

Agiles Projektmanagement mit Scrum

Herausforderungen bei der Produktivsetzung von Produktinkrementen

Peter Preuss, FOM Hochschule für Oekonomie & Management und Tobias Renk, BP Europa SE

Beim agilen Projektmanagement nach Scrum entwickeln Scrum-Teams in einer iterativen Vorgehensweise Software-Produkte. Ein wesentlicher Vorteil dieses Vorgehens ist, dass das gesamte Projekt flexibler gestaltet werden kann. So kann man beispielsweise sehr gut auf neue Anforderungen, die sich erst während des Projektverlaufs ergeben, reagieren. Ein weiterer Vorteil ist die Möglichkeit, in regelmäßigen Zeitabständen neue Versionen des Projektprodukts (sogenannte Produktinkremente) auszuliefern zu können anstatt bis zum „finalen GoLive“ am Projektende warten zu müssen. Ziel dieses Beitrags ist es, Herausforderungen zu diskutieren, die man bei der Produktivsetzung dieser Produktinkremente meistern muss.

Agile Project Management with Scrum – Challenges Involved in Delivering Product Increments

In Scrum projects self-organized teams develop software products in an iterative process. A major advantage of this approach is that the project becomes more flexible. For example, the project team can react very quickly if new requirements are discussed during the project. Another advantage is the possibility to deliver product increments on a regular base instead of having to wait until the end of the project. The aim of this article is to discuss challenges that must be overcome when these regular increments are moved into the production environment.

Keywords:
scrum, product increment

Grundsätzlich besteht das Scrum-Framework aus drei Projektrollen, drei Artefakten, fünf Ereignissen und einem fest vorgegebenen Regelwerk. Das Zusammenspiel dieser Scrum-Komponenten wird im Folgenden kurz beschrieben.

Scrum-Rollen: Jedes Scrum-Team hat einen Product Owner, einen Scrum Master und ein Entwicklungsteam. Die Verantwortung für das Software-Produkt liegt beim Product Owner. Er ist daher insbesondere verantwortlich für die Wertmaximierung und die Qualität des Produkts. Das Entwicklungsteam arbeitet selbstorganisiert und in iterativen Schleifen am Projektprodukt. Ein Entwicklungsteam sollte mindestens drei und nicht mehr als neun Mitglieder haben. Die Kernaufgabe des Scrum Masters besteht darin, im Team Verständnis für das Scrum-Framework zu schaffen, dafür zu sorgen, dass das Regelwerk eingehalten wird und alle Probleme beseitigen, die das Team bei seiner Arbeit behindern.

Scrum-Artefakte: Unter Artefakten versteht man Prozessergebnisse, die während eines Projekts erstellt werden. Scrum kennt die drei Artefakte Product Backlog, Sprint Backlog und Produktinkrement. Das Product Backlog ist

eine Liste mit den priorisierten Anforderungen an das zu erstellende Produkt. Diese Aufstellung ist niemals vollständig und existiert in der Regel so lange, wie das Projektprodukt entwickelt wird. Der Product Owner ist für das Product Backlog verantwortlich und überarbeitet kontinuierlich die Backlog-Anforderungen und deren Prioritäten. Das bedeutet, dass er die Einträge regelmäßig verfeinert, das Backlog um neue Anforderungen erweitert und obsolet gewordene Anforderungen aus der Aufstellung löscht. Je wichtiger eine Anforderung ist, desto genauer muss sie ausformuliert sein und desto größer muss deren Priorität sein. Beschrieben werden die Anforderungen häufig in Form von User Stories. Das Sprint Backlog enthält die Product-Backlog-Einträge, die in der aktuellen Projektiteration umgesetzt werden sollen. Diese Einträge werden auch als Sprint-Backlog-Items bezeichnet. Die Verantwortung für das Sprint Backlog liegt beim Entwicklerteam. Das bedeutet insbesondere, dass das Team entscheidet, welche Product-Backlog-Einträge es in der nächsten Iteration bearbeitet und wie diese in detaillierte Aufgaben (Tasks) verfeinert werden können. Während eines Sprints wird das Sprint Backlog nach der Erledigung eines Tasks vom Entwicklerteam aktualisiert. Das Er-



Prof. Dr. Peter Preuss lehrt Wirtschaftsinformatik an der FOM Hochschule für Oekonomie & Management in Stuttgart. Er ist geschäftsführender Gesellschafter der Unternehmensberatung People Consolidated GmbH, die sich auf die Einführung von SAP-Produkten für das Konzernrechnungswesen und -controlling spezialisiert hat.



Dr. Tobias Renk ist Global Service Owner für den Bereich B2C Pricing der BP Europa SE. Er ist als Experte und Keynote Speaker zu den Themen Innovation, kultureller Wandel und Digitale Transformation unterwegs. Außerdem ist er als Dozent für Unternehmensführung an der Dualen Hochschule in Stuttgart tätig.

peter.preuss@fom.de
www.fom.de

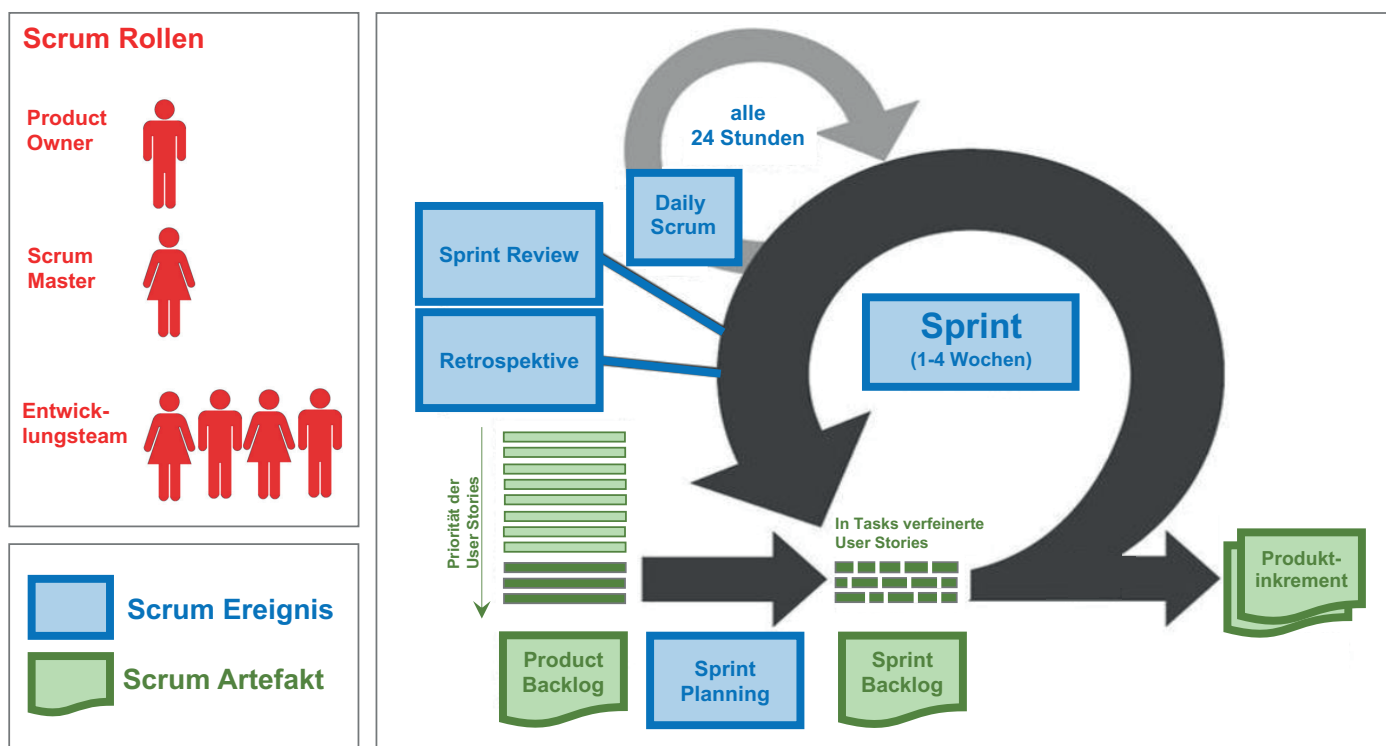


Bild 1: Scrum-Framework.

gebnis einer Iteration bezeichnet man als Produktinkrement. Dieses Artefakt besteht somit aus den in der aktuellen Iteration umgesetzten User-Stories und allen Anforderungen, die in den vorherigen Iterationen realisiert wurden. Wichtig ist, dass jedes Produktinkrement release-fähig ist, also potenziell an den Kunden auslieferbar ist.

Scrum-Ereignisse: Bei Scrum gibt es fünf Ereignisse mit festgelegter Dauer. Hierzu gehören der Sprint und vier Scrum-Meetings. Der Sprint ist ein vollständiger Iterationslauf immer gleicher Länge. Ist ein Sprint zu Ende, startet direkt im Anschluss die nächste Iteration. Während eines Sprints findet die Entwicklungsarbeit an den Sprint-Backlog-Items statt. Gestartet wird ein Sprint mit dem Sprint Planning Meeting. In dieser Besprechung fokussiert man sich auf folgende zwei Fragestellungen: Was kann in diesem Sprint realisiert werden, und wie wird diese Arbeit umgesetzt? Das Entwicklungsteam übernimmt im ersten Teil der Planungssitzung so viele User Stories aus dem Product Backlog in das Sprint Backlog, wie es im Sprint umsetzen kann. Hieraus abgeleitet formuliert dann das gesamte Scrum-Team das Ziel des Sprints. Im zweiten Teil überlegt das Entwicklungsteam, welche Tasks zum Erreichen ihres Sprintziels und zur Abarbeitung der ausgewählten Product-Backlog-Items notwendig sind. Während der Entwicklungsarbeit

findet täglich zur gleichen Zeit das Daily Scrum Meeting statt, in dem jeder Entwickler auf seine aktuelle Arbeit eingeht und den Fortschritt bis zum nächsten Daily Scrum prognostiziert. Am Ende eines jeden Sprints werden das Sprint Review Meeting und die Sprint-Retrospektive abgehalten. Zunächst wird im Sprint Review das entwickelte Produktinkrement vorgestellt. An diesem Meeting nehmen das Scrum-Team und insbesondere der Kunde teil, da man von ihm ein Feedback zum Produktinkrement benötigt. Dieses Feedback wird dann verwendet, um ggf. neue Backlog-Anforderungen zu identifizieren. In der abschließenden Sprint-Retrospektive reflektiert das Scrum-Team den Sprintverlauf und überlegt, was im nächsten Sprint verbessert werden könnte (Bild 1).

Produktivsetzung der Produktinkremente

Wie bereits erläutert, wird bei Scrum am Ende eines Sprints ein potenziell auslieferbares Produktinkrement erzeugt. Das bedeutet, dass der Kunde das Produktinkrement nutzen kann, und zwar in der Produktivumgebung. Es ist also nicht das Ziel von Scrum, sämtliche möglichen Graubereiche wie „eigentlich fertig“ oder „fast fertig“ oder „fertig, aber noch nicht final getestet“ zu erreichen. Viele Organisationen konnten diesen Schritt noch nicht in Gänze gehen. Insofern lässt sich durchaus zusammenfassen,

dass ein Großteil der heutigen Scrum-Organisationen in Wirklichkeit keine sind. Vor allem vier Aspekte, die nachgehend näher mit Erfahrungen aus der Praxis erläutert werden sollen, zeigen sich immer wieder als Hindernisse auf dem Weg zu einer echten Scrum-Organisation.

Aspekt 1: Die Wichtigkeit der Definition of Done

Ein wesentlicher Erfolgsfaktor von Scrum ist eine klare Festlegung der Definition of Done. Hierunter versteht man bestimmte Charakteristika, die eine User Story oder ein Task erfüllen muss, um am Ende eines Sprints wirklich als „done“, also als „fertig“ deklariert werden zu können. Die Definition of Done sollte in jedem Projekt individuell vom Scrum-Team festgelegt werden. Aufgrund der Definition des Produktinkrements gibt es aber ein paar Kriterien, die in jedem Scrum-Projekt erfüllt sein müssen. Definitor-of-Done-Kriterien für User Stories und Tasks könnten demnach so aussehen:

Done-Regel für User Story:

- 100 % aller Akzeptanzkriterien erfüllt
- User-Story-Tests erfolgreich abgeschlossen
- Integrationstest erfolgreich abgeschlossen
- Alle Bugs, die durch die User Story generiert wurden, wurden behoben
- Die entsprechenden Dokumentationen und Instruktionen, insbesondere die Endanwenderdokumentation, wurden fertiggestellt und übergeben

Done-Regel für Task:

- Dokumentation fertiggestellt und übergeben
- Alle Unit-Tests erfolgreich abgeschlossen
- Code Review erfolgreich durchgeführt

Die Wichtigkeit der Definition of Done darf nicht unterschätzt werden. Sie gibt am Ende dem Team die Sicherheit und Klarheit, die benötigt wird, um innerhalb eines Sprints tatsächlich ein potenziell auslieferbares Produktinkrement zu erzeugen.

Aspekt 2: Das Kreuz mit dem Testen

Im obigen Abschnitt wurde die Bedeutung des Testens sehr deutlich. Ohne erfolgreich durchgeführte Tests kann kein Code mit gutem Gewissen produktiv gesetzt werden. Hierbei sind vor allem zwei Aspekte besonders hervorzuheben.

In größeren IT-Projekten werden häufig viele Test- und Quality-Assurance-Personen eingesetzt. Dem ist prinzipiell nichts entgegenzu-

setzen. Allerdings können viele Testpersonen auch als kritisch betrachtet werden. Zum einen vergrößert sich dadurch die Teamgröße, was die Kommunikation innerhalb des Teams erschwert. Damit einher geht ein grundlegendes Prinzip agiler Entwicklungsmethoden, nämlich die Reduzierung von Übergaben (Hand-over). Zum anderen wirkt sich ein dediziertes Testteam ab einer gewissen Größe natürlich auch negativ auf das Projektbudget aus. Es gilt also, hier das richtige Maß zu finden. In der Praxis hat es sich als äußerst positiv herausgestellt, das Thema Testing in die Hände der Business-Analysten zu legen. Denn wer weiß besser über die richtige Umsetzung einer User Story Bescheid als die Business-Analysten, die die User Story geschrieben und die Akzeptanzkriterien festgelegt haben?

Der Königsweg ist freilich die Testautomatisierung, die letzten Endes unumgänglich ist, wenn man Scrum mit kurzen Entwicklungszyklen wirklich sinnvoll umsetzen will. Allerdings gibt es auch hierbei mannigfaltige Herausforderungen. Zum einen erzeugt eine Testautomatisierung zunächst hohe Kosten, ohne direkt einen Gegenwert zu liefern. Eine Organisation muss also Willens sein, dieses Investment zu tätigen, was sich üblicherweise erst nach ein bis drei Jahren auszahlt. Zum anderen muss man sich im Klaren darüber sein, was die Testautomatisierung umfassen soll. Sollen alle potenziellen Tests automatisiert werden? Mit welchen Tests fängt man bei der Automatisierung an? Kann die Automatisierung so in den Sprint integriert werden, dass sukzessive eine Testautomatisierung entsteht? In der Praxis hat sich Folgendes bewährt und als äußerst vorteilhaft herausgestellt. Wenn eine Organisation mit Testautomatisierung starten will, dann sollte man sich zunächst auf die Automatisierung der Regressionstests konzentrieren. Die Vorteile liegen auf der Hand. Regressionstests können über Nacht laufen und sparen so immens Zeit ein, um abzu prüfen, ob der neue Programmcode negative Auswirkungen auf bereits bestehenden Code hat. Als nächster Schritt ist ein Test-driven Development (TDD) erstrebenswert, was die Thematik Unit Testing stark verbessert. Das bedeutet im Prinzip nichts anderes, als dass ein Entwickler zunächst damit beginnt, ein Testprogramm zu schreiben, welches die Akzeptanzkriterien abprüft. Ist dieses erstellt, kann der Entwickler gegen dieses Programm das Produktinkrement entwickeln. Das spart am Ende auch enorm viel Zeit und zeigt einmal mehr die Wichtigkeit der Definition of Done. Eine große Herausforderung ergibt sich bei der Thematik Testautomatisierung vor allem für über Jahre

gewachsene Unternehmen mit einer durchaus komplexen IT-Landschaft.

Aspekt 3: Ein schweres Erbe:
gewachsene IT-Landschaften

Vielorts begegnet einem das Spotify-Modell als gutes Beispiel, wie Agilität in größeren Projekten umgesetzt werden kann. Kern dieses Modells ist es, möglichst autonome Teams (Squads) zu haben, die alle notwendigen Funktionalitäten und Anforderungen abdecken, um ein bestimmtes Teilprodukt unabhängig von anderen Squads weiterentwickeln zu können. In größeren Unternehmen, die über Jahrzehnte gewachsen sind, ist dieses Setup eine große Herausforderung. Im Gegensatz zu Start-Ups starten diese Unternehmen ihre agile Reise nicht auf der grünen Wiese (Green Field), sondern finden bereits eine gewachsene, und deshalb meist komplizierte IT-Infrastruktur vor (Brown Field). Diese sogenannte Legacy-Landschaft erweist sich vor allem hinsichtlich zweier Gesichtspunkte als schwierig auf dem Weg hin zu echter Agilität. Einerseits sind die Verantwortlichkeiten für verschiedene Systeme über mehrere Abteilungen verteilt, sodass eine echte Ende-zu-Ende-Betrachtung der Prozesse erschwert wird. Eine mögliche Lösung ist die Bildung bereichsübergreifender Squads. Oder anders ausgedrückt: Nimmt ein Unternehmen das Thema Agilität wirklich ernst, so wird es um eine Reorganisation nicht herumkommen. Andererseits wuchs die IT-Landschaft an unterschiedlichen Orten unterschiedlich schnell und damit sehr wahrscheinlich auch basierend auf unterschiedlichen Technologien. Diese gewachsene IT-Landschaft erfordert einen enormen Kraft- und Zeitaufwand für Unternehmen, um Scrum richtig umsetzen zu können.

Aspekt 4: Die Integration externer Lieferanten

Ein überaus wichtiger Aspekt, der oftmals nicht ausreichend betrachtet wird, ist die Integration von Lieferanten. Viele Unternehmen kooperieren mit externen Partnern, die ihrerseits selbst Strukturen und Prozesse haben, die von deren Mitarbeitern gelebt werden müssen. Dies kann am Ende eine Zusammenarbeit im Scrum-Team erheblich erschweren. Zwei Aspekte sind hier besonders zu beachten. Erstens sollte den externen Lieferanten ein Gefühl der Zugehörigkeit vermittelt werden. Denn nur so

kann ein echtes Scrum-Team heranwachsen. Beständigkeit der Teammitglieder ist dabei ein wesentliches Erfolgskriterium. Das Einfordern eines festen Lieferantenteams hat sich in der Praxis als sehr positiv gezeigt. Will heißen: Ein Wechseln der Teammitglieder aufseiten des Lieferanten sollte vermieden werden. Zweitens sollte Transparenz hinsichtlich der zu leistenden und geleisteten Arbeit erreicht werden. Bei der Zusammenarbeit mit externen Lieferanten ist das oft nicht so einfach möglich, da zum einen Informationen bewusst zurückgehalten werden können, zum Beispiel weil sie interne Angelegenheiten betreffen, und zum anderen verschiedene Tools für die Projektdurchführung eingesetzt werden. Es sollte daher immer ein konsolidiertes Backlog, in dem alle notwendigen Arbeitsbereiche offen und transparent enthalten sind, verwendet werden. Das ermöglicht es auch dem Kunden, jederzeit einen Überblick über den Projektfortschritt zu erhalten. Am Ende ist dies eine nicht zu unterschätzende Maßnahme, um Vertrauen innerhalb des Teams aufzubauen.

Resümee

In vielen Fällen wird agil als neues Allheilmittel in der Software-Entwicklung angesehen. Die zugrundeliegenden Agilen Prinzipien werden aber oft falsch interpretiert. Agil bedeutet in keinem Fall, dass man als Organisation einfach mal loslegt und guckt, wie das Projekt so läuft. Agilität, und im besonderen Scrum, beruht auf einem klar vorgegebenen Rahmenwerk und Regeln. Nur wenn dieses Rahmenwerk organisatorisch verankert wird und die Regeln eingehalten werden, bekommen agile Projekte die notwendige Struktur.

Ein wesentlicher Vorteil von Scrum besteht darin, dass man bereits während des Projektverlaufs regelmäßig Produktinkremente produktiv setzen kann. So erhält man wertvolle Kunden-Feedbacks und kann kontinuierlich prüfen, ob das Produkt auch die Kundenanforderungen erfüllt. In dem Beitrag wurde allerdings gezeigt, dass diese potenziellen Produktivsetzungen am Ende eines jeden Sprints in der Unternehmenspraxis mit größeren Herausforderungen verbunden sind.

Schlüsselwörter:
Scrum, Produktinkrement, Produktivsetzung

Literatur

- [1] Kniberg, H.; Invarsson: Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds. URL: <https://creativeheldstab.com/wp-content/uploads/2014/09/scaling-agile-spotify-11.pdf>, Abrufdatum 20.12.2019.
- [2] Preuss, P.: Scaling Scrum mit Nexus, in: Handbuch der Unternehmensberatung: Kennzahl 2155, Berlin 2019.
- [3] Schwaber, K.; Sutherland, K.: The Scrum Guide. 2017, URL: www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100, 2017, Abrufdatum 20.12.2019.