

Klassische und agile Projektrollen im agilen Umfeld

Sorgen Sie für Durchblick im agilen Rollendschungel!



Christian Czerwonka

Senior Consultant bei der
adesso SE

Management Summary

- In der agilen Projektmanagement-Praxis zeigt sich, dass es oft mehr als nur die Rollen Scrum Master, Product Owner und Entwickler gibt. Zunehmend sind auch klassische Rollen in agilen Projekten vertreten.
- Klassische Rollen, wie z.B. Business Analysten oder Software-Architekten, sind für agile Projekte grundsätzlich auch sinnvoll.
- Hinsichtlich der Rollen ist die klare Abgrenzung der Befugnisse und Kompetenzen entscheidend, oder die Unterscheidung nach formal steuernden Rollen (z.B. Scrum Master, Projektleiter) und fachlich ausgerichteten Rollen (z.B. fachliche Tester, Product Owner).
- Weitere fachliche Rollen (ohne Führungscharakter) in agilen Projekten, können z.B. Software-Architekten und Business Analysten sein. Der Software-Architekt kann mit seinem technischen Fokus z.B. zwischen Entwicklern, Product Owner, Managern und anderen Interessengruppen vermitteln.
- Bei hoher Rollenvielfalt, u.a. in Scrum-Projekten, kommt es nicht selten zu negativen Folgen, z.B. der für die Wertmaximierung zuständige Product Owner hat keinen Einfluss auf die Team-Performance (d.h. auf die Aufgabe des Scrum Masters).
- Es empfiehlt sich deshalb u.a. die Erstellung und Abstimmung einer RACI-Matrix oder die Einführung klarer Eskalationsmechanismen bei schwerwiegenden Problemen.
- Mit zusätzlichen (klassischen) Rollen in agilen Projekten, v.a. Scrum, haben Sie eine Stellschraube zu mehr Effizienz im Projekt. Vor allem, wenn Sie dabei auf eine gute Rollenklärung i.S.d. Verantwortlichkeiten, Befugnisse und Kompetenzen achten.

Projektleiter im agilen Umfeld: Ist nicht schon die Frage danach völlig verkehrt?

Projekte werden heutzutage (und immer noch zunehmend) "agil" durchgeführt. Trotz dieser "agilen" Maxime stellt man in der Praxis jedoch fest: In der Regel gibt es in agilen Projekten oft noch die Rolle eines (Gesamt-)Projektleiters. Dieser wird vom Kunden/ Projektsponsor – wie selbstverständlich – immer wieder für Projekte eingefordert. Als solcher ist der Projektleiter, wie im klassischen "Wasserfall", verantwortlich für das

Projektbudget und für dessen Einhaltung. Auch "Meilensteine" (eigentlich eines der agilen Unwörter überhaupt) sind weiterhin verbreitet (auch wenn sie sich begrifflich als "Inkrement", "Epic" oder Ähnliches verstecken). Dies alles mag vor dem Hintergrund der populären und kaum infrage gestellten agilen Paradigmen überraschen.

Aber es sind auch die eigentlichen Scrum-Rollen vertreten: der Scrum Master, der Product Owner und die Entwickler oder "Developers", wie sie im Scrum Guide bezeichnet werden. Was konkret unter Developers zu verstehen ist, lohnt einer Betrachtung. Schließlich weisen die Autoren des Scrum Guides darauf hin, dass ihr Framework auch bei anderen Arten von Projekten (also nicht nur in Software-Entwicklungsprojekten) angewendet wird. Rein logisch muss insofern der Schluss gezogen werden, dass "Developers" auch andere fachliche Domänen, wie z.B. Business Analysts (auch als Requirements Engineers bezeichnet), Architekten oder fachliche und technische Tester umfasst. Die Notwendigkeit zur Besetzung der letztgenannten Bereiche in Projekten wird heutzutage kaum bestritten.

Wie sind Rollen grundsätzlich zu unterscheiden?

In der Betrachtung von Projektrollen hat es Sinn, zwischen formal steuernden und fachlich ausgerichteten Funktionen zu unterscheiden:

Formal steuernde Rollen:

- Der (klassische) Projektleiter, der Zeit, Kosten und Qualität im Blick hat
- Der Scrum Master, der die Steigerung der Teamperformance und das Coaching in Bezug auf Scrum-Prinzipien als Aufgabe hat

Fachlich ausgerichtete Rollen:

- Der Product Owner
- Fachliche Tester
- Alle Arten von Entwicklern (Developers) im oben beschriebenen Sinne

Es fragt sich, wenn wir nun all diese Rollen in einem Projekt haben: Wie passen diese alle zusammen? Oder, anders gefragt: Besteht nicht das Risiko, dass sich bestimmte Funktionen im Projekt in die Quere kommen? Und schließlich: Wer kann bzw. soll verantwortlich gemacht werden, wenn der Projektfortschritt und die Erfüllung des fachlichen Scopes (oder erreichten Business Values) den Kosten und der Zeit hinterherlaufen? In einem solchen Projekt ist es sinnvoll, folgende Fragen zu stellen:

- **Frage an den Scrum Master:** Erwerben die agilen Teams zu wenig Maturity, d.h. Grundverständnis über die Scrum-Methoden, und entwickelt sich die Velocity (d.h. Geschwindigkeit, mit der ein Team seine Ziele erreicht) in die falsche Richtung?
- **Frage an den Product Owner:** Werden die Backlog Items nicht adäquat priorisiert oder die Storys nicht hinreichend präzise formuliert oder sinnvoll heruntergebrochen?

- **Frage an den Projektleiter:** Hat er seine Metriken zur Steuerung des Gesamtprojekts im Griff? Hat er die Steuerungsmittel (auch in Abgrenzung zum Product Owner und Scrum Master), um das Projektziel zum Erfolg führen zu können? Hat er einen geeigneten Modus gefunden (auch in Abgrenzung zum Product Owner und Scrum Master), in dem er im Sinne der Steuerung wertstiftenden Einfluss auf die Erreichung der Projektziele ausüben kann?

Projektleiter, Scrum Master, Product Owner und Entwickler: Darauf beschränkt es sich allerdings **nicht** in der heutigen Projektwelt. Die Vielseitigkeit der Aufgaben (und damit der Rollen) werden im Folgenden näher erläutert.

Blicken wir einmal auf heute im Projektalltag existierende Rollen mit Ihren Aufgaben.

Welche (traditionellen) Rollen gibt es in agilen Projekten?

Ob klassisch, agil oder hybrid: Wichtige in heutigen Projekten vorzufindende Rollen sind im Folgenden dargestellt. Diese zeichnen sich durch spezifische Verantwortlichkeiten und Aufgaben im Projektalltag aus.

Der Projektleiter (klassisch)

- hat einen Projekt- und Aufgabenfokus
- sorgt für einen klaren Projektauftrag, entwickelt einen Projektplan und steuert das Projekt anhand des Plans
- fungiert als Schnittstelle zwischen Auftraggeber, den Fachbereichen und dem Projektteam
- ist für die Projektergebnisse verantwortlich, die das Team erarbeitet

Der Scrum Master

- hat einen Team- und Prozessfokus
- hat die Rahmenbedingungen herbeizuführen und zu erhalten für ein funktionierendes Miteinander im Team (Identifikation und Handling sogenannter Impediments)
- unterstützt den Product Owner größtenteils methodisch, übernimmt jedoch keinerlei Abstimmung mit den Fachbereichen
- ist für die Produktivität (Stichwort. Steigerung der Velocity) des Projektteams verantwortlich

Der Product Owner

- Management des Product Backlogs (Analyse der Anforderungen, Schreiben von User Storys, operative Verwaltung des Backlogs)
- Wertmaximierung des Produkts des Entwicklungsteams (Vertretung des Kundeninteresses)

- Federführend verantwortlich für das Produktergebnis (Abnahme der im Sprint Review vorgestellten Ergebnisse)

Diese grundsätzlichen Rollenbeschreibungen gelten für jede Größe von Projekt. Aber wie sieht es nun in wirklich größeren Kontexten aus? Wie es bei einer Skalierung nach dem Framework "SAFe"?

Welche Rollen gibt es in skalierten Frameworks?

Vor allem größere Organisationen stehen vor der Herausforderung, dass sich agile Frameworks wie Scrum auf einzelne Projekte und Teams mit bis zu neun Entwicklern beziehen. Für große Vorhaben mit vielen Teammitgliedern ist das keine Lösung. Durch die gesteigerte Anzahl an Mitgliedern in größeren Projekten steigt auch die Anzahl der Abhängigkeiten untereinander und folglich die Komplexität.

Eben für diese Problemstellung bieten skalierte Frameworks einen Lösungsansatz, um Scrum als Methode auch im großen Rahmen und für eine große Anzahl von Produktteams einzusetzen. Dabei können Unternehmen auf verschiedene Skalierungsframeworks zurückgreifen. Diese Frameworks sind in der Regel auf Basis von Scrum oder weiteren agilen Methoden entwickelt worden. Die wichtigsten und in der Praxis geläufigsten Frameworks sind Scaled Agile Framework (SAFe), Large Scale Scrum (LeSS) und Nexus. Innerhalb dieser Frameworks kommen weitere bedeutsame Projektrollen ins Spiel.

Release Train Engineer

Kommen wir in diesem Zusammenhang auf die erste zusätzliche Rolle: der sogenannte **Release Train Engineer** (RTE) bzw. der **Chief Product Owner** in SAFe.

In einer Scaled-Agile-Umgebung mit ihren diversen Teams, die alle von Scrum Mastern unterstützt werden, kann man den Release Train Engineer als "Super Scrum Master" oder "Chief Scrum Master" bezeichnen: Er ist sozusagen der oberste Scrum Master, mit dem alle Scrum Master der einzelnen Teams zusammenarbeiten und der Hindernisse auf einer größeren, allgemeinen Projektebene (auf dieser Ebene auch als "Programm" bezeichnet) aus dem Weg räumen kann. Ähnlich dem "normalen" Scrum Master sind seine Aufgaben auf die Steigerung der Reife des Teams im Gebrauch von Scrum sowie der Produktivität gerichtet. Mittel der Umsetzung sind im Schwerpunkt Moderation und Coaching.

Darüber hinaus fungiert der RTE als Coach für das gesamte Programm und als Moderator für die meisten Abläufe und Veranstaltungen, die zu SAFe gehören. Im skalierten Scrum gibt es noch weitere Events als die herkömmlichen Scrum Events, was sicherlich auch den Overhead steigert.

Chief Product Owner

Aber nicht nur die Scrum Master erhalten einen obersten Chef, sondern auch die Product Owner durch den Chief Product Owner: An einem gewissen Punkt in einem wachsenden Projekt kann es Sinn haben, eine Hierarchie von zusammenarbeitenden Product Ownern einzuführen. Die folgende Grafik (Bild 1) zeigt beispielhaft eine solche Hierarchie, bei der jeweils ein Product Owner pro Team zur Verfügung steht, zwei

Product Line Owner jeweils für eine Gruppierung von Teams sowie ein Chief Product Owner. Der Chief Product Owner (kurz: CPO) verwaltet Anforderungen auf der obersten Ebene (strategisch) und bricht diese (häufig EPICS genannt) auf untergeordnete Ebenen herunter

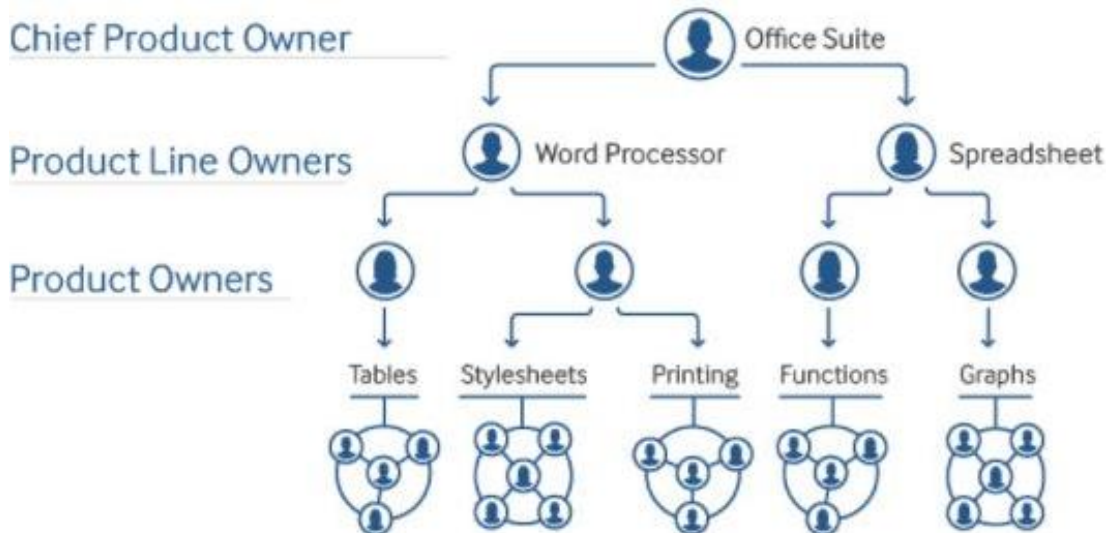


Bild 1: Die Hierarchieebenen der Product Owner in skalierten Projekten

Weitere fachliche Rollen (ohne Führungscharakter)

Eingangs erwähnte ich die m.E. in der Praxis unterschätzte inhaltliche Differenzierung der Rolle "Entwickler" in Scrum. Mit Blick auf heute angewendete Projektorganisationen finden wir (wie bereits angesprochen) Rollen wie Business Analysts, fachliche Tester und technische Tester. Aber über eine Funktion mit führendem Charakter haben wir noch nicht gesprochen. Denn typischerweise werden in Software-Entwicklungsprojekten sogenannte **Software-Architekten** eingesetzt.

Software-Architekt

Auch in diesem Fall sprechen wir zwar nicht von einer Scrum-Rolle, aber dennoch von einer höchstrelevanten Rolle, v.a. in größeren Projekten. Auch wenn der Architekt formal den Entwicklern gleichgestellt ist, trifft er doch grundlegende Designentscheidungen. Durch den Rückkopplungsprozess des Software-Architecturentwurfs wirken jedoch auch die Entwickler maßgeblich an der Architektur mit. Der Architekt trifft dabei globale Entscheidungen, lernt als Mitglied des Entwicklungsteams – zu dem er ja gehört – welches Design daraus auf tieferer Ebene entsteht und überprüft kontinuierlich, ob die Auswirkungen im Einklang mit den funktionalen und qualitativen Anforderungen des Kunden sind.

Der Architekt ist in der Rolle eines Kommunikators im Team. Er sollte mit einem technischen Fokus in der Lage sein, zwischen Entwicklern, Product Owner, Managern und anderen Interessengruppen zu vermitteln. Er kommuniziert die Architektur zwar schriftlich, aber auch im Projektalltag persönlich, ganz im Sinne des sechsten agilen Prinzips: "Die effizienteste und effektivste Methode, Informationen an und innerhalb eines

Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht." Sprint Planning und Sprint Review sind die Events in Scrum, um mit dem Entwicklungsteam über die Systemarchitektur zu diskutieren und sich auf ein gemeinsames Vorgehen zu verständigen.

Die Verbindung von fachlichen und technischen Aspekten mit Sozialkompetenz und Kommunikationsfähigkeiten stellt hohe Anforderungen an den Architekten. Bei unterschiedlichen Auffassungen im Entwicklungsteam, die etwa während des Sprint Plannings entstehen, ist es Aufgabe des Architekten, die Entscheidungsfindung in technischer Hinsicht zu unterstützen oder auch: letztlich zu entscheiden. Gerade bei verteilten Teams gibt es in der Regel mit dem Chefarchitekten eine Instanz, die notwendig für Richtungsgebung ist.

Der Architekt im agilen Umfeld arbeitet deutlich technik- und entwicklungsnahe als der Product Owner – was aber nicht heißt, dass der Product Owner nicht zumindest über ausreichend technisches Wissen verfügen sollte, das er benötigt, um die Auswirkungen von Kundenanforderungen grob einschätzen zu können.

! Der Product Owner vertritt die Produktvision, der Architekt vertritt technische Prinzipien.

Trotz aller Zusammenarbeit und Gleichwertigkeit der Kompetenzen sollte zwischen Product Owner und Architekt ein konstruktives Spannungsverhältnis bestehen: Der Product Owner hat vor allem auf die Nutzenmaximierung zu achten, d.h. auf die Wertigkeit der Features, die im Sprint umgesetzt werden, der Architekt hingegen hat Qualitätsaspekte des auszuliefernden Produkts im Fokus. Erfahrungsgemäß ist dieser Gegensatz Ursache zahlreicher Diskussionen, sobald im Projektverlauf der Terminplan eng oder das Budget knapp wird.

Business Analyst

Ohne diese Führungsrolle – die der Chef-Architekt in meinen Augen eindeutig hat – sind in heutigen Projekten **Business Analysten** für die Aufgaben der Architekten zuständig. Auch bei Business Analysten handelt es sich um keine Scrum-Rolle. Ein Business Analyst wird auch Requirements Engineer genannt und sorgt für die Richtigkeit und Vollständigkeit der Anforderungen.

Bei agilen Vorgehensmodellen steht die kontinuierliche Zusammenarbeit mit den (internen und externen) Kunden im Vordergrund. Herr über Anforderungen in einem agilen Projekt ist (wie gesehen) der Product Owner. Business Analysten hingegen können den Entwicklern auf einem genauer spezifizierten Grad diese Anforderungen in der geeigneten Detaillierung zur Verfügung stellen, damit diese das richtige Produkt entwickeln. Im Gegensatz zu langen Anforderungsdokumenten, z.B. bei Wasserfall-Vorgehensmodellen, werden tendenziell (aber nicht nur) eher "schlanke" Anforderungen z.B. in Form von User Storys geschrieben. Auch stellen Business Analysten die Anforderungen in jeder Iteration zur Verfügung und sammeln nicht zu Beginn des Projekts (möglichst) alle komplett spezifiziert Anforderungen ein. Business Analysten haben (ähnlich dem Architekten in der technischen Domäne) einen Überblick über die fachlichen Gesamtzusammenhänge (Zusammenspiel der Anforderungen) und sind ab einer gewissen Projektgröße und -komplexität in heutigen Projekten nicht mehr verzichtbar.

Kommunikation, Verhandlung und Moderation spielen dabei im agilen Umfeld eine große Rolle. Part des Business Analysten kann es sein, darauf hinzuwirken, dass die beteiligten Personen "eine Sprache sprechen". Dies kann z.B. durch ein formales (Projekt-)Glossar unterstützt werden, das der Business Analyst pflegt.

Einsatzfelder eines Business Analysten (anhand der operativen Aufgaben):

Support bei der fachlichen Ausarbeitung von Schnittstellen

- Übernahme der Schnittstellendokumentation (z.B. Interface Control Document (ICD))
- Unterstützung bei der Abstimmung mit anderen Teams (im Projekt, aber auch in der IT-Linien-Organisation)
- Unterstützung bei der Freigabedokumentation (z.B. Release-Dokumentation inkl. Liste bekannter Fehler)
- Dokumentation von fachlichen Features im Sinne einer kompakten Aufbereitung für den Kunden

Unterstützung des Product Owners

- Bei der iterativen Erhebung von Anforderungen und der Erstellung von User Storys
- Bei der Unterbreitung von entsprechenden Priorisierungsvorschlägen
- Bei der Vermittlung von Inhalten an das Entwicklerteam

Beitrag zur Verbesserung der Produktqualität

- Fachliche Unterstützung bei der Erstellung von Testfällen und Testszenarien
- Hinweisen auf mögliche Negativtestfälle
- Durchführungen von Testfällen und exploratives Testen (sofern ein Testteam nicht vorhanden ist oder unterstützt werden soll)

Tester

Vor allem das Testen ist für die Produktqualität wichtig. Für dies ist eine weitere Rolle in Projekten verantwortlich: Tester in Scrum Teams. Die Rolle der Tester lässt sich strukturieren in:

- **Test-Analyst:** Führt mit einem Schwerpunkt auf den Bereich "fachliche Testspezifikation" eigenverantwortliche Tätigkeiten für ein breiteres Testfeld durch.
- **Test-Designer:** Führt mit einem Schwerpunkt auf die Bereiche "technische Testspezifikation" und "Testrealisierung" eigenverantwortlich die Abbildung der fachlichen Testziele auf die technische Umsetzung durch und kann kleine Teams führen.
- **Testmanager:** Leitet mittelgroße Testteams (bis 6 Tester) und führt verantwortlich die Aufgaben "Testplanung" und "Teststeuerung" durch. Ist verantwortlich für das "Testreporting" und den "Testprojektabschluss". Führt das Testteam für eine oder mehrere Teststufen.

- **Testdaten-Manager:** Unterstützt die Testrealisierung und Testdurchführung bei datenlastigen Testobjekten.
- **Testautomatisierungs-Spezialist:** Führt die Kodierung von Testfällen zur automatisierten Ausführung nach Anweisung durch.

Welche Gefahren bestehen bei der Vielfalt dieser Rollen?

Wie passen diese genannten Rollen nun eigentlich zusammen? Das sogenannte Kongruenzprinzip der Organisationslehre besagt, dass die Verantwortung für eine Aufgabe untrennbar mit den dafür erforderlichen Befugnissen verbunden sein sollte. Betrachten wir die oben skizzierten Rollen, stellt man fest:

- Der für die Team-Performance **zuständige Scrum Master hat keine Ergebnisverantwortung und auch keinen Stakeholder-Kontakt.**
- Der für die Wertmaximierung **zuständige Product Owner hat keinen Einfluss auf die Team-Performance** (d.h. auf die Aufgabe des Scrum Masters).
- Der **Projektleiter hat weder Einfluss auf das Produkt noch auf die Team-Performance**, muss aber die Budgets (und irgendwie auch das Ergebnis) verantworten.

Dies hat Konsequenzen. Ist das Projekt erst einmal in einer Schieflage, lässt sich schwer analysieren, was die Ursache ist. Folgende Ursachen sind möglich:

- **Es gibt ein Scope-Problem:** Die Geschäftsanforderungen werden nicht geeignet adressiert oder nicht klar genug im Rahmen von User Storys operationalisiert
- Die **Teamperformance (Stichwort Velocity) reicht nicht aus**, um Sprintziele zu erreichen. Möglicherweise ist allein die Durchführung der Scrum Events ein echter Zeitkiller.
- Die **Kommunikation mit der Kunden-/ Sponsorseite ist unzureichend**, so dass unproduktiver Druck von außen in die Teams getragen wird.

Sagen wir es direkt: wer trägt die Verantwortung? Wir haben ja gesehen: Viele Rollen haben Einfluss auf den Gesamterfolg eines Projekts, gerade auch Rollen mit Führungscharakter. Insofern ist es umso schwerer, in kritischen Situationen nun die entscheidende Stelle auszumachen.

Beispiel für eine Rollenverteilung in der Praxis

In einem realen Projekt, in dem ich selbst als Business Analyst mitwirkte, existierten die nachfolgend aufgeführten Rollen. Es handelte sich hierbei um ein Software-Entwicklungsprojekt orientiert an SAFe. Projektgegenstand war der Bau eines Kundenportals im Bereich Wirtschaftsauskunftei. Es handelte sich prinzipiell um ein internes Projekt, das mit beträchtlicher externer Unterstützung durchgeführt wurde. Gesamtprojektleiter und Scrum Master waren hierbei extern. Nun zu den Rollen:

- 1 (Gesamt-)Projektleiter

- 1 Release Train Engineer (RTE)
- 1 Chief Architect
- 3 Scrum Master (einer der Scrum Master betreute 2 Scrum Teams)
- 4 Product Owner für die 4 Scrum Teams
- 1 Product Manager
- 1 IT Product Manager
- 3 Business Analysts
- 5 Tester
- 4 Teammitglieder Infrastrukturteam
- 6-8 (dies änderte sich im Projektverlauf immer wieder) Entwickler: Und dies in einem Softwareprojekt!

Besagtes Projekt geriet "gefühl" – und das drückt an sich schon ein Problem aus – in eine Schiefelage. Diese empfundene Schiefelage bestand in der aus Auftraggebersicht in der zu späten Fertigstellung des Produkts, obwohl der Zieltermin vorab willkürlich gesetzt war. Zudem beruhte die Setzung des Zieltermins nicht auf einer Aufwandsschätzung durch Experten. Vielmehr wurde die vorhandene Aufwandsschätzung ignoriert.

Aus heutiger Sicht sehe ich folgende Probleme:

- Geeignete Metriken zur Messung des Projektfortschritts und zur Steuerung des Projekts wurden nicht definiert und nachgehalten. So allein ergab sich die Situation, dass das Projekt "vom Gefühl her" den Termin(-zielen) hinterherlief.
- Der Umstand, dass etwa die Hälfte aller Projektmitglieder in irgendeiner Form Leitungsaufgaben in der Linie hatten, führte dazu, dass ein erheblicher Teil des Projektteams fast permanent in Status- und Abstimmungsterminen verschwand und für direkt produktive Tätigkeiten kaum Kapazitäten hatte.
- Die Tatsache, dass in Folge der Situation der Projektsponsor ausgerechnet Scrum Master als Verantwortliche ausmachte, zeigte ein fehlendes Verständnis für Rollen in Scrum und damit für agile Projektabwicklung als Solches insgesamt.

Letzter Punkt spricht einen Aspekt an, der nicht zu vernachlässigen ist, wenn es um (vermeintlich) agile Projektumsetzung geht, es stellen sich aus meiner Sicht folgende Fragen für einen Auftraggeber:

- Wird der agile Ansatz für ein Projekt vom Projektsponsor getragen?
- Weiß er eigentlich, was das bedeutet und beinhaltet?
- Ist der agile Ansatz für die konkrete Aufgabenstellung der Richtige?

Empfehlungen für die Praxis

Wenn Projekte initiiert werden, ist in der Regel wenig strittig, welche Rollen (und Qualifikationen) zur Erreichung der Projektziele notwendig sind. Mit der Erstellung eines Projekt-Organigramms (Übersicht über die Projektorganisation) ist der Umstand scheinbar schnell berücksichtigt. Scheinbar. Die genaue Definition und Abgrenzungen von Verantwortung im Detail jedoch ist umso empfehlenswerter, je mehr Rollen in einem Projekt unterscheiden werden (müssen). Ich empfehle folgendes:

Für den Auftraggeber und die Beteiligten in der Projektinitiierung (z.B. der angehende Projektleiter):

- **Pessimistisch planen und denken:** Gerade in Frühphasen eines Projekts ist der Optimismus groß und man ist dazu verleitet zu denken: "Das findet sich alles schon (irgendwann)." Hier sollten gleich zu Beginn Risiken identifiziert und entsprechende Puffer vorgesehen werden.
- **Erstellung und Abstimmung einer RACI-Matrix** (s. Bild 2) mit den Funktionen "führt aus", "verantwortet das Ergebnis", "wirkt mit", "muss informiert werden" und "entscheidet". Nur so bekommt man Unklarheiten bezüglich der Verantwortlichkeit in den Griff.
- **Kritisch hinterfragen, ob das agile Vorgehen überhaupt das Richtige ist:** Passt es zum Projektgegenstand? Sind die Vorteile von Scrum für das, was erreicht werden soll, eigentlich maßgeblich? Und: Passt es zur Unternehmenskultur? Zwingt man ein Projekt in ein scheinbar agiles Korsett, das eigentlich von Grund auf nicht gewollt ist? Seien Sie nicht zwanghaft agil, weil es gerade in Mode ist.

Für den Projektleiter in der Projektdurchführung:

- **Den Scrum Master in die Pflicht nehmen:** Mit der Administration und Nachpflege von JIRA Boards und Impediment-Listen, Nachbereitung von Terminen wirkt der Scrum Master oft gut ausgelastet. Seine Coaching-Rolle – in klarer Abgrenzung zum Product Owner – ist sein wichtigster Part zu Steigerung der Team Velocity.
- **Festlegung von Regelterminen über die Scrum Events hinaus** mit ihren jeweiligen Zielsetzungen (Was soll erreicht werden?). Dabei kritisch überprüfen, welcher Teilnehmerkreis jeweils wirklich notwendig ist.
- **Einhaltung der Zeiten bei den time-boxed Scrum Events.** Werden diese regelmäßig und systematisch nicht eingehalten, stecken häufig andere Probleme dahinter: fehlende Teamkommunikation jenseits des Dailys, unklar formulierte Storys bestehend aus losen Stichwörtern und Platzhaltern etc.
- **Klare Eskalationsmechanismen bei schwerwiegenden Problemen** durch aussagekräftige und transparente Statusberichterstattung.

	Projektziele erstellen	Beitrag/Praxis ermöglichen	Anforderungen erheben	Funktionsweise/Konzept erstellen	Ergonomisches Design erstellen	IT-Infrastruktur konzipieren	Sicherheitskonzept erstellen	Bestenfalls Pflichtbarkeit	Abnahme/Pflichtbarkeit
Lenkungsausschuss	A	I	I	I	I	I	I	I	A
Projektleitung	R	A	A	A	A	A	A	A	R
System-Analyse	C	R	R	R	I	I	I	C	I
Design	C	I	I	I	R	I	C	I	
IT-Abteilung	C	I	I	I	I	R	I	C	I
Qualitätssicherung		I	I					R	I
Verkaufsabteilung	C	I	I	C	C	I	I	C	I
ERP-Verantwortlicher	C	I	I	C		C		C	I
Sicherheitsbeauftragter	C	I	I				R	C	I
Externer Berater		C							
Betriebsrat	I								

- R – Responsible: verantwortlich für die Durchführung
- A – Accountable: verantwortlich für die Entscheidungen im kaufmännischen und im rechtlichen Sinne
- C – Consulted: eine Person (Rolle), die zur Beratung herangezogen werden muss
- I – Informed: jemand, der informiert werden muss

Bild 2: Beispiel für eine RACI-Matrix

Fazit

Auch in der heutigen, "agilen" Welt ist u.a. der Projektleiter (immer noch) nicht verschwunden. Aber auch sonst wirken mehr Rollen auf den Projekterfolg ein als z.B. die reinen "Entwickler" in Scrum. Allerdings liegen Aufgaben wie Scope Management und Steigerung der Team-Performance in der agilen Welt nicht mehr in der primären Verantwortung des Projektleiters. Für die Team-Performance gibt es mit dem Scrum Master einen Spezialisten. Die Fachlichkeit ist in der Domäne des Product Owners verankert und es gibt als technisches Pendant die Architektenrolle, jedoch in verschiedenen Ausprägungen: Chef-Architekt, Software-Architekt oder schlicht IT-Architekt.

Klare Kompetenz- und Verantwortlichkeitsabgrenzung ist wichtig, um zu vermeiden, dass sich die Rolleninhaber in der Entscheidungsfindung in die Quere kommen, und um eine unnötige "Verkopfung" eines Projekts zu vermeiden. Dies wäre der Fall, wenn z.B. viele entscheiden, aber keiner mehr praktisch Entscheidungen umsetzt. Gerade diese Umsetzer sind jedoch wichtig: Während der Scrum Guide nur "Developers" kennt, gibt es in der Praxis oft Business Analysts und (eher noch häufiger) Tester in Projektteams. Business Analysten müssen den Product Owner beim Anforderungsmanagement unterstützen und somit entlasten. Zusätzlich haben sie mit Entwicklern und Testern ihre Abnehmer bzw. Kunden und sind damit eine klare Stellschraube zur Effizienz im Projekt.

Hat Ihnen dieser Artikel gefallen?

Bewerten und kommentieren Sie den Artikel auf projektmagazin.de!

[zum Artikel](#)