachen Regeln von Scrum sorgen dafür, dass jeder Grund, der für Schwierigkeiten bei der Software- und/oder Produktentwicklung im jeweiligen Unternehmen sorgt, erkannt wird. In vielen Unternehmen führt dies zu



Ein Sprint Burn-down Chart in Agilo for Scrum (Abb. 1).

dem Versuch, Scrum so zu verdrehen, dass die Fehler verdeckt werden, statt das Problem zu beseitigen. Genau das ist aber nach Meinung der Scrum-Erfinder die Ursache für Aussagen wie „Scrum ist chaotisch“ oder „Scrum funktioniert bei uns nicht“.

Das Spektrum möglicher Konflikte, die durch die Einführung von Scrum auftauchen können, ist vielfältig. Es reicht von Managern, die zwar Transparenz fordern, aber auf keinen Fall für ihre eigene Arbeit, über klassische Projektleiter, die erst verstehen müssen, welche Rolle sie bei Scrum einnehmen, bis hin zu Kunden, von denen Mitarbeit gefordert

ist. Nicht umsonst gibt es eigens aufgelegte Zertifizierungsprogramme für Scrum Master und Product Owner.

Scrum verändert Strukturen

Bei Vorträgen vor Projektleitern und Entwicklern ist meist eine der brennendsten Fragen, wie Scrum zu den bestehenden Strukturen und Rollen passt.

Wie erwähnt definiert Scrum drei unterschiedliche Rollen – Product Owner, Scrum Master und Team. Die entsprechenden Äquivalente lassen sich auch in klassischen Projekten finden – die

Scrum auf Deutsch

In seinem Blog auf heise Developer hat sich Bernd Oesterreich mit der Problematik der englischen Begrifflichkeiten dieses Verfahrens beschäftigt und ein paar Eindeutschungen vorgeschlagen:

Sprint: Iteration

Sprint Backlog: Iterationsplan oder Aufgabenliste

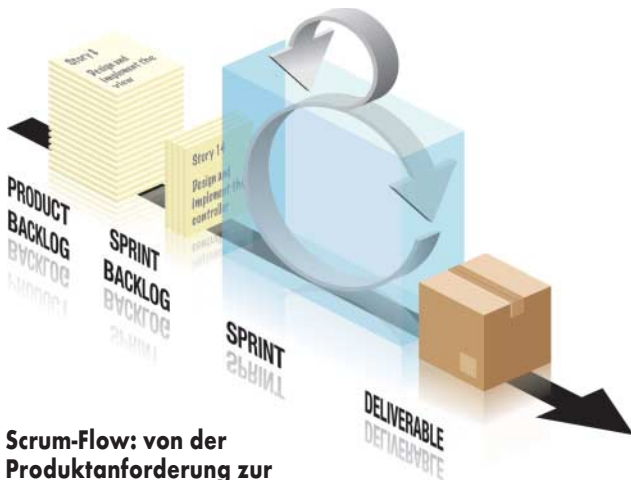
Product Backlog: Featureliste

Burn-Down Chart: Abarbeitungsdiagramm

Daily Scrum-Meeting: tägliches Team-Treffen

Product Owner: Produktverantwortlicher oder Produktmanager

Mehr dazu ist unter <http://www.heise.de/developer/Ge-danken-zur-Sprache-in-Scrum-/blog/artikel/137770> nachzulesen.



Scrum-Flow: von der Produktanforderung zur lieferbaren Anwendung (Abb. 2)

Bedeutung und Stellung der Scrum-Rollen ist jedoch eine völlig andere:

- Product Owner statt Produktmanager,
- Scrum Master statt Team- oder Projektleiter,
- Das Team übernimmt selbst eine Managementrolle.

Die Namensgebung macht es deutlich – bei Scrum ist der Produkt Owner nicht nur der Manager des Produktes, sondern der „Besitzer“ und somit für es gesamtverantwortlich. Product Owner zu sein geht mit einer Reihe von Anforderungen einher, die sich zum Teil mit denen an den „klassischen“ Projektmanager decken:

- verantwortlich zu sein für den Erfolg dessen, was das Team liefert,
- Entscheidungen über Wichtigkeit und Prioritäten zu treffen,
- dafür zu sorgen, dass das Team das Gesamtbild des Produktes nicht aus den Augen verliert,
- die User Stories für das Entwicklerteam vorzubereiten,
- über ein hohes Wissen über das fachliche Umfeld zu verfügen,
- die Ergebnisse zu validieren und auf ihre Qualität zu überprüfen,
- zeitnah auf Rückfragen zu reagieren,
- kontinuierlich mit allen Stakeholdern, Geldgebern und dem Team zu kommunizieren,
- die finanziellen Aspekte des Projekts zu kontrollieren.

Anders als bei anderen Methoden ist das Team bei Scrum nicht nur das ausführende Organ, das Tasks vom Projektleiter bekommt, sondern es entscheidet eigenverantwortlich, welche Anforderungen beziehungsweise User Stories es in einem Sprint erledigen kann. Es erstellt seine Tasks selbst und ist verantwortlich für die Umsetzung – das Team wird zum Manager. Natürlich gelingt dies nur, wenn bestimmte Rahmenbedingungen gegeben sind wie ein gut vorbereitetes Product Backlog.

Der Master als Diener

Dieses neue Selbstverständnis des Teams und die damit verbundenen Aufgaben und Verantwortlichkeiten verändern auch die Rolle des Teamleiters/Projektleiters. Der Scrum Master hat nicht mehr die Aufgabe, Arbeiten zu delegieren und das Projekt zu planen, sondern er sorgt dafür, dass das Team alle Voraussetzungen hat, die selbstgesteckten Ziele zu erreichen. Er räumt Hindernisse aus dem Weg, verschafft dem Team ein optimales Arbeitsumfeld, coacht und ist verantwortlich für die Einhaltung der Scrum-Regeln – er wird zum sogenannten Servant Leader.

Dieses veränderte Rollenverständnis ist einer der wichtigsten Aspekte, wenn man

Scrum verstehen und im Unternehmen einführen will.

Scrum funktioniert nur, wenn man den Mut hat, die Scrum-Regeln und die Ideen, die hinter Scrum stehen, wirklich umzusetzen. Denn Scrum ist viel mehr als nur die Summe aller Rollen, Zeremonien und Artefakte.

11 schlimme Scrum-Anti-Pattern

Bei der Einführung von Scrum geht nicht immer alles von Anfang an glatt. Alle Beteiligten müssen sich an das neue Vorgehen gewöhnen und sind unsicher, was zu tun ist, wenn Unerwartetes geschieht. Solche Schwierigkeiten führen häufig dazu, „bekannte“ Mechanismen „nur für diesen einen Fall“ anzuwenden, anstatt das Problem nach Scrum-Art zu lösen. Dadurch geht leider der Scrum-Effekt schnell verloren.

Hier die schlimmsten Scrum-Anti-Pattern auf einen Blick:

- Der Scrum Master weist Tasks zu und zerstört so das sich selbststeuernde und -organisierte Team.
- Der Sprint wird von außen gestört. Es werden neue Aufgaben eingefügt oder Änderungen an den gewählten (committed) User Stories vorgenommen.
- Es finden keine Review Meetings statt.
- Es findet keine Retrospective statt.
- Es gibt keine regelmäßigen Daily Standup Meetings.
- Der Product Owner hat keine ausreichende Kompetenz und nimmt seine Rolle nicht wahr.
- Es gibt kein „1 to 1 Agreement“, also keine persönliche Vereinbarung, zwischen dem Product Owner und dem Team.
- Die Beteiligten sind nicht ausreichend geschult.
- Meetings sind nicht Time-Boxed.
- Anstatt Features werden Aktivitäten erfasst.

– Man schickt ein paar Scrum Master zum Training und denkt, nun werde alles gut.

Zum Schluss sei noch auf ein Standardargument gegen Scrum eingegangen: Das Verfahren sei ungeeignet zur Systementwicklung. Doch entgegen gelegentlich verbreiteter Behauptungen eignet sich Scrum durchaus für die Entwicklung komplexer Systeme und umfangreicher, langwieriger Projekte. Denn gerade hier spielt eine genaue Planung eine große Rolle, und die kontinuierliche Integration neuer Funktionen stellt sicher, dass es am Ende kein böses Erwachen gibt. Viele Produktionsbetriebe arbeiten nach „lean“-Prinzipien, und das Softwareentwicklungs-Pendant Scrum schafft Synergie.

Und außerdem ...

Und last but not least – Scrum macht Spaß: Nicht das Zuweisen von Aufgaben, sondern das kooperative Arbeiten und Entscheiden im Team sind wichtiger Bestandteil von Scrum. Softwareentwicklung wird als das gesehen, was sie ist: Eine kreative und vielseitige Tätigkeit, die nur dann richtig funktioniert, wenn alle Verantwortung übernehmen. (JS)

MARION EICKMANN

gehört zu den Gründern der agile42 GmbH, die auf das Consulting, Training und Coaching von agilen Vorgehensweisen wie Scrum spezialisiert ist.

Literatur

- [1] Andrew Scotland; Scrum @ the BBC. in: JAOO 2005. Aarhus, Dänemark
- [2] Ken Schwaber; The Enterprise and Scrum, Microsoft Press 2007

Teil 1: Die Rolle des Product Owner



Balance halten

Marion Eickmann

Wer agile Softwareentwicklung betreiben will, trifft früher oder später unweigerlich auf Scrum. Erfolg verspricht dieses Vorgehensmodell allerdings nur dann, wenn man die definierten Rollen mit den richtigen Personen besetzt.

Scrum ist ein iterativer und inkrementeller Prozess für die Produktentwicklung und die Organisation von Teams. Dieses Vorgehensmodell ersetzt die klassischen Rollen und Strukturen durch agile und transparente Abläufe. Eine der „neuen“ Rollen ist die des Produktverantwortlichen – im Scrum-Jargon und im Folgenden der Product Owner. Um seine Aufgaben, Verantwortlichkeiten und persönlichen Fähigkeiten geht es in diesem ersten Teil. Ein zweiter Artikel wird sich im nächsten Heft mit der Rolle des Scrum Master befassen.

Wie alle agilen Vorgehensweisen basiert Scrum auf wenigen einfachen Regeln und ist daher leicht zu verstehen. Gleichzeitig erfordert es große Disziplin.

Ein Kampf auf zwei Seiten

Dazu zählt die hohe Eigenmotivation des Projektteams. Damit es schnell und effektiv arbeiten kann, muss es sich selbst organisieren und festlegen, welche Aufgaben es während eines vorgegebenen kurzen Zeitraums, dem sogenannten Sprint, erledigen kann. Diese Selbstverantwortung trägt maßgeblich zur Eigenmotivation des Projektteams bei. Entscheidend für den Erfolg von Scrum ist die Rolle des Product Owner, da er als Schnittstelle zwischen den Teams und anderen involvierten Parteien (Stakeholdern) agiert.

In Firmen, die nach Scrum arbeiten, sind die Aufgaben und Verantwortungsbereiche der jeweiligen Product Owner nie identisch. Angefangen bei der Auswahl der Person für diese Rolle mit den richtigen und notwendigen Skills, über gezielte Trainings- und Schulungsmaßnahmen bis hin zum Verantwortungsbereich ist die Rolle des Product Owner – kurz PO – die komplexeste bei diesem Vorgehen.

Häufig hat der PO auf beiden Seiten zu „kämpfen“.

Während die Teams in festgelegten Zeitabschnitten (time-boxed) und vom Scrum Master „beschützt“ arbeiten können, muss sich der Product Owner nicht selten mit dem Vertrieb, dem Marketing, der Geschäftsführung oder dem Kunden auseinandersetzen, um auf der anderen Seite den Teams präzise Softwareanforderungen (User Stories) präsentieren zu können (siehe Kasten „Kriterien für User Stories“).

Der Product Owner ist verantwortlich für die Rendite, den Return on Investment (ROI). Er validiert die Ergebnisse und prüft, ob die Qualität aus Sicht der Endnutzer akzeptabel ist. Außerdem entscheidet er, wie wichtig einzelne Funktionen und mit welcher Priorität sie zu behandeln sind. Dem Team muss der Product Owner zudem eine Vorstellung davon vermitteln, wie das fertige Produkt aussehen soll (Produktvision). Damit das Team effektiv arbeiten kann, muss er auf Rückfragen schnell reagieren.

Somit erfüllt er in einem Projekt die Rolle des Kommunikators und muss kontinuierlich mit allen Stakeholdern, Sponsoren und dem Team im Gespräch sein. Last but not least ist es seine Aufgabe, die finanzielle Seite der Produktentwicklung zu koordinieren, was ihm durch die kontinuierliche Arbeit sowie das Festlegen einer Prioritätenliste der anfallenden Aufgaben (Product Backlog) gelingt.

Diese diversen Anforderungen zeigen, dass die Auswahl der „richtigen“ Person für die Rolle des PO von großer Bedeutung für den Erfolg eines Projekts ist.

Vom Kommunikationstalent ...

Der Product Owner ist nicht immer der einzige, der Anforderungen definiert und stellt. Oft sind viele Stakeholder, etwa das Marketing oder für die Qualitätssicherung (QA) verantwortliche

Abteilungen involviert, die auf unterschiedlichen Ebenen Anforderungen formulieren. So braucht der Vertrieb möglichst schnell eine bessere Performance, die QA-Abteilung muss sicherstellen, dass bestimmte Sicherheitsstandards erfüllt sind, und das Marketing möchte das neue Corporate Design schnellstmöglich eingebaut sehen.

Aufgrund dieser Vernetzung zieht die Realisierung der kleinsten User Story oft weite Kreise durch das gesamte Unternehmen. Und so unterschiedlich die Interessen der verschiedenen Stakeholder sein mögen, eines haben alle gemein: Ihre eigenen Anforderungen sind immer die wichtigsten.

Die Aufgabe des PO ist nun, zum einen die Anforderungen zu gewichten, das Geschäftsinteresse der Firma (Business Value) zu wahren und zu koordinieren, wann ein Team welche Anforderungen realisiert. Zum anderen muss er über starke kommunikative, argumentative und diplomatische Fähigkeiten verfügen, um für ein wechselseitiges Verständnis zwischen den verantwortlichen Stakeholdern zu sorgen und so einen allgemeinen Konsens zu erreichen.

Das Vertrauen der beteiligten Parteien, deren Mitarbeit und der bidirektionale Informationsfluss stellen sicher, dass Ergebnisse erzielt, aber auch Probleme schnell er-

ID	Summary	Business Value Points	Roif	Importance	User Story Points
#8	Req 4	3000	n.a.		n.a.
#2	Req 2	2000	n.a.		n.a.
#19	5th User Story for Req 2			Mandatory	
#39	Req 7	1200	n.a.		n.a.
#40	Story for Requirement 7			Linear	
#37	req 6	1200	n.a.		n.a.
#38	This Story belongs to req 6				
#1	Req 1	500	38.46		13.0
#3	3rd User story for Req 1			Linear	
#12	2nd User Story for Req 1			Exciter	
#10	Req 5		n.a.		n.a.
#5	Req 3		n.a.		n.a.
#6	Req3 user story			Mandatory	

Beispiel eines Product Backlog der Open-Source-Scrum-Software Agilo. Der Product Backlog ist sortiert nach Business Value und enthält alle Anforderungen sowie User Stories, die für den nächsten Sprint relevant sind (Abb. 1).

kannt und offen angesprochen werden.

... zum Domain-spezialisten

Wer jetzt glaubt, dies sei schon alles, liegt weit daneben. Ein Product Backlog besteht ja nicht nur aus den Anforderungen (Pflichten), sondern auch aus den sich daraus ergebenden User Stories (Lasten), die das Team umsetzen soll. Der Product Owner muss also neben der eben beschriebenen Fähigkeit, Anforderungen zu erheben (Requirements Elicitation) und zu priorisieren, in der Lage sein, die Anforderungen in konkrete Aufgaben

herunterzuberechnen. Dies setzt umfassendes Wissen darüber voraus, was das Produkt am Ende können muss, beziehungsweise realistisch einzuschätzen, wofür der Nutzer der Software am Ende bereit ist zu zahlen.

Eine solche Rolle existiert in traditionellen Unternehmensstrukturen bisher nicht. Aus diesem Grund entbrennt bei der Einführung agiler Methoden wie Scrum spätestens, wenn es um die Besetzung und die Verantwortung des PO geht, eine kontroverse Diskussion. Die Frage, welche der klassischen Rollen im Unternehmen er übernehmen sollte oder könnte, ist nicht einfach zu beantworten. Je nach Unternehmensstruktur könnten dies der Anforderungsanalyst, der Business Analyst, der Projektleiter, der Produktmanager oder andere sein.

Zunächst ein Blick auf den klassischen Projektmanager. Bei einer Umstellung auf Scrum bleibt ihm eigentlich nur die Wahl zwischen der Rolle des Scrum Master (analog zum Team- oder Projektleiter) oder der des Product Owner. Die Entscheidung fällt dabei häufig auf die des Scrum Master, da er vordergründig mehr mit den Aufgaben eines Projektmanagers

gemein zu haben scheint und enger mit dem Entwicklungsteam zusammenarbeitet.

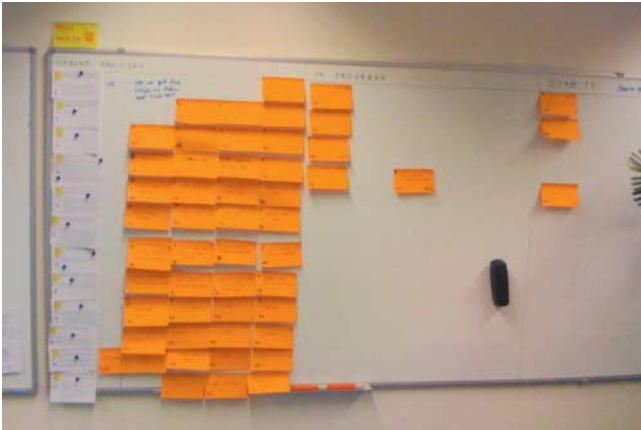
Diese Entscheidung ist richtig, wenn er in erster Linie auf die Einhaltung der Scrum-Regeln achten und dem Team Hilfestellung geben soll, der Fokus also auf der „Projektbegleitung“ liegt. Der Begriff Agile Leader beschreibt dahingehend die Rolle des Scrum Master am besten (siehe „Onlinequellen“). Gehören jedoch eher die Analyse, Steuerung, Budgetierung und Kundenkommunikation zu den geforderten Aufgaben, sollte der frühere Projektmanager eher die Rolle des Product Owner übernehmen.

Scrum-Zeremonien sind Fixpunkte

Scrum kennt drei Zeremonien, von denen zwei für den Product Owner von Bedeutung sind. Während des „Sprint Planning Meeting“ stellt der PO dem Team die wichtigsten Anforderungen vor und diskutiert sie mit seinen Mitarbeitern. Dafür stellt er im Vorfeld im Product Backlog die Anforderungen zusammen und versteht sie mit Prioritäten. Er muss Fragen des Teams beantworten und die Richtung – das „Big Picture“ – sowie die



- Ein Projektteam, das nach dem agilen Vorgehensmodell Scrum arbeitet, muss drei Rollen besetzen: die des Product Owner, des Scrum Master und des Teams.
- Die Rolle des Product Owner ist entscheidend für den Erfolg von Scrum, da er die Kommunikation und die Zusammenarbeit zwischen unterschiedlichen Parteien gewährleisten muss.
- Sollte in einem Unternehmen keine Person mit den geforderten unterschiedlichen Fähigkeiten zu finden sein, können mehrere Mitarbeiter ein Product Owner Team bilden; Ansprechpartner für die anderen Projektbeteiligten bleibt weiterhin eine Person.



So kann ein Product Backlog auf dem Whiteboard aussehen (Abb. 2).

Ziele veranschaulichen, damit das Team mit der Umsetzung der Anforderungen zielgerichtet beginnen kann.

Am Ende dieses Meetings verpflichtet sich das Team (Commitment), eine bestimmte Anzahl der priorisierten Anforderungen bis zum Ende des Sprint zu erfüllen, und legt gemeinsam mit dem Product Owner fest, welchen Kriterien die Software für die spätere Abnahme genügen muss. Das Team beginnt nun mit dem Sprint, und der Product Owner hat währenddessen Zeit, die nächsten Anforderungen zu „sammeln“ und für das kommende Sprint Planning Meeting nach Prioritäten zu ordnen.

An den täglichen Zusammenkünften des Teams – den Daily Stand Up Meetings oder Daily Scrums – können neben dem Product Owner alle anderen Stakeholder teilnehmen, jedoch ist es ihnen nicht erlaubt zu sprechen. Diese Meetings dienen ausschließlich

dem Team und dem Scrum Master dazu, über Erledigtes, neue Aufgaben oder aufgetretene Probleme zu sprechen.

Der Product Owner wird erst wieder am Ende des Sprint, im Review Meeting, tätig. Hier stellt ihm das Team die erledigten Aufgaben vor, und er überprüft anhand der Abnahmekriterien, ob die Vorgaben erfüllt oder Nacharbeiten notwendig sind.

Nicht jeder ist ein Übermensch

Kommunikationstalent, Domainspezialist, Anforderungsingenieur und Businessanalyst – all das sollte ein guter Product Owner sein. Aber was tun, wenn es im Unternehmen keine Person gibt, die alle diese Fähigkeiten vorzuweisen hat, oder wenn die Produktentwicklung so komplex ist, dass verschiedene Teams auf ein gemeinsames Ziel hin arbeiten?

Einer der ausschlaggebenden Gründe für den Erfolg von Scrum ist die direkte und klare Kommunikation zwischen den einzelnen Rollen. Das Team „verspricht“ einer Person, nämlich dem Product Owner, dass es bestimmte Aufgaben bis zum Ende des Sprint erfüllen wird. Lässt man mehr als eine Person als Product Owner agieren, verliert dieses Commitment seine Verbindlichkeit.

Eine Lösung für dieses Dilemma ist das Product Owner Team, das verschiedene Skills vereint und unterschiedliche Tätigkeiten kombiniert. Die Mitglieder dieses Teams identifizieren die Anforderungen, verwalten das Budget, schreiben User Stories et cetera. Der Product Owner selbst sammelt alle Informationen und bleibt dem Team gegenüber der verantwortliche Ansprechpartner. Auf diese Art und Weise können Know-how und Fähigkeiten vieler Personen zusammenspielen, ohne dabei den wichtigen „Face-to-Face“-Aspekt von Scrum zu vernachlässigen.

Da jedes Unternehmen unterschiedliche Strukturen hat, muss man Methoden wie Scrum jeweils anpassen. Dies bedeutet nicht, Scrum selbst zu verändern, sondern lediglich die wichtigen Regeln in den Kontext eines Unternehmens zu integrieren.

Fazit

Dass die Rolle des Product Owner komplex und häufig

schwierig ist, liegt zum einen an seinem umfangreichen Aufgabengebiet sowie daran, dass von ihm diplomatische Fähigkeiten gefordert werden. Er muss die unterschiedlichen Wünsche aller Stakeholder nach Wichtigkeit sortieren und sicherstellen, dass alle im Unternehmen diese Prioritätenliste akzeptieren.

Zum anderen, und dieser Grund scheint entscheidend, sehen viele Unternehmen Scrum lediglich als „Softwareentwicklungs-Thema“ und sind sich der Tragweite der Scrum-Einführung nicht bewusst. Ein Produkt herzustellen, sei es Software, Hardware oder die Kombination aus beiden, erfordert die Zusammenarbeit aller Beteiligten, von der Idee und dem Herausarbeiten des Funktionsspektrums über die Realisierung und Lieferung bis hin zu Änderungswünschen und neuen Anforderungen. Alle an diesem Prozess Beteiligten sollten auch an Scrum beteiligt sein.

Scrum ist eine effektive und transparente Methode zur Optimierung der Zusammenarbeit aller am Produktentwicklungsprozess Beteiligten, und der Product Owner fungiert sozusagen als Schnittstelle. Das macht seinen „Job“ interessant und schwierig zugleich. (ka)

MARION EICKMANN

gehört zu den Gründern der agile42 GmbH, die auf das Consulting, Training und Coaching von agilen Vorgehensweisen wie Scrum spezialisiert ist.

Literatur

- [1] Marion Eickmann; *Geregelte Selbstbestimmung; Ein Plädoyer für agile Softwareentwicklung mit Scrum; iX 6/09, S. 110*

Kriterien für User Stories

- I Independent (unabhängig)
- N Negotiable (verhandelbar)
- V Valuable (wertvoll)
- E Estimable (schätzbar)
- S Small (klein genug)
- T Testable (testbar)

Onlinequellen

- Tim Listers Vortrag zur Rolle des Agile Leader
www.infoq.com/presentations/agile-leadership-tim-lister
- Scrum Alliance Webseite
www.scrumalliance.org/user_groups
- Scrum User Group in Berlin
www.scrumtisch.net
- Scrum User Group in Karlsruhe
scrumusergroupkarlsruhe.pbworks.com
- Scrum User Group in München
www.agiletuesday.org/tiki-index.php

Teil 2: Die Rolle des Scrum Master



Facettenreich

Marion Eickmann

Als agile Methode des Projektmanagements macht Scrum immer mehr von sich reden. Wer das Vorgehensmodell erfolgreich einsetzen will, sollte sich vorab die Beschreibungen der drei zu besetzenden Rollen genau ansehen und nicht etwa den Scrum Master mit einem Teamleiter verwechseln.

Unternehmen, die auf Scrum „umgestellt“ haben, sind effizienter als zuvor, Teams, die nach Scrum arbeiten, produktiver und mit Scrum realisierte Projekte besser planbar. Diese Aussagen sind wahr, aber nur, wenn Scrum als ganzheitliche Methode richtig eingesetzt wird.

Eine wichtige Rolle bei diesem Prozess kommt dem Scrum Master zu. Dieser zweite Teil des Scrum-Tutorials beschäftigt sich mit dessen Verantwortlichkeiten, was er leisten kann, und wo es Grenzen gibt.

Scrum kennt drei Managementrollen, die gemeinsam das Scrum-Team bilden: den Product Owner, den Scrum Master und das Development Team. Die drei Rollen stehen gleichberechtigt nebeneinander und umfassen spezielle Aufgaben und Verantwortlichkeiten. Der Product Owner, dessen Rolle ein Artikel in der letzten iX beleuchtet hat [1], ist dafür zuständig, Anforderungen zu erheben, zu priorisieren und dem Team die „Vision“ und das Ziel zu verdeutlichen – also zu klären, wie das fertige Produkt aussehen soll. Das Development Team zeichnet verantwortlich für die Entwicklung des Produkts, für seine Qualität und die technische Umsetzung.

Die Aufgaben des Scrum Master und wie sich seine Rolle gegenüber der eines Teamleiters bei klassischen Vorgehensweisen unterscheidet, sind Kern dieses Artikels.

Die Frage nach dem „Agile Leader“ stellte Tim Lister auf der Agile Conference 2006 (siehe „Onlinequellen“, [a]). Betrachtet man Abbildung 1, mag man im ersten Moment glauben, der kleine freche Kerl ganz vorn führt die Gruppe an, denn er bereitet den Weg und stürmt voraus. Aber weit gefehlt. Der eigentliche „Leader“ ist das Küken, das sich ganz am Ende der Gruppe umschaute, das sicherstellt, dass keine Gefahr droht, und das alles im Blick und Griff hat.

Gretchenfrage: Wer führt wen?

Dieses Bild macht den Unterschied zwischen einem klassischen Projektleiter und einem Scrum Master deutlich und veranschaulicht das Rollenverständnis eines agilen Leaders, der sich selbst zurücknimmt und hinter das Team tritt.

Den augenscheinlichsten Unterschied zwischen einem Teamleiter und einem Scrum Master repräsentiert die Namensgebung. Während der eine das Team leitet und vorgibt, was inhaltlich zu tun ist, sorgt der andere dafür, dass das Team die Regeln von Scrum, also die Methode an sich, beachtet und umsetzt.

Der Scrum Master mischt sich nicht in die entwicklungs-spezifischen Entscheidungen des Teams ein, sondern steht höchstens beratend zur Seite. Aktiv greift er nur dann ein,

iX-TRACT

- Als Inhaber der zweiten wichtigen Rolle in einem Projekt-Team, das nach dem Vorgehensmodell Scrum arbeitet, vermittelt der Scrum Master zwischen dem Product Owner und dem Team.
- Er moderiert bei Konflikten und räumt, soweit möglich, Hindernisse aus dem Weg, die die Arbeit des Teams beeinträchtigen.
- Der Scrum Master ist nicht für den Erfolg des Projekts verantwortlich; seine Aufgabe ist es vielmehr, die Einhaltung der Scrum-Regeln zu gewährleisten.

Scrum-Qualifizierungen auf einen Blick

Scrum Master/Product Owner:

Die Titel Certified Scrum Master (CSM) und Certified Scrum Product Owner (CSPO) setzen die Teilnahme an einem zweitägigen Training voraus, das die theoretischen Grundlagen des Vorgehensmodells mit praktischen Anwendungsfällen verbindet. Eine Prüfung am Ende des Trainings ist für 2010 geplant, um die Qualität der Zertifizierung zu erhöhen.

Scrum Practitioner:

Wer sich Scrum Practitioner nennen möchte, muss nachweisen, dass er mindestens ein Jahr lang in Scrum-Projekten gearbeitet hat und über die Zertifizierung zum CSM beziehungsweise CSPO verfügt. Darüber hinaus muss er einen Report schreiben, in dem er seine Erfahrungen schildert und bewertet.

Scrum Trainer:

Um als Scrum Trainer zugelassen zu werden, muss der Bewerber folgende Unterlagen bei der Scrum Alliance einreichen und bestimmte Anforderungen erfüllen:

- die Zusammenfassungen der eigenen Erfahrungen mit Scrum in einem Resümee,

- die detaillierte Beschreibung der eigenen Trainingserfahrungen,
- eigenes Trainingsmaterial,
- Referenzen von mindestens zwei Co-Trainern, mit denen er ein gemeinsames Training absolviert hat, sowie
- weitere Unterlagen, die die Qualität des Anwärters belegen.

Scrum Coach:

Weltweit gibt es nur zwanzig Scrum Coaches (in Deutschland nur zwei: Andrea Tomasini und Christoph Matthis), da die Anforderungen für diese Zertifizierung sehr hoch sind. Ein Bewerber muss folgende Voraussetzungen erfüllen:

- mindestens 1500 Stunden Scrum-Coaching/-Beratung in den letzten fünf Jahren nachweisen können,
- zwei Kunden benennen, die seine Coaching-Skills bestätigen und beschreiben,
- eine umfangreiche Beschreibung der Projekte und Unternehmen, bei denen er Scrum eingeführt hat, liefern sowie
- in einem Interview zur Analyse und Verifizierung der beschriebenen Vorgehen Rede und Antwort stehen.

Quelle: www.ScrumAlliance.org

wenn das Team oder ein anderer am Projekt Beteiligter (Stakeholder) die Scrum-Regeln verletzt. Während ein Teamleiter häufig Aufgaben verteilt und für deren Erledigung verantwortlich zeichnet, versucht ein erfahrener Scrum Master, dem Team Hinweise und Impulse zu geben, den richtigen Weg einzuschlagen, das richtige Verfahren anzuwenden oder die richtige Technologie auszuwählen. Kurz gesagt, der Scrum Master agiert eher als Team-Coach denn als Teamleiter.

Schwierigkeiten ausräumen

Eine weitere wichtige Aufgabe des Scrum Master ist es, Schwierigkeiten (Impediments), die das Team bei der

Arbeit behindern oder stören, aus dem Weg zu räumen. In der Regel lassen sich die Probleme in drei Kategorien einteilen.

In die erste Gruppe fallen solche, die das Team nicht lösen kann. Beispielsweise kann es keine Performance-Tests durchführen, wenn keine Hardware da ist. Oder die IT-Abteilung stellt keinen Bugtracker zur Verfügung. Möglicherweise ist die bestellte Software noch immer nicht eingetroffen. Oder aber der Vertriebsleiter war schon wieder da und wollte, dass jemand „schnell mal“ dieses oder jenes Feature einbaut.

In die zweite Gruppe fallen Schwierigkeiten, die durch die Organisationsstruktur oder durch strategische Entscheidungen entstehen. Vielleicht ist das Büro für die wichtigen

Meetings oder die Teamarbeit nicht geeignet – beispielsweise weil es keine Medien gibt. Ein oft anzutreffender Fehler ist, dass der Scrum Master Personalverantwortung gegenüber den Teammitgliedern hat. Dies ist häufig ein Überbleibsel aus der klassischen Projektleiterrolle, führt bei Scrum jedoch immer zu Interessenskonflikten und geht gegen den Grundsatz: Das Team in Scrum ist eine Managementrolle und gleichberechtigt neben dem Scrum Master und dem Product Owner. Ein weiterer Aspekt kann die für das neue Projekt unzureichende Internet-Bandbreite sein.

Schließlich gibt es drittens die Probleme von Einzelpersonen. Jemand braucht Hilfe beim Debugging. Ein anderer kann eine Aufgabe allein nicht lösen und benötigt jemanden

zum Pair Programming. Wieder ein anderer muss seinen Rechner neu einrichten.

Obwohl ein Scrum Master bestimmte Aufgaben nicht selbst erledigen kann und soll, ist er dennoch verantwortlich für die Beseitigung und Klärung des Problems, und muss angemessene Maßnahmen ergreifen. Diese Aufgabe ist häufig zeitaufwendig und erfordert ein gewisses Maß an Durchsetzungsvermögen und Rückgrat. Der Scrum Master sollte optimale Arbeitsbedingungen für das Team schaffen und so gewährleisten, dass es die Sprint-Ziele erreichen kann – also die für einen kurzen, Sprint genannten, Zeitraum definierten Vorhaben.

Ein Träger zweier Hüte

Bei der Einführung von Scrum kommt immer wieder die Frage auf, ob der Scrum Master Teil des Entwicklerteams und dementsprechend in die Entwicklung eingebunden sein kann. Die Antwort auf diese Frage ist ein klares Nein. Je nach Projektsituation ist zwar denkbar, dass der Scrum Master des einen Teams parallel als Entwickler in einem anderen Team arbeitet, wichtig ist jedoch: Scrum baut auf die direkte Kommunikation, die



Foto: Jim Rider

Wer ist hier der Agile Leader (Abb. 1)?

Tutorialinhalt

Teil I: Die Rolle des Product Owner

Teil II: Die Rolle des Scrum Master

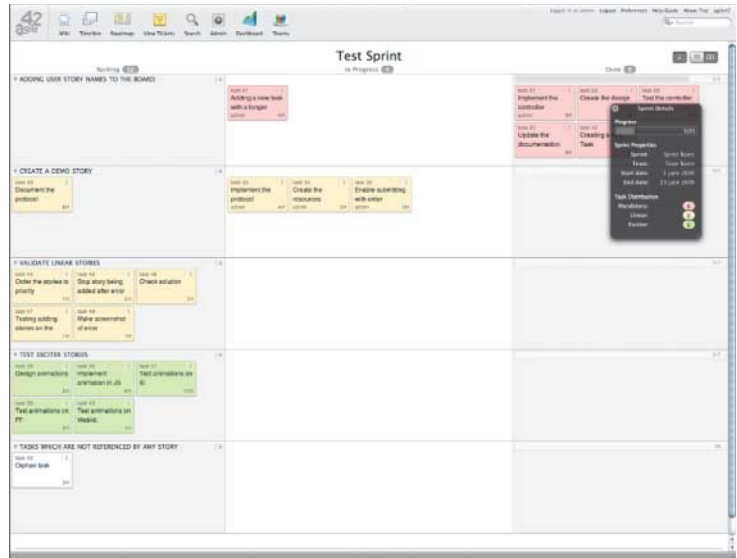
Teil III: Die Rolle des Teams

eindeutige Zusage (Commitment), bestimmte Aufgaben (Tasks) in einer vorgegebenen Zeit zu erledigen, sowie klare Strukturen. Eine Person, die gleichzeitig zwei verschiedene Hüte aufhat, kann aber im Zweifelsfall nicht unbeteiligt und objektiv urteilen und handeln. Selbst die Entscheidung, ob eine Task erledigt oder ein Impediment ausgeräumt werden soll, würde zwangsläufig für die Personalunion Scrum Master/Entwickler zu Interessenkonflikten führen.

Die Verantwortung für den Erfolg des Projekts trägt nicht der Scrum Master. Das Team geht gegenüber dem Product Owner ein Commitment ein, bestimmte Anforderungen, sogenannte User Stories, während des nächsten Sprint umzusetzen. Demzufolge sind das Team und der Product Owner verantwortlich für das Produkt und den Erfolg des Projekts. Der Scrum Master hingegen ist zuständig dafür, dass das Team Scrum als Projektmanagement-Methode annimmt und nach deren Regeln arbeitet. Ein passender Vergleich wäre der zu einem Coach im Sport. Er bereitet das Team auf das Spiel vor, aber wenn das Match beginnt, ist das Team selbstverantwortlich und muss allein zeigen, was es kann.

Ein Scrum Master nimmt sich als Person zurück, coached das Team, themati-

Das Online-Whiteboard des Scrum-Werkzeugs Agilo for Scrum unterstützt den Scrum Master und das Team beim Sprint Planning Meeting sowie den Daily Scrums. In verteilten Team ersetzt es das Whiteboard (Abb. 3).



siert Fehler – auch persönlicher Natur – und schafft so eine Teamkultur, die durch Zusammenarbeit, sowohl innerhalb des Teams als auch gegenüber den anderen Stakeholdern, und durch eine offene Kommunikation bezüglich auftretender Schwierigkeiten geprägt ist.

Scrum im gesamten Unternehmen

Obwohl sich seine Aufgabe auf die Einhaltung des Prozesses konzentriert, trägt er dennoch entscheidend zum Projekterfolg bei. Denn: ohne Scrum Master kein Scrum und ohne Scrum kein Projekterfolg.

Sowohl in der Scrum Alliance [b] als auch in vielen öffentlichen Blogs wird derzeit viel über den Begriff des Scrum Master und über seine Bedeutung bei der Einführung von Scrum diskutiert. Die Bedeutung und der Einsatz von Scrum haben sich in den letzten Jahren verändert. Dem

Prinzip „Inspect and Adapt“ folgend, findet Scrum nicht mehr nur im Rahmen einzelner Projekte statt, sondern entwickelt sich mehr und mehr zur Managementmethode, die die Kommunikation und Zusammenarbeit im gesamten Unternehmen verbessert und so die Produktivität steigert. Oft hört man Begriffe wie „Scrum of Scrums“ oder „Enterprise Scrum“, die deutlich machen, dass dieses Vorgehensmodell weit über die Grenzen eines einzelnen Entwicklungsprojekts hinausgewachsen ist.

Zertifizierungen ja oder nein

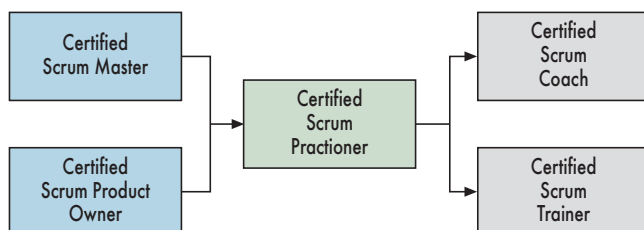
Abhängig von der Komplexität der Unternehmens- und Projektsituationen erfordert die Einführung von Scrum häufig die Analyse der strategischen Ausrichtung, der Prozesse und der Ziele einer Firma. Man muss Chancen und Risiken abwägen und bewerten sowie eine klare Roadmap für die Einführung und Umsetzung von Scrum definieren. Diese Tätigkeiten erfordern ein hohes Maß an Prozess- und Projekt-Know-how sowie Erfahrungen aus dem Umfeld der Unternehmensberatung. Ob jemand in der Lage ist, Scrum einzuführen, hängt von seinen Erfahrungen und Fähigkeiten ab und

nicht vom „Titel“ Scrum Master oder einer Zertifizierung.

Oft stellt sich die Frage, ob eine Zertifizierung zum Scrum Master nötig ist oder nicht. Die Antwort darauf heißt nicht eindeutig ja oder nein. Es kommt darauf an, welches Ziel jemand mit einer Zertifizierung verfolgt. Die notwendigen Trainings werden alle von CSTs (Certified Scrum Trainers) durchgeführt, was für sich genommen für eine hohe Qualität der Ausbildung bürgt. Inwieweit die Teilnehmer jedoch sich selbst und eigene Erfahrungen einbringen oder ob sie nur an dem Training teilnehmen, um ihren Lebenslauf aufzuwerten, darüber sagt die Zertifizierung nichts aus.

Ziel der Ausbildung zum CSM und CSPO (siehe Kästen „Scrum-Qualifizierungen“) ist es, den Teilnehmern die Grundlagen des Scrum-Framework zu vermitteln und zu lehren, wie die Verantwortlichkeiten auf die drei Rollen, drei Artefakte und drei Zeremonien von Scrum aufgeteilt sind. Wer mit dieser Erwartungshaltung in ein Zertifizierungstraining geht, wird sicherlich nicht enttäuscht.

Bedingt durch den großen Zuspruch, den Scrum derzeit erfährt, lassen sich fast täglich neue Scrum-Tools auf dem Markt finden. Wobei der Begriff nicht nur Softwarewerkzeuge bezeichnet, sondern Hilfsmittel jeder Art, die



Vom Scrum Master zum Scrum Coach führt ein langer Weg mit vielen Erfahrungen (Abb. 2).

Onlinequellen

- [a] Tim Listers Vortrag zur Rolle des Agile Leader
www.infoq.com/presentations/agile-leadership-tim-lister
- [b] Scrum Alliance Webseite
www.scrumalliance.org/user_groups
- [c] Scrum-Tools
www.agile42.com/cms/pages/tools/
- [d] Mitglieder der Scrum Alliance
www.scrumalliance.org/community/members

die Arbeit von Scrum Master, Product Owner und Team in Vorbereitung und während der Sprints unterstützen. Zum einen wären da das nicht weg-zudenkende Whiteboard und die praktischen Klebezettel, die sowohl im Sprint Planning Meeting als auch während der Daily Standup Meetings einen Überblick über den aktuellen Stand des Sprint geben.

Für alle Teams, die verteilt arbeiten, gibt es eine Online-variante des Whiteboard als Software (siehe Abbildung 3).

Des Weiteren unterstützen praktische Kartenspiele wie „Scrum Planning Poker“ für das Schätzen von Aufwänden [c] oder das „Business Value Game“ für die Priorisierung von Anforderungen die Scrum-Teams auf einfache und doch effektive Weise.

Bezüglich der Tools gilt in jedem Fall: Vor dem Einsatz einer Scrum-Software sollte das Team die Methode beherrschen und verinnerlicht haben. Dann erweist sich ein Scrum-Tool in der Regel als nützlich,

egal, ob es um die Erfassung, Versionierung und Historisierung von Anforderungen, um die Beschreibung von User Stories oder Tasks oder um die Planung des Sprint geht. Eine gute und weitgehend vollständige Liste solcher Werkzeuge mit Bewertungen gibt es bei www.userstories.com.

Fazit

Der Scrum Master spielt insofern eine wichtige Rolle, als er die Interessen der beiden anderen Scrum-Rollen ausbalanciert: Der Product Owner agiert in erster Linie im Hinblick auf das Geschäftsinteresse der Firma (Business Value), und das Team fordert mehr Qualität, für die es verantwortlich zeichnet. Um in diesem Dualismus die Balance zwischen beiden Werten zu wahren, muss der Scrum Master Unterstützer und Verhand-

lungskünstler sein. Er steht nicht zwischen Product Owner und Team, sondern sorgt für eine klare Kommunikation und moderiert bei Konflikten. Auf diese Weise bereitet er den Weg, damit die Produkt- und Sprint-Ziele erreicht werden können. (ka)

MARION EICKMANN

gehört zu den Gründern der agile42 GmbH, die auf das Consulting, Training und Coaching von agilen Vorgehensweisen wie Scrum spezialisiert ist.

Literatur

- [1] Marion Eickmann; Balance halten; Scrum; Teil 1: Die Rolle des Product Owner; iX 8/09, S. 104

 www.ix.de/ix0909122



Anzeige



Teil 3: Die Rolle des Entwicklungsteams

Teamgeist

**Marion Eickmann,
Andrea Tomasini**

Weder das Budget noch Strategien
oder Techniken sind verantwortlich für das
Gelingen von IT-Projekten. Die erfolgreiche
Realisierung steht und fällt mit der Zusammen-
arbeit des Development Team.

Wie die vorangegangenen zwei Tutorial-Teile zu Scrum ([1], [2]) gezeigt haben, kennt diese agile Methode der Softwareentwicklung drei maßgebliche Rollen. Der Product Owner ist zuständig für die Erhebung und Priorisierung der Anforderungen und das damit verbundene Erstellen des Product Backlog (Anforderungskatalog). Der Scrum Master sorgt für die Einhaltung der Scrum-Regeln und hält dem Team den Rücken frei, wenn Probleme, sogenannte Impediments, auftreten. Dritter im Bunde ist das Entwicklungsteam (Development Team) – im Folgenden Team genannt –, dessen Rolle dieser Artikel beleuchtet. Zusammengefasst bilden die drei Rollen das Scrum Team – nicht zu verwechseln mit dem Development Team.

Teams, die nach Scrum arbeiten, sind selbst organisiert, wobei diese Art der Organisation nicht mit Demokratie gleichzusetzen ist. Abhängig von der Projektphase, den Er-

fahrungen und dem Wissen der einzelnen Personen kristallisiert sich situationsabhängig ein „Leader“ heraus. Die Führung des Teams ist nicht vorgeschrieben, sondern ergibt sich aus der Kombination von Fragestellung und den Fähigkeiten der Beteiligten, die in

interdisziplinären Teams zusammenarbeiten.

Selbstorganisierte Einheit

Diese Arbeitsweise hat zwei wichtige Vorteile, auf denen

das „Scrum-System“ basiert. Zum einen steht am Ende eines jeden Sprint ein „potenziell auslieferbares Produkt“ – sprich ein Produkt, das getestet und dokumentiert ist, sich nicht mehr grundsätzlich ändert und so zu jeder Zeit in das Gesamtsystem integrierbar ist (siehe unten). Das macht eine Auslieferung an den Kunden prinzipiell nach jedem Sprint möglich. Zum anderen erlaubt die Kombination diverser Fähigkeiten eine Betrachtung der Aufgabenstellungen aus unterschiedlichen Blickwinkeln. Das Team behält das Gesamtbild im Auge, erkennt Probleme eher und findet schneller Lösungen.

Ein Team, im Sinne von Scrum, ist nicht nur eine Gruppe von Personen, die zufällig in derselben Firma, Abteilung oder demselben Büro arbeitet und ein gemeinsames Ziel verfolgt. Der Aufbau eines solchen Kollektivs setzt Zeit voraus. Bruce Tuckman präsentierte 1965 ein einfaches Modell, demzufolge die

Teambildung in vier Phasen erfolgt: Bildung des Teams (Forming), Erkennen und Lösen von Konflikten (Storming), Aufgabenverteilung (Norming) und Zusammenarbeit (Performing). Genauer betrachtet bedeutet das für die einzelnen Phasen Folgendes:

Forming: Die Gruppe kommt zusammen, lernt sich kennen, und Beziehungen bilden sich heraus. Jede(r) sucht seinen beziehungsweise ihren Platz im Team.

Storming: Die Teammitglieder identifizieren Unstimmigkeiten bezüglich des Vorgehens und der Methoden, die jeder für sich als richtig bewertet. Das führt zu Konflikten, die offene Diskussionen hervorrufen und von jedem erfordern, den eigenen Standpunkt klar zu vertreten. Es gilt außerdem, eigene Grenzen zu erkennen.

Hilfestellung von außen

In dieser Phase ist es äußerst wichtig, das Team zu coachen. Das Team muss lernen, Gründe für Konflikte zu erkennen und Lösungsstrategien zu entwickeln, die auf Best Practices beruhen sollten. Ein guter Team Coach darf in dieser Situation nicht aktiv in die Gruppendynamik eingreifen. Er wirkt lediglich von außen ein und leistet Hilfestellung, die den Einzelnen befähigen soll, selbstständig das richtige Vorgehen zu finden. In Scrum nimmt der Scrum Master diese Rolle ein. Er sorgt dafür, dass das Team effektiv und produktiv wird, in-

dem er darauf achtet, dass es die Scrum-Prinzipien einhält.

Norming: In dieser Phase arbeitet das Team daran, alle nötigen Best Practices für das gemeinsame Vorgehen zu definieren und zu implementieren, um alles, was die tägliche Arbeit stören könnte, zu beseitigen. In dieser Phase identifiziert die Gruppe die Fähigkeiten, Expertise und Talente, die die einzelnen Mitglieder einbringen können.

Performing: In der letzten Phase hat das Team Unabhängigkeit und Selbstbewusstsein erlangt und kann die meisten Schwierigkeiten, die sich während der Produktentwicklung ergeben, selbst ausräumen. Die Reife und Effizienz der angewandten Best Practices wird kontinuierlich verbessert und optimiert, und das Team erreicht die ihm eigene Geschwindigkeit und Produktivität.

Die Rolle, die das Entwicklungsteam einnimmt, unterscheidet Scrum von vielen anderen Software-Entwicklungsmethoden. Während bei den meisten Vorgehensmodellen das Team Aufgaben zugeteilt bekommt und ein Teamleiter festlegt, wie welche Arbeiten zu erledigen sind, organisiert sich das Team in Scrum selbst. Bei der Planung eines Sprint stellt der Product Owner den Product Backlog mit den von ihm priorisierten Anforderungen vor. Gemeinsam werden wichtige Details und Besonderheiten besprochen, damit das Team ermessen kann, welche der priorisierten Anforderungen es im anschließenden Sprint erledigen kann. Die

Entscheidung darüber beruht auf Verhandlung und Kommunikation zwischen Team und Product Owner. Gemeinsam versuchen die Inhaber dieser beiden Scrum-Rollen, einen Weg zu finden, ein möglichst optimales Ergebnis zu erzielen.

Das Team als Manager

Das Team agiert hier als Manager. Niemand schreibt ihm vor, was es zu tun hat oder welche Requirements in den nächsten Sprint einfließen, sondern es wählt selbst aus dem priorisierten Backlog die User Stories (Anforderungen oder Items) aus, die es im nächsten Sprint für erreichbar hält. Das ist eine gravierende Änderung gegenüber traditioneller Softwareentwicklung. Das Team ist von Anfang an involviert und allein verantwortlich für die Implementierung der ausgewählten Backlog Items während des Sprint.

Ein solches Vorgehen stellt sicher, dass das Team die Verantwortung – oder das Besitzrecht („Ownership“) – für das Produkt innehat. Es wird lernen (learning by doing) was es bedeutet, ein (potenziell) auslieferbares Produkt zu entwickeln, denn zu diesem Zeitpunkt hat es sich bereits dazu verpflichtet (committed), alles zu tun, um das gesteckte Sprint-Ziel zu erreichen.

Gelingt es dem Team nicht, bis zum Ende des Sprint alle gewählten Funktionen fertigzustellen, überlegt es gemeinsam mit dem Product Owner

und dem Scrum Master, warum es das Ziel nicht erreichen konnte. Die Gründe dafür werden aufgedeckt und besprochen, um sie beim nächsten Mal rechtzeitig zu erkennen und – noch wichtiger –, um Lösungsstrategien zu entwickeln, die sicherstellen, das unter allen Umständen und in der kürzesten Zeit immer eine bestimmte Qualität erreicht werden kann. Schließlich überlegt man gemeinsam, wie viel Aufwand für die nicht erledigten Aufgaben (undone work) im nächsten Sprint einzuplanen ist.

Traditionell versus inkrementell

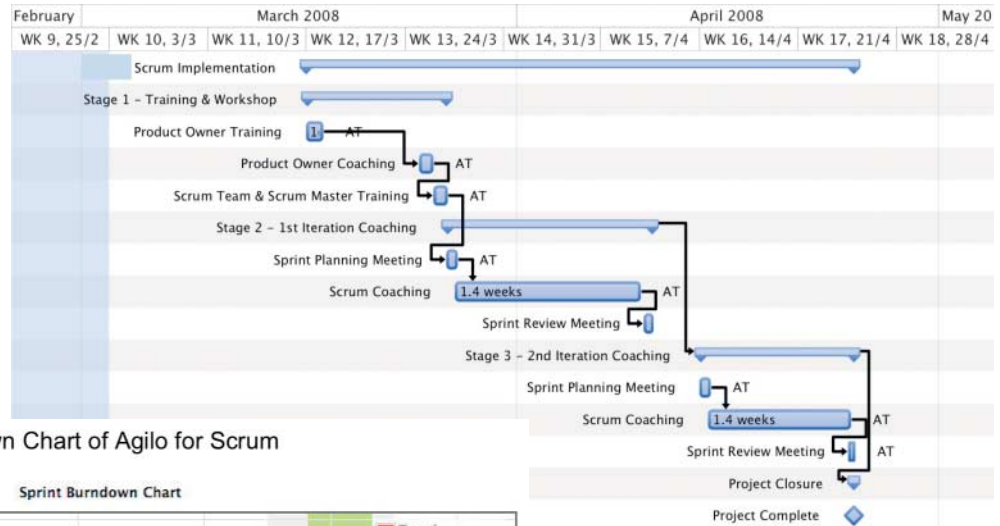
Ein fundamentaler Unterschied zwischen agilen Methoden und traditioneller Vorgehensweisen ist, dass die Entwicklung eines Produkts iterativ erfolgt und die Programmierer auch bereits implementierte Funktionen später noch verbessern können. In traditionellen Vorgehensweisen, typischerweise Wasserfallmodellen, entsteht das Produkt in Phasen, die jeweils einen spezifischen Fokus haben.

Als Beispiel hierfür soll die Phase der Anforderungserhebung und -analyse dienen. Beim phasenorientierten Vorgehen versucht das Team vor dem Beginn der eigentlichen Entwicklung, alle Anforderungen detailliert zu beschreiben und festzulegen (Pflichten/Lastenheft). Der Begriff „versucht“ ist hier absichtlich gewählt, denn in der Regel basiert diese Aufstellung auf Annahmen und Vermutungen: Nur in wenigen Fällen sind alle Änderungen und neu aufkommenden Wünsche an das Produkt von vornherein erkennbar. Da es im klassischen Wasserfall-Vorgehen nicht erlaubt ist, in den Phasen zurückzuspringen (Wasser fließt ja auch nicht bergauf), erfordert diese Phase viel Zeit und Aufwand, um in einer möglichst vollständigen Anforderung



- Neben dem Product Owner und dem Scrum Master kommt dem Development Team in einem Scrum-Projekt die dritte wichtige Rolle zu.
- Das Team organisiert sich selbst und ist dafür verantwortlich, dass am Ende eines Sprint ein gut dokumentiertes Produkt steht, das getestet ist und sich in die bisherige Software integriert.
- Da das Development Team iterativ und inkrementell arbeitet, kann es neue Anforderungen leichter umsetzen und gleichzeitig das angestrebte Ergebnis im Auge behalten.

Während der Gantt Chart (rechts) zeitlich aufeinanderfolgende Phasen grafisch darstellt, zeigt Agilos Burn Down Chart (unten) den aktuellen Status des Sprint (Abb. 1).



Burn Down Chart of Agilo for Scrum

Start: 16.04.2009 End: 28.04.2009 Status:

Sprint Burndown Chart



rungsliste und somit einer guten Definition des Produkts zu resultieren. Erst anschließend kann die eigentliche Entwicklung beginnen.

Die steigende Komplexität der Produkte, die Geschwindigkeit mit der der Markt heute neue Funktionen fordert, und viele andere Einflüsse lassen den Unsicherheitsfaktor bei der Vorabdefinition der Anforderungen beziehungsweise des Produkts immer größer werden. Um die Komplexität in den Griff zu bekommen, führen Teams beim phasenorientierten Vorgehen häufig mehr und mehr Formalien und starre Prozesse ein, was sich negativ auf die Geschwindigkeit des Entwicklungsprozesses auswirkt. Ein weiterer Nachteil dieses Ansatzes ist, dass das Produkt als Ganzes implementiert und getestet wird. Das bringt einen erheblichen Aufwand bezüglich Integration und Fehlerbehebung mit sich.

Beim Wasserfall ist das wohl größte und bekannte

Problem, dass die Entwickler das fertige Produkt erst zu einem sehr fortgeschrittenen Zeitpunkt sehen. Da sie Fehler so erst spät erkennen, sind Änderungen teuer und zeitaufwendig. Am Beispiel des Anforderungsmanagements bedeutet dies, dass eine falsche Anforderung sich durch die gesamte Produktentwicklung ziehen kann.

Neue Ideen gleich umsetzen

Agile Vorgehensweisen wie Scrum basieren auf iterativem und inkrementellem Vorgehen, das es ermöglicht, die „wertvollsten“ Funktionen stets im Auge zu behalten. Die iterative Entwicklung erlaubt es, das Produkt jederzeit aktuellen Anforderungen entsprechend neu zu definieren und zu verbessern. Das Team kann neue Ideen zeitnah umsetzen und Entscheidungen aufgrund von Fakten (die aktuelle Produktversion) und

nicht auf der Basis von Vermutungen (die Vorstellung oder Idee einer Funktion) treffen.

Damit ein Team am Ende einer jeden Iteration – in Scrum dauert ein Sprint zwischen zwei und vier Wochen – eine fertige Produktversion vorlegen kann, muss es die neuen Funktionen entwerfen, implementieren, in die bereits vorhandene Software integrieren und testen. Nur wenn diese Voraussetzungen erfüllt sind, gilt der Sprint als erfolgreich. Dafür ist eine funktionsübergreifende Arbeitsweise des Teams unumgänglich. Der Aspekt der Selbstorganisation kommt hier ebenfalls zum Tragen, denn die extrem strengen Deadlines setzen voraus, dass das Team die Best Practices kontinuierlich verbessert und Lösungen gemeinsam findet.

Da das Produkt erst nach erfolgreichem Testen als fertig (done) gilt, spielt die Testautomatisierung eine entscheidende Rolle. Ohne sie wird es

dem Team nicht gelingen, innerhalb der kurzen Sprint-Phasen die Qualität aller integrierten Neuentwicklungen zu sichern. Bezüglich des Testens sind in den letzten 15 Jahren, ursprünglich initiiert durch das Extrem Programming (XP), viele Best Practices entstanden, darunter Automated Testing, Continuous Integration und Automated Build.

Immer ein großes Ganzes

Eine der größten Errungenschaften der iterativen, inkrementellen Softwareentwicklung ist, dass am Ende einer jeden Iteration ein komplettes Produkt steht, nicht nur einzelne, voneinander unabhängige Teile. Das Resultat ist lauffähig und kann sowohl in möglicherweise bereits existierende Software integriert werden als auch als Basis für die weitere Entwicklung dienen. Der Product Owner ist



Fünf Aspekte, die sich gegenseitig beeinflussen, können die Teamarbeit behindern (Abb. 2).

nicht auf Annahmen angewiesen, welche Funktion richtig, falsch oder zu verbessern ist. Er kann die Entwicklung kontinuierlich in die gewünschte Richtung steuern.

Für das Team, das die Verantwortung für die Entwicklung trägt, gilt: Besser ein oder zwei Features weniger entwickeln, aber dafür immer das fertige Stück Software im Auge behalten. Die anfängliche Priorisierung gewährleistet, dass Funktionen, die das Team nicht während des Sprint implementieren kann, nicht die sind, die den höchsten Stellenwert haben. Die Entscheidung darüber, ob das Produkt den Qualitätsanforderungen genügt und somit der Sprint erfolgreich war, trifft am Ende des Sprint der Product Owner.

Fünf Aspekte, die behindern

Trotz aller Vorsätze und Bemühungen kann es passieren, dass ein Team nicht produktiv arbeiten kann. Patrick Lencioni [3] beschreibt fünf Aspekte, die verantwortlich dafür sind, dass ein Team nicht erfolgreich agiert und „funktionierte“ (Abb. 2).

Als Erstes nennt er fehlendes Vertrauen. Teammitglieder müssen einander offen gegenüberstehen. Das gelingt nicht, wenn sie nicht in der Lage sind, den anderen Teammitgliedern sowohl positives als auch negatives Feedback zu geben. Ohne Vertrauen las-

sen sich keine Konflikte lösen, die beispielsweise entstehen können, wenn jemand neue Ideen einbringt. Die Angst vor Konflikten ist der zweite Aspekt, der die Arbeit behindert. Zwanghaft taktische Diskussionen mit genau überlegten Kommentaren sind die Folge, wenn die Angst, etwas falsch zu machen, oder das Bestreben, Konflikten aus dem Weg zu gehen, überwiegt.

Fehlendes Commitment führt Lencioni als dritten Punkt an. Offene Diskussionen und freie Meinungsäußerung sind Voraussetzungen für ein erfolgreiches Scrum-Projekt. Wenn ein Team keine Rechenschaft ablegen muss, behindert das ebenfalls die Zusammenarbeit. Für etwas, hinter dem man nicht steht, will man auch nicht verantwortlich gemacht werden. Schließlich wirkt es sich nachteilig aus, wenn die Mitarbeiter das Gesamtbild nicht vor Augen haben. Die Einzelnen müssen ihre individuellen Wünsche hinter denen des gesamten Teams zurückstellen. Nur dann gelingt es, ein gemeinsames Ziel zu erreichen. Die beschriebenen Faktoren bedingen sich gegenseitig. Ohne Vertrauen kein Mut zu Konflikten, ohne Commitment kein Big Picture.

Mit Scrum verteilt entwickeln

Zu guter Letzt ein paar Sätze zum Thema Scrum und Off-

shoring. Kommunikation spielt eine wichtige Rolle, daher ist es nachvollziehbar, dass bei Scrum wie bei vielen anderen agilen Methoden die Entwicklung an einem gemeinsamen Standort (Co-Located Development) bevorzugt und angeregt wird. Langfristig ist das nicht nur kostengünstiger, sondern auch effektiver. Man muss sich weder über Zeit- zonen, noch über unterschiedliche Kulturen oder andere Kommunikationsbrüche Gedanken machen.

Tools erleichtern den Alltag

Die Realität zeigt aber, dass viele Unternehmen nicht auf verteilte Entwicklung verzichten können. In diesem Fall gilt es, einige Besonderheiten zu beachten. Zum einen sollte das Team möglichst als Ganzes an einem Standort arbeiten, was die Daily Scrums erleichtert und Voraussetzung für agile Methoden wie das Pair Programming ist.

Gute Online-Werkzeuge können zum anderen die Kommunikation und die Zusammenarbeit unter den Standorten erleichtern. Beispielsweise durch eine Sprint-Planung per Videokonferenz oder dadurch, dass die Tools an allen Standorten dieselbe Information bereitstellen. Webbasierte Scrum Tools unterstützen die Zeit- zonen, stellen versionierte Daten zur Verfügung und erleichtern mit einem Online-Whiteboard die täglichen Meetings und Planungen. Wichtig ist und bleibt: Die Methode kommt vor dem Werkzeug, aber bei der verteilten Entwicklung vereinfachen die richtige Soft-

ware und eine gute Infrastruktur die täglichen Aufgaben.

Fazit

Scrum nutzt die Synergien, die bei der Bildung und der gemeinsamen Arbeit von Teams entstehen. Überträgt man diesen Gedanken von einzelnen Teams, in denen fünf bis maximal zehn Personen Hand in Hand auf ein Ziel hinarbeiten, auf Abteilungen, die übergreifend zusammenarbeiten, bis hin zur gesamten Organisation, ist die Voraussetzung für die erfolgreiche Realisierung von Projekten geschaffen. So einfach sich das Vorgehen nach Scrum anhört, die Umsetzung steht und fällt mit dem Willen der Beteiligten, Änderungen zu akzeptieren. Denn die sind unvermeidlich – sowohl bezüglich der Firmenkultur, Karriereplanung, Teamstrukturen als auch der Projektleitung. (ka)

MARION EICKMANN

gehört zu den Gründern der agile42 GmbH, die auf das Consulting, Training und Coaching von agilen Vorgehensweisen wie Scrum spezialisiert ist.

ANDREA TOMASINI

gehört ebenfalls zu den Gründern der agile42 GmbH und ist einer der 20 zertifizierten Scrum Coaches weltweit.

Literatur

- [1] Marion Eickmann; Balance halten; Teil 1: Die Rolle des Project Owner; iX 8/09, S. 104
- [2] Marion Eickmann; Facettenreich; Teil 2: Die Rolle des Scrum Master; iX 9/09, S. 122
- [3] Patrick Lencioni; The Five Dysfunctions of a Team; San Francisco 2002, ISBN 0-7879-6075-6



Tutorialinhalt

- Teil I: Die Rolle des Product Owner
- Teil II: Die Rolle des Scrum Master
- Teil III: Die Rolle des Teams