

eingereicht am: 20.10.2018
überarbeitete Version: 05.12.2018

Scrum – der Weg zum agilen Projektmanagement

Ralf T. Kreuzer

Ein unverzichtbares Werkzeug des agilen Projekt-Managements ist Scrum. Der Begriff „Scrum“ steht im Englischen für „Gedränge“ beim Rugby-Spiel und soll zum Ausdruck bringen, dass diese Methode durch Verdichtung eine höhere Schnelligkeit, Kampfkraft, Beweglichkeit und gleichzeitig gute Ergebnisse erzielen möchte. Hierzu ist es erforderlich, dass ein hochqualifiziertes und eingefleischtes Team eng zusammenarbeitet, sich schnell an veränderte Spielsituationen anpasst und sich hierbei selbst organisiert. Das setzt allerdings ein umfassendes Empowerment des Teams voraus. Gleichzeitig sind wenige, aber präzise Regeln zu beachten – ganz so wie beim Rugby auch! In diesem Beitrag wird aufgezeigt, wie dieses Konzept im Rahmen des Entrepreneurial Marketing nutzbar gemacht werden kann.

Scrum is an indispensable tool for agile project management. The term "scrum" stands for "crowd" in the rugby game and should express that this method wants to achieve a higher speed, fighting power, mobility and at the same time good results. For this it is necessary that a highly qualified and ingrained team works closely together, adapts quickly to changing game situations and organizes itself. However, this itself requires a comprehensive empowerment of the team. At the same time, a few precise rules have to be observed – just like in rugby! This article shows how this concept can be used in the context of entrepreneurial marketing.

Prof. Dr. Ralf T. Kreuzer ist seit dem Jahr 2005 Professor für Marketing an der Berlin School of Economics and Law sowie Marketing und Management Consultant, Trainer und Coach. Zuvor war er 15 Jahre in Führungspositionen bei Bertelsmann, Volkswagen und der Deutschen Post tätig. Kreuzer hat durch Publikationen und Keynotes maßgebliche Impulse zu den Themen CRM, Online-Marketing, digitaler Darwinismus, digitale Transformation, Change-Management und Künstliche Intelligenz gegeben und viele Unternehmen beraten sowie Führungskräfte trainiert. kreuzer.r@t-online.de

1. Warum Scrum für das Entrepreneurial Marketing von Bedeutung ist

Die Mehrheit der Leser und Leserinnen wird mit den **klassischen Instrumenten des Projekt-Managements** gut vertraut sein; schließlich wurden diese seit Jahrzehnten weitgehend unverändert gelehrt und eingesetzt. Um der zunehmenden Veränderungsgeschwindigkeit bei den heute zu meisternden Herausforderungen gerecht zu werden, müssen diese klassischen Vorgehensweisen im Zuge eines Entrepreneurial Marketing allerdings zunehmend durch neue Methoden ergänzt bzw. abgelöst werden (vgl. weiterführend Kreuzer 2018, S. 207ff.; Röpstorff/Wiechmann 2016; Sutherland 2015).

Die hier vorgestellte Methode Scrum gehören in den Themenkreis des **agilen Projekt-Managements**. Wie sich das agile vom klassischen Projekt-Management unterscheidet, zeigt Abb. 1.

Klassisches Projekt-Management	Agiles Projekt-Management
<ul style="list-style-type: none"> ▪ Anforderungen sind – scheinbar – von Anfang an bekannt. ▪ Veränderungen der Anforderungen während des Projektverlaufs sind nicht vorgesehen. ▪ Veränderungen der Anforderungen (Change-Requests) gehen mit hohen zusätzlichen Kosten einher. ▪ Anforderungen werden primär „technisch“ beschrieben („Domäne der Techniker“). ▪ Der Entwicklungsprozess ist sequentiell angelegt. ▪ Der Prozess des Projektmanagement folgt klaren und unveränderlichen Regeln und Normen. ▪ Dem Kunden wird nur das finale Ergebnis präsentiert; Zwischenstände sieht der Kunde nicht. ▪ Wenn es im Projektverlauf „hakt“, werden definierte Milestones verschoben. ▪ Es kommen große Teams zum Einsatz, die strikt hierarchisch strukturiert sind (Aufgaben werden von oben zugeteilt). ▪ Die Mitglieder des Entwicklungsteams sind ausgewiesene Spezialisten. ▪ Die Teammitglieder sitzen räumlich getrennt und arbeiten an verschiedenen Projekten parallel. ▪ Die Kommunikation läuft über umfangreiche Projektdokumentationen und lange Meetings mit umfassenden To-Do-Listen. ▪ Die Aufwandschätzung erfolgt durch den Projektleiter oder Experten. 	<ul style="list-style-type: none"> ▪ Anforderungen sind am Anfang unscharf – und werden auch so behandelt. ▪ Veränderungen der Anforderungen während des Projektverlaufs sind eingeplant und willkommen. ▪ Spätere Änderungen der Anforderungen sind von vornherein eingeplant und verursachen nur geringe Zusatzkosten. ▪ Anforderungen werden konsequent aus der Nutzer-/Anwenderperspektive beschrieben („User-Stories“). ▪ Der Entwicklungsprozess ist iterativ und inkrementell. ▪ Der Prozess des Projektmanagements wird – basierend auf den Erfahrungen – laufend optimiert. ▪ Dem Kunden werden die fortlaufenden Ergebnisse präsentiert, um sein Feedback einzuholen. ▪ Bei Schwierigkeiten wird der Fokus erhöht und sofort nach einer Problemlösung gesucht. ▪ Kleine Teams werden eingesetzt, die sich in hohem Maße selbst organisieren (Aufgaben werden selbständig übernommen). ▪ Die Teammitglieder verfügen neben Expertenwissen auch über generalistisches Wissen („T-Format“). ▪ Das Team sitzt räumlich zusammen und fokussiert sich auf ein Projekt. ▪ Es herrscht ein hohes Maß an informeller Kommunikation; Standup-Meetings werden durchgeführt. ▪ Die Aufwandschätzung wird konsensual im Team vorgenommen.

Abb. 1 Unterschiede zwischen klassischem und agilem Projekt-Management (Quelle: Kreuzer 2018, S. 208; in Anlehnung an Preußig 2015, S. 41f.)

Um in einer sich dynamisch verändernden Welt bestehen zu können, müssen neue Methoden eingesetzt und gleichzeitig die Kompetenz-Landkarte der Unternehmen weiterentwickelt werden. Welche Veränderungen dabei anzustreben sind bzw. welche Kompetenzen entsprechend aufzubauen sind, zeigt Abb. 2. Die dort beschriebene

nen „heute geforderten Kompetenzen“ liegen den weiteren Ausführungen zugrunde. Dabei dürfte es neu gegründeten Unternehmen viel leichter fallen, von Anfang an die heute besonders geforderten Kompetenzen aufzubauen. Etablierte Unternehmen tun sich damit viel schwerer, weil es dazu häufig eines umfassenden Change-Managements bedarf.

Bisher dominierende Kompetenzen	Heute geforderte Kompetenzen
<ul style="list-style-type: none"> ▪ Extrapolatives Vorgehen i.S. von „Mehr vom Gleichen“ ▪ Perfektion vor Schnelligkeit – Fokus auf Time-to-Market ▪ Hierarchische Organisation – transaktionale Führung ▪ Eher enge Führung über Ziele ▪ Bewahren des Bewährten 	<ul style="list-style-type: none"> ▪ Disruptives Denken und Handeln – Heraustrreten aus der eigenen Komfortzone ▪ Schnelligkeit vor Perfektion – Fokus auf Time-to-Value ▪ Verstärktes Arbeiten in Netzwerken – transformationale Führung ▪ Führung stärker über Vision und „Sinn“ (Purpose) ▪ Mut zu Veränderung und zum „Constant Change“

Abb. 2 Veränderungen in der Kompetenz-Landkarte zum Bestehen in der VUCA-Welt (Quelle: Kreutzer 2018, S. 208)

2. Zentrale Orientierungspunkte der Methode Scrum

Der Begriff „Scrum“ steht im Englischen für „Gedränge“ beim Rugby-Spiel und soll zum Ausdruck bringen, dass diese Methode durch Verdichtung eine höhere Schnelligkeit, Kampfkraft, Beweglichkeit und gleichzeitig gute Ergebnisse erzielen möchte. Hierzu ist es erforderlich, dass ein hochqualifiziertes und eingefleischtes Team eng zusammenarbeitet, sich schnell an veränderte Spielsituationen anpasst und sich hierbei selbst organisiert. Das setzt allerdings ein umfassendes **Empowerment des Teams** voraus. Gleichzeitig sind wenige, aber **präzise Regeln** zu beachten – ganz so wie beim Rugby!

Dieses spezifische Vorgehensmodell ist zunächst vor allem zur agilen Software-Entwicklung eingesetzt worden. Heute wird Scrum bei einer großen Vielzahl von Aufgabenstellungen – auch außerhalb der IT – eingesetzt. Gerade auch im Marketing, im Vertrieb und bspw. bei der gemeinsamen Entwicklung an Plattform-Konzepten kommt Scrum heute zum Einsatz. Wenn nachfolgend meistens von „Produkt“ die Rede ist, dann kann das eine Software, aber auch eine anders geartete Lösung sein.

Das hier zugrunde liegende Vorgehen basiert auf den Werten des so genannten **Manifestes für agile Software-Entwicklung** (vgl. Beck et al. 2018):

- Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge.
- Funktionierende Software ist von größerer Bedeutung als eine umfassende Dokumentation.
- Eine intensive Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlungen.

- Das Reagieren auf Veränderung ist viel bedeutender als das bloße Befolgen eines einmal entwickelten Plans.

Vor diesem Hintergrund wurde das **Scrum-Vorgehensmodell** entwickelt. Es basiert auf der Erfahrung, dass viele Projekte viel zu komplex und das Einsatzfeld viel zu dynamisch ist, um Anforderungen bereits zu Beginn des Entwicklungsprozesses umfassend definieren und damit planen zu können. Der Grund liegt darin, dass viele Anforderungen sowie mögliche Lösungsideen beim Start eines Projektes noch gar nicht bekannt sind oder bekannt sein können. Um dieses Manko zu überwinden, gilt es im Zuge des Entwicklungsprozesses möglichst früh **belastbare Zwischenergebnisse** zu erzielen.

Konkret bedeutet dies, dass beim Einsatz der Scrum-Methodik nicht nur die Lösung, sondern auch deren Planung inkrementell und iterativ entwickelt werden können. „Inkrementell“ bedeutet, dass der Gesamtprozess in kleinere Schritte zerlegt wird. „Iterativ“ – zu verstehen als „wiederholend“ – zeigt an, dass bestimmte Arbeitsschritte u.U. mehrfach zu durchlaufen sind, um zu einer überzeugenden Lösung zu gelangen.

3. Scrum-Team und weitere Akteure

Beim Scrum sind verschiedene Parteien zu unterscheiden. Im Mittelpunkt des Scrum-Prozesses steht das **Scrum-Team** (vgl. Abb. 3). Es setzt sich zusammen aus einem **Product-Owner** (der eigentliche Auftraggeber), einem **Scrum-Master** und mehreren Personen, die zusammen das **Entwicklungsteam** bilden. Der **Business-Owner** ist der Verantwortliche für die Gesamtorganisation, in die das Scrum-Team eingebettet ist. **Stakeholder** sind vor allem die Personen, für die eine Lösung entwickelt werden soll (meist Kunden bzw. die finalen Anwender).

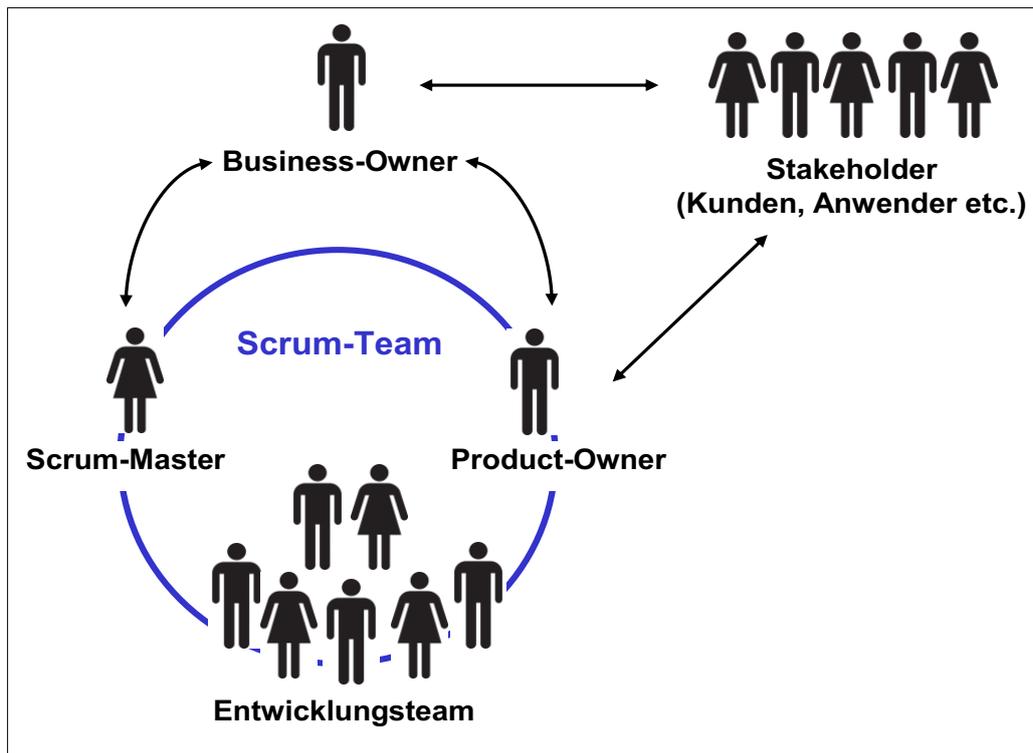


Abb. 3 Scrum-Team und weitere Akteure
(Quelle: Kreuzer 2018, S. 211)

Die Rollen der **Mitglieder des Scrum-Teams** werden hier vorgestellt:

- **Product-Owner:** Der Product-Owner trägt die Gesamtverantwortung für das Projekt – sowohl hinsichtlich der faktischen Lösung als auch für den wirtschaftlichen Erfolg (vgl. Abb. 3). Er repräsentiert quasi den internen Auftraggeber. Dazu definiert und priorisiert er die im Zuge des Entwicklungsprozesses zu erreichenden Produkteigenschaften (das so genannte Product-Backlog). Hierzu führt der Product-Owner die Anforderungen an das Produkt in Zusammenarbeit mit dem Entwicklungsteam und den Stakeholdern zusammen. Um die gewünschte Agilität im Prozess zu erreichen, sollte der Product-Owner mit den notwendigen Kompetenzen ausgestattet sein, um alle notwendigen Entscheidungen eigenständig – und damit auch schnell – treffen zu können.
- **Scrum-Master:** Der Scrum-Master trägt die Verantwortung für das Gelingen des Scrum-Prozesses (vgl. Abb. 3). Hierfür arbeitet der Scrum-Master intensiv mit dem Entwicklungsteam zusammen, meist ohne ihm selbst anzugehören. Die Kernaufgabe des Scrum-Masters besteht darin, dass er die Scrum-Regeln einführt und deren Einhaltung überprüft. Gleichzeitig ist er der Ansprechpartner, um Störungen des Scrum-Prozesses zu überwinden. Der Scrum-Master nimmt damit gegenüber dem Entwicklungsteam die Funktion eines Coaches ein, der in kritischen Situationen unterstützt.
- **Entwicklungsteam:** Für die Lieferung der in den Sprint-Phasen zu erreichenden Produktfunktionalitäten ist das Entwicklungsteam verantwortlich (vgl. Abb. 3). Das Team selbst hat dabei die vereinbarten Qualitätsstandards zu berücksichtigen. Eine Kernaufgabe des Entwicklungsteams ist auch die Aufwandsabschätzung. Hierzu greift das Team auf die Einträge im Product-Backlog bzw. auf das Product-

Backlog-Refinement zurück. Das Entwicklungsteam organisiert sich selbst. Das Team muss sich bspw. nicht vorschreiben lassen, wie es bspw. die Backlog-Einträge umzusetzen hat, sondern kann relativ frei agieren. Dafür ist das erforderliche Empowerment des Teams sicherzustellen.

Das durch die Träger dieser Rollen gebildete **Scrum-Team** tritt mit den Stakeholdern genannten Beteiligten in Kontakt. Diese **Stakeholder** stehen außerhalb von Scrum (vgl. Abb. 3). Zu den Stakeholdern gehören die folgenden Gruppen:

- **Kunden:** Kunden sind die Personen, für die das Produkt entwickelt wird. Kunden können im Unternehmen selbst angesiedelt sein, bspw. als Fachabteilung, für die eine Entwicklung durchgeführt wird. Kunden sind allerdings besonders häufig außerhalb des Unternehmens anzutreffen. Es stellt eine Kernaufgabe des Product-Owners dar, in einem intensiven Dialog mit den Kunden zu stehen, um deren Anforderungen und Wünsche in den Entwicklungsprozess einfließen zu lassen.
- **Anwender:** Von den Kunden unterscheiden sich die Anwender dadurch, dass sie diejenigen sind, die das Produkt tatsächlich nutzen (vgl. Abb. 3). Die Anwender sind – im Gegensatz zum Kunden – von besonderer Bedeutung, weil sie letztlich das Produkt tatsächlich im täglichen Einsatz nutzen werden. Deshalb ist ihr Blick „auf die Dinge“ von großer Wichtigkeit. Folglich sollte der Product-Owner großen Wert darauf legen, dass beim Auseinanderfallen von Anwendern und Kunden beide Parteien in den Sprint-Review einbezogen werden.
- **Business-Owner/Management:** Der Business-Owner ist dafür verantwortlich, dass die Rahmenbedingungen – vor allem die notwendigen Ressourcen – für den Scrum-Einsatz stimmen. Dazu gehört zunächst die Bereitstellung der notwendigen Mitarbeiter (Product-Owner, Scrum-Master und Entwicklungsteam), die ungestört arbeiten können müssen. Außerdem sind die erforderlichen Räumlichkeiten und Arbeitsmaterialien sicherzustellen.

4. Scrum-Prozess

Um den Scrum-Prozess zu verstehen, müssen zunächst für den Ablauf relevante Begriffe der Scrum-Terminologie vorgestellt werden. Im Folgenden sind **drei Artefakte** zu definieren. Der Begriff „Artefakt“ leitet sich aus den lateinischen Begriffen „art/artis“ für „Handwerk“ und „factum“ für „das Gemachte“ ab. Folglich steht Artefakt für von Menschen erzeugte oder veränderte Gegenstände. Im Scrum-Kontext sind es die folgenden Artefakte:

- **Product-Backlog:** Das Product-Backlog stellt eine strukturierte Aufzählung der Anforderungen an das zu entwickelnde Produkt bzw. die gewünschte Lösung dar (vgl. Abb. 4). Diese Anforderungen sind dabei nicht technischer Natur, sondern beschreiben fachlich und aus Anwendersicht, was geleistet werden soll. User-Stories können diese Perspektive sehr gut zum Ausdruck bringen. Dieses Product-Backlog ist nicht statisch, sondern wird im laufenden Prozess kontinuierlich weiterentwickelt. Das Product-Backlog stellt den Ausgangspunkt der Arbeit des Entwicklungsteams dar. Dabei ist das Product-Backlog nie vollständig, sondern ein **Work-in-Progress**. Dort definierte Anforderungen können ergänzt, präzisiert, aber auch gelöscht werden. Zusätzlich können neue Anforderungen

aufgenommen werden. Auch die Prioritäten der jeweiligen Einträge können im Zuge des Prozesses verändert werden. Ihre Definition erfolgt durch den Product-Owner, der sich dabei an Kriterien wie wirtschaftlichem Nutzen, Relevanz für den Anwender, Risiko usw. orientiert. Die Verantwortung für die Entwicklung des Product-Backlogs liegt beim Product-Owner. Dieser verantwortet auch die kontinuierliche Aktualisierung, Verfeinerung und ggf. Neustrukturierung des Product-Backlog. Dieser Prozess wird als **Product-Backlog-Refinement** bezeichnet (vgl. Abb. 4).

- **Sprint-Backlog:** Die Ergebnisse des Sprint-Planning bilden das Sprint-Backlog (vgl. Abb. 4). Es stellt den Plan der in einem Sprint zu erledigenden Aufgaben dar. Dazu werden die für einen Sprint ausgewählten Einträge des Product-Backlogs in einzelne Arbeitsschritte (Tasks) heruntergebrochen. Zu diesen Tasks gehören Aufgaben wie Entwicklung, Test und Dokumentation.
- **Product-Increment:** Unter Product-Increment ist die Summe aller Product-Backlog-Einträge zu verstehen, die während des aktuellen sowie allen vorangegangenen Sprints erstellt wurden (vgl. Abb. 4). Am Ende eines jeden Sprints erfolgt die Lieferung eines fertigen Teilprodukts. Dieses wird in das Product-Increment aufgenommen. Diese Teilleistung soll so ausgestaltet sein, dass sie an den Kunden ausgeliefert werden kann bzw. könnte (Stichwort: Potentially Shippable Product). Dafür muss diese Teilleistung in einem nutzbaren Zustand vorliegen und damit der Definition-of-Done (was zu erreichen war) entsprechen.

Ein Überblick über den **Gesamtablauf eines Scrum-Prozesses** zeigt Abb. 4. Ausgangspunkt des Prozesses ist dabei jeweils das Product-Backlog. Auf die Inhalte der weiteren Phasen wird etwas später eingegangen.

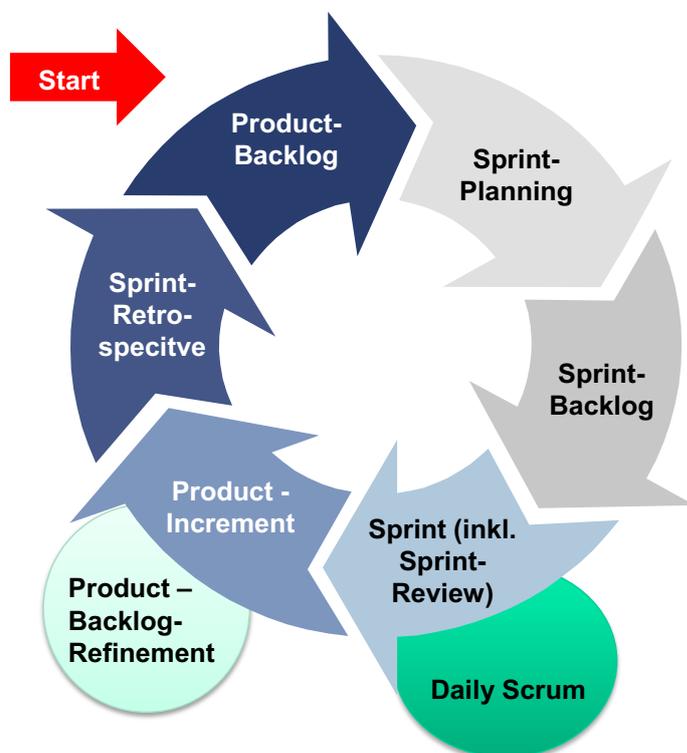


Abb. 4 Gesamtablauf eines Scrum-Prozesses
(Quelle: Kreutzer 2018, S. 214)

Das Vorgehensmodell Scrum basiert auf den folgenden **Aktivitäten**:

- **Sprint-Planning**: Das Sprint-Planning ist eine Kernaufgabe des Entwicklungsteams (vgl. Abb. 4). Die im Product-Backlog definierten Anforderungen des Produktes werden durch das Entwicklungsteam inkrementell – Schritt für Schritt – in ein bis vier Wochen langen Intervallen umgesetzt. Diese Intervalle werden **Sprints** genannt. Ein Sprint ist folglich ein Arbeitsabschnitt, in dem ein Teil einer Funktionalität des Produktes entwickelt wird. Im Zuge des Sprint-Planning werden die für einen Sprint ausgewählten Einträge des Product-Backlogs in einzelne Arbeitsschritte – hier **Tasks** genannt – heruntergebrochen. In diesem Planungsprozess wird die inkrementelle Vorgehensweise besonders gut sichtbar. Tasks sollten jeweils nicht mehr als einen Tag in Anspruch nehmen. Das Ergebnis dieser Planungsphase ist das **Sprint-Backlog**. Es definiert die während eines Sprints abzuarbeitenden Arbeitsschritte. Der Sprint startet mit einem Planungsschritt (**Sprint-Planning**) und endet mit einer kritischen Bestandsaufnahme des Erreichten (**Sprint-Retrospektive**; vgl. Abb. 4). Im Zuge des **Sprint-Planning-Prozesses** gilt es, zwei zentrale Fragen zu beantworten:

1. Was-Frage: Was soll im kommenden Sprint entwickelt werden, d.h. welche Ziele sind zu erreichen?

2. Wie-Frage: Wie soll die Arbeit im kommenden Sprint organisiert werden?

Dieser – häufig zweigeteilte Planungsprozess – soll nicht mehr als zwei Stunden für jede zu planende Sprint-Woche dauern. Ist ein Sprint auf vier Wochen ausgelegt, sollte die Planung folglich nicht mehr als acht Stunden dauern.

Zur **Beantwortung der Was-Frage** hat der Product-Owner die folgende Aufgabe: Er muss dem Entwicklungsteam die im Product-Backlog definierten und damit zu erreichenden Produkteigenschaften mit den zugewiesenen Prioritäten vorstellen. Dafür ist es wichtig, dass das Product-Backlog im zuvor abgeschlossenen Sprint im so genannten Product-Backlog-Refinement vorbereitet wurde (vgl. Abb. 4). Konkret bedeutet das, dass die priorisierten Produkteigenschaften dort geordnet und die Einträge für den nächsten Sprint abgeschätzt wurden. Basierend auf diesen Informationen entwickelt das gesamte Scrum-Team – also Product-Owner, Scrum-Master und Entwicklungsteam – zunächst ein gemeinsames Verständnis der zu bewältigenden Arbeit. Im Zuge dieses Prozesses werden die gewünschten Eigenschaften und die Akzeptanzkriterien für die angestrebte Lösung definiert. Zusätzlich müssen die Anforderungen bzw. Kriterien festgelegt werden, anhand derer am Ende des Sprints bewertet werden kann, ob die angestrebte Funktionalität tatsächlich erfüllt ist. Hier spricht man auch von der Definition-of-Done. Unter einer **Definition-of-Done** wird das gemeinsame Verständnis des gesamten Scrum-Teams verstanden, wann eine Arbeit als fertig zu bezeichnen ist. Dazu werden zu Beginn eines Sprints in der Definition-of-Done bestimmte Qualitätskriterien und weitere – meist nicht-funktionale – Anforderungen definiert. Die zu Beginn eines Projektes verfasste Definition-of-Done kann im Laufe des Prozesses verfeinert werden.

Es ist dabei das erklärte Ziel, während eines Sprints die **Fertigstellung eines auslieferbaren Produktes** zu erreichen. Dabei handelt es sich um ein Produktteil (auch **Product-Increment**; vgl. Abb. 4), das nicht nur ausreichend getestet, sondern auch so weit in die Gesamtlösung integriert ist, das es für den Benutzer freigegeben werden könnte. Im nächsten Schritt des Planungsprozesses schätzt das Entwicklungsteam die Anzahl der Product-Backlog-Einträge ab, die es im nächs-

ten Sprint abarbeiten kann. Die Verantwortung für diese Aufwandsabschätzung liegt beim Team selbst. In welcher Reihenfolge Einträge bearbeitet werden, wird dagegen vom Product-Owner definiert. In einer Zusammenführung dieser beiden Perspektiven definiert das Scrum-Team gemeinsam das **Sprint-Ziel**. Zur **Beantwortung der Wie-Frage** legt das Entwicklungsteam im nächsten Schritt fest, welche einzelnen Aufgaben (Tasks) notwendig sind, um das Sprint-Ziel zu erreichen und damit die in Aussicht gestellten Product-Backlog-Einträge liefern zu können. Die Verantwortung für diese Planung liegt beim Entwicklungsteam. Bei Bedarf kann in dieser Phase mit dem Product-Owner Rücksprache gehalten werden, um offene Fragen zu klären.

Das Ergebnis des Sprint-Planning-Prozesses ist das **Sprint-Backlog** (vgl. Abb. 4). Es stellt eine detaillierte Planung für den nächsten Sprint dar. Es umfasst die geplanten Product-Backlog-Einträge sowie die Aufgaben für deren Umsetzung. Zur Visualisierung wird häufig ein Taskboard verwendet. Dieses liefert einen Überblick über die aktuellen Aufgaben.

- **Daily-Scrum:** Fester Bestandteil von Scrum ist ein regelmäßiges Treffen des Entwicklungsteams zu Beginn eines jeden Arbeitstages eines laufenden Sprints. Dieses wird als **Daily-Scrum** oder auch als **Daily-Standup** bezeichnet (vgl. Abb. 4). Es sollte die Dauer von 15 Minuten nicht überschreiten. Product-Owner und Scrum-Master sind in diese Aktivität eher unterstützend eingebunden. Im Kern geht es bei dieser Aktivität um einen unmittelbaren Informationsaustausch – vor allem innerhalb des Entwicklungsteams. Beim Daily-Scrum geht es folglich primär um einen Überblick über den erreichten Stand der Arbeit. Problemlösungen selbst sollen im Daily-Scrum nicht erarbeitet werden.
- **Sprint-Review:** Am Ende eines jeden Sprints steht der Sprint-Review als wichtiger Teil des Sprints selbst (vgl. 4). Deren Dauer umfasst maximal eine Stunde pro Sprint-Woche. In dieser Aktivität überprüfen das gesamte Scrum-Team sowie die Stakeholder das erreichte Ergebnis (Product-Increment). Hierzu präsentiert das Entwicklungsteam seine erzielten Ergebnisse. Diese werden mit den definierten Sprint-Zielen abgeglichen. Basierend auf den dabei gewonnenen Erkenntnissen werden die nächsten Schritte diskutiert.
- **Sprint-Retrospektive:** Am Ende eines jeden Sprints steht die Sprint-Retrospektive. Sie sollte pro Sprint-Woche nicht mehr als 45 Minuten dauern. Bei einem 4-Wochen-Sprint wären für diese Aktivität ca. drei Stunden anzusetzen. Während dieser Zeit überprüft das gesamte Scrum-Team selbst seine eigenen Vorgehensweisen, um Schwachstellen zu erkennen und Optimierungen bereits im nächsten Scrum umzusetzen. Dabei wird das Entwicklungsteam vom Scrum-Master unterstützt, der idealerweise seine Erfahrungen aus einer Vielzahl ähnlicher Prozesse einbringen kann. Bei dieser **Analyse auf einer Meta-Ebene** werden die eigenen Vorgehensweisen offen und konstruktiv kritisch beleuchtet.
- **Product-Backlog-Refinement:** Das Product-Backlog-Refinement (teilweise auch als Backlog-Grooming bezeichnet; „Grooming“ steht dabei für „Pflege“) ist ein laufender Prozess. In diesem entwickeln der Product-Owner mit dem Entwicklungsteam das Product-Backlog kontinuierlich weiter. Dies stellt eine zentrale Funktion des Scrum-Konzeptes dar, denn beim Scrum-Prozess wird nicht über viele Monate auf Basis eines einmal verabschiedeten Lasten-/Pflichtenheftes agiert. Ganz im Gegenteil: Neue Erkenntnisse, gesammelte Erfahrungen sowie neue Anforderun-

gen von Kunden und Anwendern können während des gesamten Entwicklungsprozesses laufend integriert und der weiteren Arbeit zugrunde gelegt werden.

- **User-Story:** Unter User-Story versteht man eine spezifische Technik, um die Anforderungen an ein Produkt aus der Perspektive des Anwenders zu verfassen. Dazu wird im Allgemeinen die Alltagssprache des Anwenders genutzt. Diese User-Stories werden verwendet, um im Product-Backlog die Produkteigenschaften zu beschreiben, die sich ein Anwender wünscht. Solcher User-Stories werden häufig wie folgt formuliert: „Als Nutzer von X erwarte ich die Funktion Y bzw. die Eigenschaft Z, um damit den Nutzen A zu erreichen.“ Für eine App zur Coupon-Verwaltung könnte eine solche User-Story lauten: „Als *Payback*-App-Nutzer möchte ich durch einen Button alle relevanten Coupons, die heute noch gültig sind, aktivieren. Dabei sollte sich die Aktivierung nur auf die Coupons beziehen, deren Akzeptanzstellen ich heute – ggf. nach und nach – erreiche.“
- **Taskboard:** Mit dem Taskboard steht dem Scrum-Team ein wichtiges Instrument zur Visualisierung des Sprint-Backlogs zur Verfügung. In diesem ist dokumentiert, welche Product-Backlog-Einträge jeweils für einen Sprint ausgewählt wurden. Zusätzlich ist zu erkennen, welche Tasks dazu zu bearbeiten sind und welchen Bearbeitungsstand diese erreicht haben. Hierfür kommt häufig eine **Kanban-Tafel** zum Einsatz (vgl. Abb. 5).



Abb. 5 Scrum – Kanban-Tafel
(Quelle: Kreutzer 2018, S. 223)

Häufig besteht das Taskboard aus den in Abb. 5 zu sehenden vier Spalten. In der ersten Spalte werden die Anforderungen aus dem Product-Backlog definiert, die das Entwicklungsteam für den jeweiligen Sprint ausgewählt hat. Die weiteren Spalten enthalten – wie in Abb. 5 zu sehen – die zur Umsetzung erforderlichen Tasks in ihrem jeweiligen Bearbeitungsstand.

- **Planungspoker (Planning-Poker):** Eine große Bedeutung kommt bei Scrum den Methoden zu, um die Aufwände für die verschiedenen Tasks abzuschätzen. Dieser Prozess wird als **dynamisches Verfahren** ausgelegt. Hierfür gilt es zunächst, dass die Mitglieder des Entwicklungsteams eigenständig und unbeeinflusst erste Schätzungen vornehmen. Um diesen Prozess zu beschleunigen, kommt der so genannte **Planungspoker** zum Einsatz (vgl. vertiefend Kreutzer 2018, S. 224f.).

- **Impediment-Backlog:** Hiermit trägt der Scrum-Master für alle sichtbar die beobachteten Arbeitsbehinderungen zusammen. Das Impediment-Backlog umfasst häufig nicht nur die festgestellten Hindernissen, sondern kann bereits Aufgaben zu ihrer Lösung und den jeweils erreichten Stand der Abarbeitung aufzeigen.

Durch die konsequente Umsetzung der hier aufgezeigten Vorgehensweise kann in Projekten eine deutlich höhere Agilität erreicht werden.

5. Wie Sie Scrum im Zuge des Entrepreneurial Marketing nutzen können

Da beim erstmaligen Scrum-Einsatz viele interne Widerstände zu überwinden sind, bietet sich zunächst meist eine externe Unterstützung an. So kann in höherem Maße ein Erfolg der Methode sichergestellt werden. Stehen in Ihrem Unternehmen laufend umfassende Entwicklungsaufgaben an – und das ist in Start-ups ja meistens der Fall – so sollten Sie allerdings möglichst früh eine eigene Scrum-Kompetenz aufbauen.

Denken Sie dabei daran, dass gute Scrum-Teams zu ihrer Entwicklung Zeit benötigen, weil eine andere Art der Zusammenarbeit in Inhalt und Geist angestrebt wird. Eine unverzichtbare Erfolgsvoraussetzung für das Gelingen von Scrum ist außerdem ein umfassendes Empowerment des Scrum-Teams.

Scrum gehört heute zur Toolbox eines modernen Managements dazu. Jedes Unternehmen sollte sich mit den Grundlagen von Scrum beschäftigen und entsprechende interne Kompetenzen aufbauen. Wichtig ist, dass der erste Einsatz professionell begleitet wird, um ein Erfolgserlebnis zu ermöglichen.

Literatur

Beck, K./Beedle, M./Bennekum, A. van/Cockburn, A./Cunningham, W./Fowler, M./Grenning, J./Highsmith, J./Hunt, A./Jeffries, R./Kern, J./Marick, B./Martin, R.C./Mellor, S./Schaber, K./Sutherland, J./Thomas, D. (2018): Manifest für Agile Softwareentwicklung, <http://agilemanifesto.org/iso/de/manifesto.html>, Zugriff: 26.09.2018.

Kreutzer, R. (2018): Toolbox für Marketing und Management, Kreativkonzepte – Analysewerkzeuge – Prognoseinstrumente, Wiesbaden.

Preußig, J. (2015): Agiles Projektmanagement, Scrum, Use Cases, Task Boards & Co., Freiburg.

Röpstorff, S./Wiechmann, R. (2016): Scrum in der Praxis: Erfahrungen, Problemfelder und Erfolgsfaktoren, Heidelberg.

Sutherland, J. (2015): Scrum: The Art of Doing Twice the Work in Half the Time, New York.

Schlüsselwörter

Daily-Scrum, Scrum, Sprint, Scrum-Team, Product-Owner, Scrum-Master, Product-Backlog, Sprint, User-Story, Kanban-Tafel