www.bt-magazin.de 4.2013 Heft 15 € 4,90 S&S

Business Technology

Business Technology

Lean Innovation & IT Leadership

New School of IT – IT muss neu gedacht werden

EAM Light – Pragmatik ist gefragt

Die agile Organisation – der Erfolgsfaktor

Alternativen zur Rolle des Product Owners oder: Wie bleiben wir agil?

Die Skalierung von Agilität in großen Organisationen

Scrum ist definiert für einzelne Teams und in dieser Größenordnung gut erprobt, umfangreich beschrieben und weit verbreitet. Bei einem Einsatz von Scrum in großen Unternehmen und bei der Entwicklung eines gemeinsamen Produkts durch eine große Anzahl von Teams werden wir in der Praxis aber zunächst allein gelassen: Literatur wie auch Erfahrungsberichte haben bislang keine Patentrezepte geliefert [1], [2], [3]. Die Realität zeigt jedoch große Herausforderungen. Ein zentrales Problem bei der Skalierung von Scrum ist die Skalierung der Rolle des Product Owners. Er wird schnell zum Bottleneck. Wie organisieren wir die Aufgaben der Product-Owner-Rolle in einer großen Organisation mit einer Vielzahl agiler Teams, die an einem gemeinsamen Produkt arbeiten? Dieser Artikel diskutiert Lösungsansätze und beschreibt eine favorisierte Organisationsstruktur im Detail.

AUTOR: PATRICK DAUT

Der Product Owner als Bottleneck? Um uns den tatsächlichen Herausforderungen bei der Arbeit mit Scrum in einer großen Organisation anzunähern, stellen wir uns die Arbeit in und die Zusammenarbeit zwischen einer großen Zahl von Scrum-Teams vor. In der Praxis ist in der Regel jedem Team ein eigener Product Owner zugeordnet. Steigt die Anzahl der Teams, die gemeinsam an einem Produkt arbeiten, entsteht Abstimmungsbedarf zwischen den Product Ownern. Hier liegt ein zentrales Problem bei der Skalierung von Scrum: Die Rolle des Product Owners und damit die Anforderungsseite beziehungsweise die Beziehung zum Kunden skaliert nicht, der Product Owner ist ein Bottleneck.

DER PRODUCT OWNER IN GROSSEN ORGANISATIONEN

Wie kommt es zu diesem zentralen Engpass? Gehen wir davon aus, dass in einer großen Organisation eine Abteilung typischerweise größer als ein ideales Scrum-Team ist, so müssen kleinere Teams gebildet werden. Das heißt also, mit der Einführung von Scrum werden Teams und Abteilungen in kleinere Teams zerlegt, um deren jeweilige Größe zu begrenzen. Dadurch entsteht eine – theoretisch beliebig hohe – Anzahl von Teams, die meist höher ist als die Anzahl der Abteilungen. Gehen wir weiter davon aus, dass diese Teams an einem einzigen gemeinsamen Produkt arbeiten. Die Herausforderung ist hier eine organisatorische: Da Scrum selbst den Fall mehrerer Teams nicht betrachtet, ist zunächst der Umgang damit nicht definiert. Naheliegend ist die Möglichkeit, jedem Team einen dedizierten Product Owner zuzuordnen. Doch wie stellen wir dann die Konsistenz des Gesamtprodukts sicher? Es besteht ein Koordinationsbedarf zwischen den Product Ownern. Um diesen zu decken, muss eine Organisationsstruktur geschaffen werden, die im Scrum-Framework so zunächst nicht beschrieben ist.

Betrachten wir andererseits die Möglichkeit, allen Teams nur einen einzigen gemeinsamen Product Owner zur Verfügung zu stellen, so besteht das beschriebene Problem nicht: Es gibt eine einzige Person mit der inhaltlichen Verantwortung für das Produkt. Allerdings muss der Product Owner dann mit steigender Anzahl von Teams zum Flaschenhals werden: Neben der Priorisierung von Anforderungen beinhaltet die Product-Owner-Rolle weiterhin die Aufgabe, während der Detaillierung von Anforderungen bzw. während deren Umsetzung dem Team als zentraler Ansprechpartner und ggf. als zentraler Kanal zum Kunden zur Verfügung zu stehen. Die ständige Verfügbarkeit des Product Owners in seiner Funktion als „On-site Customer“ ist für den Erfolg agilen Vorgehens zentral, da Kommunikation nach Bedarf einer detaillierten Spezifikation im Vorfeld vorgezogen wird [4], [5]. Mit steigender Anzahl von Teams wird der Product Owner überlastet und bremst die Arbeit der Teams aus. Darüber hinaus kommen mit steigender Organisations- und Produktgröße weitere Stakeholder, darunter verschiedene Managementpositionen, unternehmensweite Querschnittsfunktionen wie Security und Compliance oder Investoren, weitere Kunden etc. hinzu. Für deren Betreuung sieht Scrum „nur“ die Rolle des Product Owners vor. **Abbildung 1** zeigt die Fülle an Schnittstellen zu Stakeholdern, deren Koordination der Product Owner regeln muss.

Da Scrum den Fall mehrerer Teams nicht betrachtet, hat nach der originären Definition ein Scrum-Team einen dedizierten Product Owner. Ebenso ist explizit vorgesehen, dass die Rolle des Product Owners durch eine Person ausgefüllt wird, nicht durch ein Komitee oder eine Gruppe. Hier liegt das Dilemma bei der Ska-

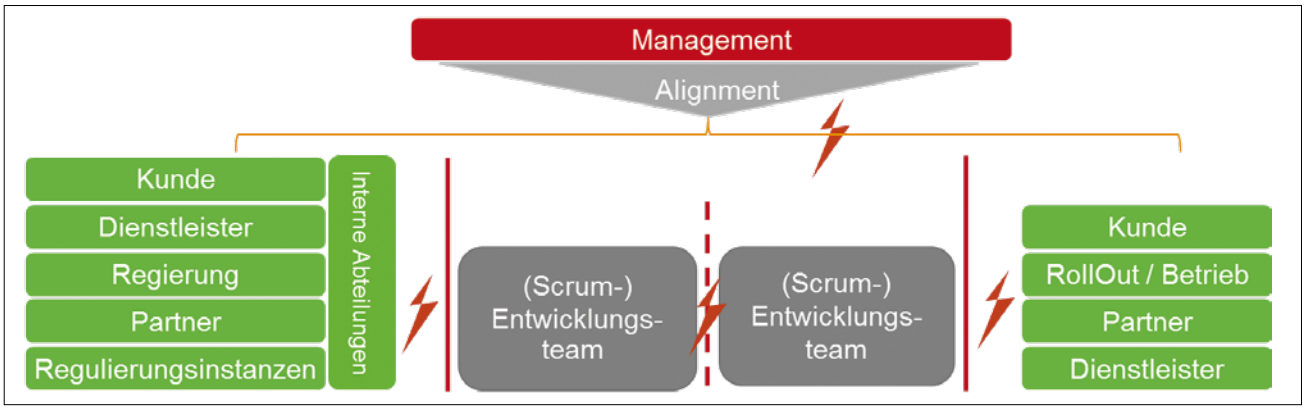

Abb. 1: Relevante Stakeholder und Schnittstellen zu ihnen

Abb. 2: Funktionale Aufteilung der Product-Owner-Rolle

lierung der Product-Owner-Rolle, denn es gibt einen guten Grund dafür, dass sie nur durch eine Person ausgefüllt sein sollte: Der Product Owner soll inhaltlich für das Produkt verantwortlich sein und verantwortlich gemacht werden. Der Product Owner darf als einziger Anforderungen priorisieren und übernimmt damit die Verantwortung für die Ausgestaltung des Produkts zu einem bestimmten Zeitpunkt. Stakeholder nehmen darauf Einfluss, aber nur indirekt. Durch diese Singularität des Product Owners entsteht der eingangs beschriebene Engpass.

Eine Überlastung des Product Owners führt dazu, dass Entwicklungsteams bei Design und Implementierung auf den Product Owner warten müssen. Dadurch sinkt der Durchsatz im Entwicklungsprozess. Außerdem werden häufige Fokuswechsel bei den Entwicklern erforderlich, um entstehende Wartezeit mit anderen Tätigkeiten zu füllen. Entsprechend der Lean-Idee handelt es sich hier um „Waste“: Aufwand und Zeit, die nicht zur Schaffung von Wert beitragen [6], [7].

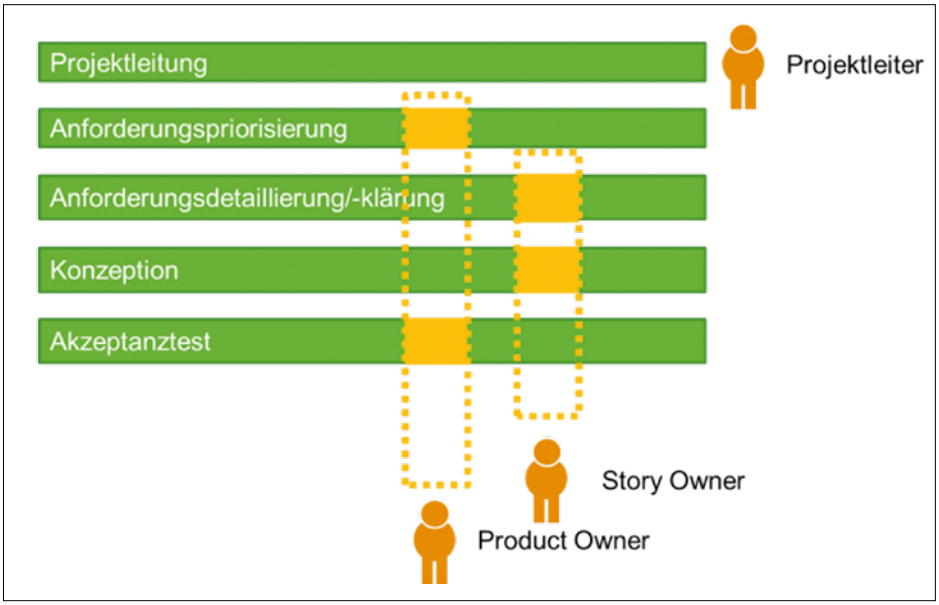

Abb. 3: Vertikaler Schnitt in Rollen

MODULARISIERUNG DER AUFGABEN DES PRODUCT OWNERS

Um das beschriebene Problem zu lösen, sind verschiedene Ansätze denkbar. Grundsätzlich ist eine Modularisierung erstrebenswert. Bearbeitet jeweils ein Team einen Teil eines Produkts, wobei diese Teile möglichst unabhängig voneinander sind, sinken auch die Abhängigkeiten zwischen den Teams und damit der Koordinationsaufwand. Eine gewisse Spezialisierung erhöht die Effizienz und steigert das Detailwissen in der betroffenen Domäne. Eine vollständige Unabhängigkeit wird sich aber in der Praxis nicht erreichen lassen; falls doch, handelt es sich nicht mehr um eines, sondern um mehrere Produkte, und das hier im Fokus stehende Problem besteht per Definition nicht. Außerdem bringt die Spezialisierung eines Teams auf ein Modul oder ein Teilprodukt einen Nachteil mit: Ein größeres Feature mit dem Schwerpunkt des Aufwands auf einem Modul kann nicht ohne Weiteres zur schnellen Umsetzung auf mehrere Teams verteilt werden. Bei der Erreichung der Ziele der kurzfristigen Amortisierung und des frühzeitigen Erhaltens von Feedback kann ein Team dann zum Engpass werden. Ein „Feature“ ist hier eine Gruppe von User Stories, deren Nutzen gemeinsam höher ist als die Summe des Nutzens der einzelnen Stories. Es handelt sich um eine übergeordnete Ebene, in der betriebswirtschaftliche Betrachtungen, beispielsweise des ROIs, im Fokus stehen.

Gehen wir deshalb also von Abhängigkeiten zwischen der Arbeit der verschiedenen Teams aus. Eine Möglichkeit der Skalierung besteht darin, jedem Team einen dedizierten Product Owner zuzuordnen und den Abstimmungsbedarf zwischen ihnen durch eine hierarchische Struktur sicherzustellen. Diese beinhaltet einen übergeordneten Chief Product Owner.

Können sich die Teams auf einen relativ abgeschlossenen Bereich konzentrieren und ist die Anzahl der Teams klein, so reicht eine flache Struktur

Die Story Owner können Entscheidungen auf Story-Ebene eigenverantwortlich treffen.

mit einem Chief Product Owner und den Teams mit ihren Product Ownern darunter aus. Bei größeren Abhängigkeiten zwischen den Teilprodukten bzw. den Teams oder spätestens ab einer Teamanzahl größer neun wird wiederum der Chief Product Owner überlastet. Eine mehrstufige Hierarchie entsteht. Bei weiterer Skalierung müssen weitere Ebenen mit Product Ownern hinzukommen. Der Kommunikationsbedarf steigt.

Betrachten wir alternativ die Möglichkeit, die Rolle des Product Owners anhand seiner Aufgaben in mehrere Rollen aufzuteilen. Grob umrissen übernimmt der Product Owner die folgenden Aufgaben:

- Priorisierung der Anforderungen
- Unterstützung bei der Detaillierung von Anforderungen, Repräsentation des Kunden, Klärung von Detailfragen
- Unterstützung bei Konzeption und Design
- Durchführung der Akzeptanztests
- Stakeholder-Management

Scrum führt diese Aufgaben in einer Rolle zusammen. Bei klassischen Vorgehensweisen hingegen übernimmt ein Businessanalyst die Priorisierung und Detaillierung der Anforderungen, ein Architekt das Design und die Konzeption sowie der Kunde oder Businessanalyst die Abnahme. Ein Projektleiter übernimmt das Management der Stakeholder. Trennt man die Aufgaben entsprechend klassischer Vorgehensweisen rein funktional in diese Rollen über alle Anforderungen hinweg, wie in **Abbildung 2** dargestellt, kann aufgrund der funktionalen Spezialisierung leicht skaliert werden. Allerdings gehen die Vorteile des agilen Vorgehens verloren: Es entsteht beispielsweise ein Team von Businessanalysten, die ein vorgelagert klassisches Requirements Engineering betreiben und Konzepte lange vor deren Umsetzung generieren. Es wird „Big Upfront Design“ betrieben, dessen Wert immer stärker verfällt, während es auf die Umsetzung wartet. Reinertsen beispielsweise spricht von den durch Wertverfall entstehenden Kosten als „Holding Costs“ und nennt sie als einen Parameter für die Bestimmung der optimalen „Batch Size“ [8, S. 35 f.].

Wie kann ein Mittelweg aussehen, der einerseits die Agilität erhält, andererseits aber Skalierung erlaubt? Ein Vorschlag ist, die globalen Aufgaben von denen zu trennen, die sich auf eine Anforderung zentrieren lassen. Der Product Owner ist weiter verantwortlich für die Anforderungspriorisierung und bestimmt damit über das Produkt. Die Detailabstimmung mit dem Kunden zu einer Story, deren Ausarbeitung zusammen mit dem Entwicklungsteam und die Funktion der Schnittstelle zum Kunden bei Rückfragen auf Story-Ebene wird davon getrennt und in einer zweiten Rolle, dem „Story Owner“, zusammengefasst. **Abbildung 3** zeigt den entstehenden Schnitt durch die Funktionen.

Die Organisationsstruktur sieht also folgende Rollen vor: einen Product Owner, mehrere Teams sowie pro Story einen Story Owner, der bei Abstimmungen zu einer Story, Konzeption etc. mitwirkt (**Abb. 4**).

STORY OWNER: EINE NEUE ROLLE MACHT DIE STRUKTUR SKALIERBAR

Die letztgenannte Option – die Kombination aus Product Owner und Story Owner – ist insgesamt vorteil-

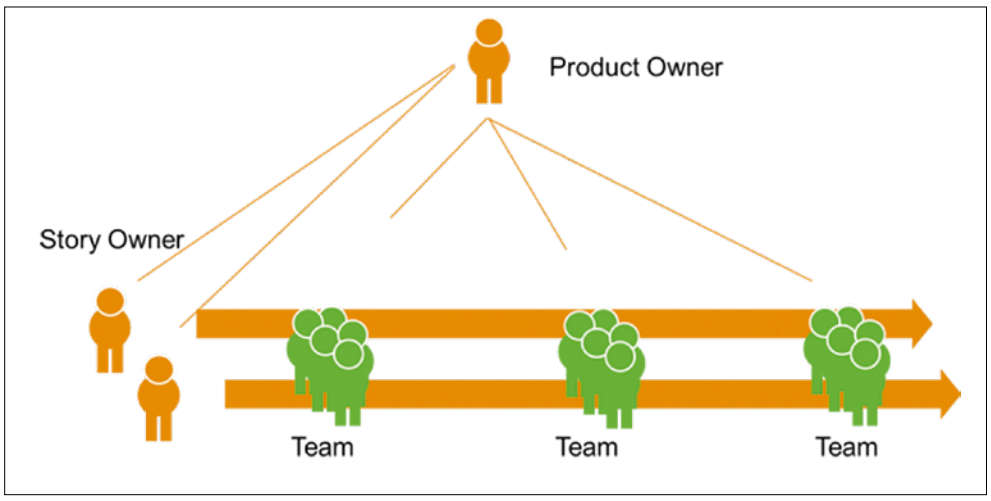

Abb. 4: Story-Owner-Struktur

hafter. Durch die Verteilung auf einen Product Owner und mehrere Story Owner hat man weiter eine einzige Rolle und eine einzige Person, die für die Priorisierung zuständig und damit für die Wertmaximierung des Produkts verantwortlich ist. Sie übernimmt letztlich die Verantwortung und kann sie weder abgeben noch kann sie in einer Gruppe „verwaschen werden". Aufwändige Einigungsprozesse finden nicht statt. Die genauere Ausgestaltung von Anforderungen kann dagegen besser ausgelagert und auf mehrere Köpfe verteilt werden. Letztlich werden viele Entscheidungen die Hierarchie aufwärts delegiert und führen dort zu einer Überlastung. Die Folge sind Wartezeiten, die wiederum die Durchlaufzeit von Anforderungen durch den Prozess erhöhen.

Bei einer Lösung mit unabhängigen Story Ownern hingegen greifen Dezentralisierung und Delegation von Entscheidungen soweit abwärts wie möglich in eine Hierarchie ein, um Skalierbarkeit zu erreichen. Hierbei handelt es sich um die Verfolgung eines Aspekts der „Lean"-Idee: Empowerment der Basis und mehr Verantwortung soweit entlang der Hierarchie herunter zu delegieren, bis genug Wissen für eine Entscheidung vorhanden ist [9], [10]. Eine grundsätzliche Empfehlung ist, Entscheidungen, die in unregelmäßigen Abständen zu treffen sind, große Auswirkungen nach sich ziehen oder signifikante Economies of Scale haben, zu zentralisieren und dagegen schnell zu treffende Entscheidungen, deren Effekte einem Wertverfall im Laufe der Zeit unterliegen, zu dezentralisieren. Diesem Prinzip wird hier gefolgt: Die Story Owner können Entscheidungen auf Story-Ebene eigenverantwortlich treffen. So entfällt Wartezeit sowohl auf Entscheidungen als auch auf Abstimmungsleistungen eines überlasteten Product Owners im Falle von Entscheidungszentralisierung [8, S. 246 ff.]. Damit wird „Waste" eliminiert. Der Durchsatz im Gesamtsystem steigt und die Durchlaufzeit eines einzelnen Features durch den Entwicklungsprozess sinkt (**Abb. 5**).

Nun kann argumentiert werden, dass ein dedizierter Product Owner pro Team von Vorteil in der

Rolle	Verantwortung	Aufgaben	Stakeholder
Entwicklungsteam	Verantwortlich für das Erreichen des selbstgesetzten Sprintziels sowie für die Definition eines Wegs dazu.	• Definiton des Sprintziels • Bestimmung des realisierbaren Arbeitsumfangs • Design und Implementierung der durch User Stories repräsentierten Anforderungen	• Story Owner • Weitere Entwicklungsteams • Product Owner
Story Owner	Verantwortung für Umsetzung der ihm zugewiesenen Stories, Epics und Themen entsprechend der vom Product Owner vergebenen Priorität. Verantwortung für die Detailabstimmung mit Kunden und inhaltliche Ausgestaltung einer Story.	• Unterstützung bei der Detaillierung von Anforderungen • Unterstützung bei Konzeption und Design • Durchführung der Akzeptanztests • Vorstellung der jeweiligen Stories im Sprint Planning • Je nach Thema auch das Finden eines adäquaten Modus für die Abstimmung mit involvierten Teams	• Product Owner • Kunde • Entwicklungsteams
Product Owner	Verantwortlich für die Wertmaximierung des Produkts.	• Priorisierung der Items im Product Backlog • Abstimmung mit Story Ownern • Akzeptanztest des integrierten Produkts	• Kunde • Story Owner • Entwicklungsteams
Scrum Master	Verantwortlich dafür, dass Scrum verstanden und umgesetzt wird.	• Ein Scrum Master pro Team • Organisation und Durchführung des Sprint Plannings	• Product Owner • Story Owner • Entwicklungsteam • Organisation
Projektleiter	Verantwortlich für Stakeholder-Management sowie dafür, Visibilität und Interessenvertretung in der Gesamtorganisation zu erzeugen. Seine Arbeit ist als „Service" für Product Owner und Teams zu verstehen.	• Information der Stakeholder • Feedback und Anregungen von Stakeholdern mitnehmen • Den Product Owner bei konkreten Anforderungen hinzuziehen • Den Product Owner von Politik entlasten	• Kunde • Management • Diverse weitere interne und exteren Stakeholder • Querschnittsfunktionen (Security ...)
Scrum-Team	Bestehend aus Entwicklungsteam und Scrum Master. Die Story Owner sind nicht Teil eines Teams, da die von ihnen betreuten Anforderungen sich auf mehrere Teams verteilen können. Ziel ist es, ein Feature möglichst schnell zu einem nutzbringenden Ergebnis zu führen und dazu ggf. auch Stories eines Features über mehrere Teams zu verteilen.		

Tabelle 1: Definition der beteiligten Rollen [8]

Zusammenarbeit zwischen Team und Product Owner ist. Explizit begründet werden kann dies aber nicht. Hier hingegen wird die These aufgestellt, dass ein Team leichter mit einem Story Owner pro Story kommunizieren kann und es vorteilhafter ist, dass zwischen den Story Ownern durch ihren Fokus auf jeweils eine Story relativ wenig Abstimmungsbedarf besteht. Diese Konstellation macht eine bessere Skalierung möglich. Somit folgt die Form der Funktion. Die in Scrum vorgesehenen Meetings, zusammen mit parallel zu den Sprints durchgeführten Groomings, dienen der Identifikation von Abhängigkeiten und Berührungspunkten zwischen den Stories. Die tiefere Analyse dieser Abhängigkeiten und deren Behandlung erfolgt dann bilateral zwischen den Story Ownern. Dadurch wird der Product Owner stark entlastet.

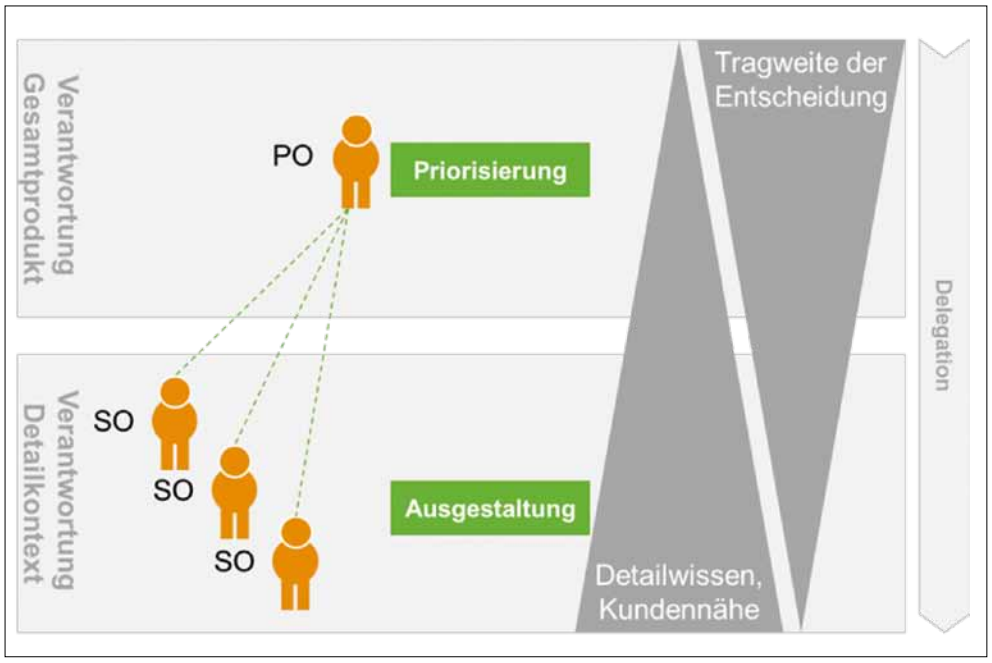

Abb. 5: Verteilung der Verantwortung auf Product Owner und Story Owner

Zur weiteren Entlastung kann in dem hier beschriebenen Modell neben den Product-Owner- und Story-Owner-Rollen bei Bedarf ein Projektleiter eingeführt werden. Gemeint ist damit allerdings nicht ein Projektmanager im klassischen Sinne. Vielmehr handelt es sich um eine, den Product Owner hinsichtlich des Stakeholder-Managements und politischer Belange entlastende Rolle ohne Verantwortung und Entscheidungskompetenz für das Produkt. Tabelle 1 fasst die Rollen des hier beschriebenen Modells und deren Definition zusammen.

Der Prozess, wie auch generierte Artefakte, bleiben wie für Scrum definiert bestehen. Die Mitwirkung in den Meetings verändert sich wie in den Rollenbeschreibungen (Tabelle 1) ausgeführt. Ausnahmen dazu bilden das Sprint-Planning-Meeting und Groomings sowie der Sprint Review: In diesen Regelmeetings werden die Story Owner benötigt. Im Sinne einer erstrebenswerten Synchronisierung aller Teams können diese Meetings für alle Teams zum gleichen Zeitpunkt stattfinden [8, S. 186 ff.], [3, S. 305 ff.]. Nun werden die Stories meist so auf die Story Owner verteilt sein, dass deren Kompetenz in mehreren Teams gefragt ist. Eine Lösung dafür ist, die Meetings mit leichtem zeitlichem Versatz stattfinden zu lassen. Alternativ stellen Story Owner zu Beginn des Sprint-Planning-Meetings ihre Stories vor und stehen im Anschluss für eine definierte Zeitspanne allen Teams für Fragen zur Verfügung. Zum Ende des Plannings besuchen die Story Owner wieder die Räume der Teams, die an ihren jeweiligen Stories arbeiten werden. Eine Möglichkeit, Sprint Reviews in dieser Struktur durchzuführen ist, die Reviewmeetings der Teams mit thematischen Berührungspunkten in einem konkreten Sprint zusammenzulegen. Empfehlenswert ist weiterhin in größeren, aber regelmäßigen Abständen ein gemeinsames Meeting, in dem allen Teams vom Product Owner die Vision vermittelt wird. Hier können zudem in einer praktischen Demonstration Ergebnisse bezogen auf das gesamte Produkt und dessen Wertentwicklung präsentiert werden [1]. Hierzu kann auch das Management eingeladen werden.

Entscheidend bei der Umsetzung ist die Implementierung der Rolle des Story Owners: Er „repräsentiert“ eine bzw. mehrere Stories eines Features und ist in diesem Kontext zentraler Ansprechpartner – sowohl für die Teams, als auch für den Kunden. Dabei bildet er das koordinierende Bindeglied zwischen diesen Teams und stimmt sich andererseits mit den anderen Kollegen in seiner Rolle über Schnittmengen ab. Eine Konkurrenz der Story Owner um die Kapazität der Teams entsteht nicht, da der Product Owner weiterhin als einzige Instanz über die Priorisierung im Product Backlog entscheidet. Weiterhin gilt es im Auge zu behalten, dass die hier

vorgeschlagene Trennung von Product und Story Owner dazu dient, bei der Skalierung entstehenden zusätzlichen Koordinationsbedarf zu decken. Sie soll keine direkte Kommunikation – wo auch immer möglich und sinnvoll – unterbinden.

AGIL BLEIBEN – TROTZ SKALIERUNG

Die beschriebene Struktur eignet sich nach den Erfahrungen gut für eine Skalierung agilen Vorgehens in mittlerer Größenordnung von einigen wenigen bis zu mehreren Dutzend Teams. Zum einen wird ein Engpass durch hohe Arbeitsbelastung beim Product Owner bei steigender Anzahl von Teams aufgelöst, zum anderen bleibt die Struktur schlank und agil. Für jedes Feature und jede Story existiert mit dem Story Owner eine Person mit fachlichem Verständnis des Themas sowie der ungeteilten Verantwortung dafür, das Thema voranzubringen und mit den Teams zusammen im Detail auszugestalten. Mit der Anzahl der konkreten Stellen, die die Story-Owner-Rolle ausfüllen, begrenzt man den Work in Progress in der Struktur. Weitere Skalierung erfolgt durch das Hinzufügen weiterer Story Owner. Sie wird nicht mehr durch den Product Owner begrenzt; der Flaschenhals ist aufgelöst.

Links & Literatur

[1] http://scaledagileframework.com/, Zugriff: 11.10.2013
[2] Leffingwell, Dean: „Software Agility – Best Practices for Large Enterprises“, Addison-Wesley, Pearson Education, 2007
[3] Leffingwell, Dean: „Agile Software Requirements – Lean Requirements Practices for Teams, Programs, and the Enterprise“, Addison-Wesley, Pearson Education, 2011
[4] http://www.extremeprogramming.org/rules/customer.html, Zugriff: 11.10.2013
[5] Boehm, Barry W.; Turner, Richard: „Balancing Agility and Discipline: A Guide for the Perplexed“, Addison-Wesley, 2004, S. 44 f.
[6] Ohno, Taiichi: „Toyota Production System: Beyond Large-scale Production“, Productivity Press, 1988
[7] Poppendieck, Mary B.; Poppendieck, Tom: „Implementing Lean Software Development: From Concept to Cash“, Kapitel 2.2, Addison-Wesley, 2007
[8] Reinertsen, Donald G.: „The Principles of Product Development Flow – Second Generation Lean Product Development“, Celeritas Publishing, 2009, z. B. S. 32 ff.
[9] http://de.wikipedia.org/wiki/Lean_Management, Zugriff: 11.10.2013
[10] http://de.wikipedia.org/wiki/Lean_Production, Zugriff: 11.10.2013
[11] http://scrum.org/Portals/0/Documents/Scrum Guides/2013/ScrumGuide.pdf, Zugriff: 11.10.2013

Patrick Daut

ist Consultant mit dem Schwerpunkt E-Business bei der Cassini Consulting. Er verfügt über Erfahrungen aus zahlreichen Projekten für Konzerne und auch mittelständische Unternehmen. Zurzeit beschäftigt er sich insbesondere mit der Optimierung von Prozessen in der Softwareentwicklung sowie agilen Vorgehensweisen. Patrick Daut ist Dipl.-Wirtschaftsinformatiker. Seit 2011 arbeitet er als Berater bei der Cassini Consulting am Standort Hamburg.