



□ Bernd Stehr, M.Sc.

(Bernd.Stehr@t-systems.com)

ist als Consultant bei der T-Systems on site services GmbH tätig und berät Kunden aus der Automotive-Branche in den Bereichen Prozessmanagement und IT-Projektmanagement.



□ Artur Strasser, M.Sc.

(Artur.Strasser@t-systems.com)

ist als Senior Consultant bei der T-Systems on site services GmbH tätig und berät Kunden aus der Automotive-Branche. Weiterhin promoviert er zum Thema „IS Offshoring“ an der TU Dresden und ist Dozent für Projektmanagement, Software- und Requirements Engineering an mehreren Hochschulen in Niedersachsen.

Der Requirements Engineer – Auslaufmodell oder Erfolgsfaktor?

Viele Unternehmen wechseln zur agilen Softwareentwicklung und verwenden Methoden wie z. B. Scrum. Das Rollenmodell von Scrum enthält den Requirements Engineer in seiner traditionellen Ausprägungsform nicht. In diesem Beitrag wird diskutiert, ob sich der Requirements Engineer daher zum Auslaufmodell entwickeln könnte oder nach wie vor einen entscheidenden Erfolgsfaktor darstellen wird. Es werden Erfahrungen aus der Praxis vorgestellt und Handlungsempfehlungen gegeben, um dieser Situation begegnen zu können.

Einleitung

Die Einführung und Professionalisierung des Requirements Engineerings (RE) im Unternehmen birgt weitreichende Erfolgsaussichten. Neben der Steigerung des Projekterfolgs konnte u. a. bei 59 % der Unternehmen, die RE einsetzen, die Qualität der Softwareprodukte wesentlich verbessert werden. Dies ergab eine Umfrage, an der sich ca. 300 Probanden im Sommer letzten Jahres beteiligt haben (vgl. [Ada13]). Ein qualifizierter Requirements Engineer, der die Methoden und Vorgehensweisen des RE beherrscht, nimmt häufig eine wesentliche Rolle in klassischen IT-Projekten ein und trägt maßgeblich zum Projekterfolg bei.

Nun findet in vielen Unternehmen ein Wandel von der klassischen zur agilen Softwareentwicklung unter Verwendung von Methoden, wie z. B. Scrum oder Kanban, statt (vgl. [Fuc12]). Im Rollenmodell von Scrum ist die traditionelle Rolle des Requirements Engineers nicht explizit enthalten.

Ein Blick in den Scrum Guide (vgl. [Sch13]) zeigt auf, dass Scrum sich im Hinblick auf das RE auf die Beschreibung von Aufgaben, die der Anforderungsverwaltung zuzuordnen sind, beschränkt und

die Rolle des Product Owners die dazugehörigen Aufgaben verantwortet. Woher die Anforderungen kommen, wie sie ermittelt und formuliert werden, lässt Scrum bewusst offen (vgl. [Wir11]).

Weiterhin lässt sich in der Praxis beobachten, dass die Product Owner-Rolle häufig durch einen Mitarbeiter aus einer Fach- oder IT-Abteilung eingenommen wird, der vergleichsweise geringe RE-Kenntnisse besitzt. Diese Aspekte werfen einige Fragen auf:

- Sollte dem RE, das sich in klassischen Vorgehensmodellen als Erfolgsfaktor dargestellt hat, in agilen Projekten weniger Aufmerksamkeit beigemessen werden?
- Ist das RE in agilen Projekten von hoher Relevanz?
- Nimmt der Product Owner die RE-Aufgaben selbstständig wahr oder benötigt er Unterstützung dabei?

Die Praxis zeigt, dass diese Fragen nicht eindeutig beantwortet werden können. Im Nachfolgenden wird das RE im klassischen und im agilen Umfeld näher betrachtet und es werden Handlungsempfehlungen gegeben, um – je nach Projektsituation – die

erforderlichen RE-Aufgaben im Projekt bewältigen zu können.

RE im klassischen Umfeld

In klassischen Vorgehensmodellen (z. B. Phasen- und Wasserfallmodell) umfasst die Softwareentwicklung in Abhängigkeit der Komplexität eines Projektes und der IT-Projektorganisation den Einsatz verschiedener Rollen. Die wesentlichen Rollen im klassischen Umfeld – nach Projektträger und IT-Projektteam differenziert – sind in **Abbildung 1** dargestellt.

Der Projektträger setzt sich in der Regel aus dem Lenkungsausschuss bzw. Steuerkreis, dem Auftraggeber bzw. Kunden sowie dem IT-Projektcontrolling zusammen. Das IT-Projektteam besteht im Kern aus den Rollen des IT-Projektleiters und ggf. weiteren Teilprojektleitern sowie dem Entwicklungsteam, bestehend aus dem Requirements Engineer, Softwareentwickler, Softwaretester und Systemarchitekten.

Der Requirements Engineer nimmt im Entwicklungsteam eine zentrale Rolle ein. Er ist direkter Ansprechpartner für den Auftraggeber/Kunden zur Klärung der fachlichen Anforderungen und arbeitet eng mit der IT-Entwicklung zwecks technischer Realisierung zusammen.

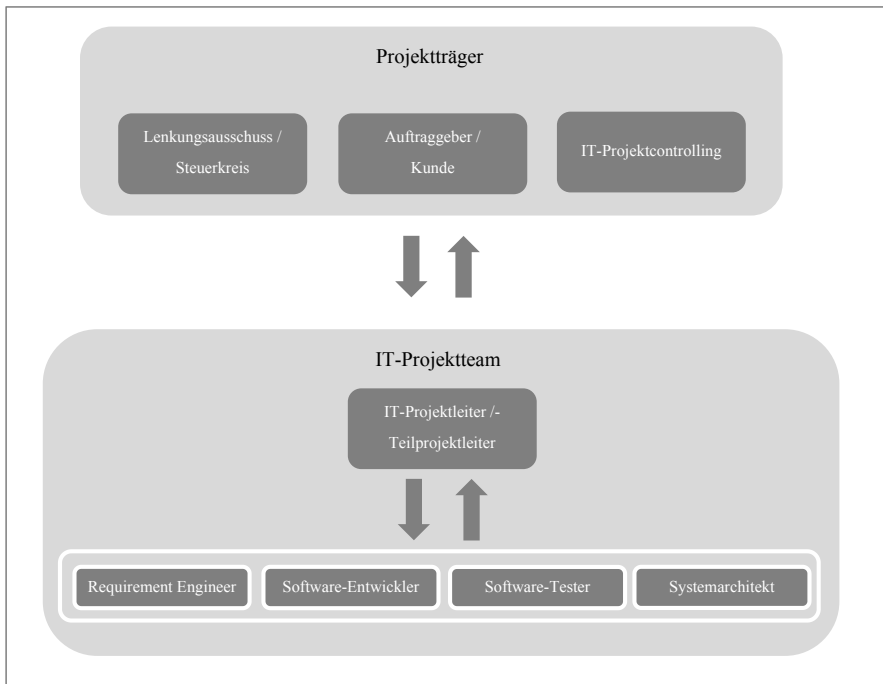


Abb. 1: IT-Projektumfeld in der klassischen Softwareentwicklung

Das RE ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen, u. a. mit dem Ziel, Konsens unter den Stakeholdern über die jeweiligen Anforderungen herzustellen. Der Kunde befindet sich hinsichtlich seiner Anforderungen in enger Abstimmung mit dem Requirements Engineer, welcher für das Ermitteln, Dokumentieren, Validieren und Verwalten der Anforderungen verantwortlich ist.

Hierfür nutzt der Requirements Engineer entsprechende Methoden. Zur Erhebung von Anforderungen benutzt er z. B. Befragungs-, Beobachtungs- und Kreativitätstechniken. Die Dokumentation von Anforderungen erfolgt häufig in Prosaform (Lastenhefte).

Darüber hinaus besitzt der Requirements Engineer die Fähigkeit, Anforderungen nicht nur textuell, sondern auch modellbasiert zu dokumentieren. Hierfür können z. B. verschiedene Modellierungsformen und Notationen der UML zum Einsatz kommen. Dazu gehören u. a. Aktivitäts- und Klassendiagramme. Methoden zur Prüfung der Anforderungen, die der Requirements Engineer häufig einsetzt, sind beispielsweise die Stellungnahme, perspektivbasiertes Lesen sowie das Prototyping. Das Verwalten der Anforderungen findet u. a. mittels Attributierung und Priorisierung statt, was durch den Einsatz von Softwarelösungen unterstützt werden kann.

In klassischen Vorgehensmodellen, die nach wie vor häufig zum Einsatz kommen, werden die Anforderungen zu Beginn erhoben und es wird streng sequenziell nach vorgegebenen und aufeinanderfolgenden Phasen vorgegangen. So ist das Ergebnis der Problemanalyse und Grobplanung im Phasenmodell die Beschreibung aller

Anforderungen. Ähnlich ist es beim weiterentwickelten Wasserfallmodell, bei dem das RE zu Beginn stattfindet und zusätzlich Iterationen, im Idealfall zwischen zwei direkt aufeinanderfolgenden Phasen, möglich sind.

RE im agilen Umfeld

Die am stärksten verbreitete leichtgewichtige Methode des Software Engineerings ist Scrum bzw. Varianten von Scrum, wie die aktuelle Studie des Softwareherstellers VersionOne in Zusammenarbeit mit Analysis.Net Research unterstreicht (vgl. [Ver13]).

Scrum besteht aus wenigen Rollen, Regeln, Prinzipien und Werten. Das IT-Projektteam im agilen Umfeld sieht sich ebenso wie im klassischen Umfeld dem Projektträger gegenüber. Die Zusammensetzung im IT-Projektteam verändert sich jedoch, denn bei Scrum weichen die Rollen von denen bei klassischen Vorgehensmodellen ab (vgl. **Abbildung 2**).

Es existieren die Rollen des Product Owner, Scrum Masters und die des Entwicklungsteams. Die letztgenannte Rolle wird gemäß [Sch13] nicht in die jeweiligen Teildisziplinen (Entwickler, Tester, Analyst, usw.) unterteilt. Alle Teammitglieder werden als Entwickler bezeichnet, können jedoch spezialisierte Fähigkeiten besitzen und praktizieren.

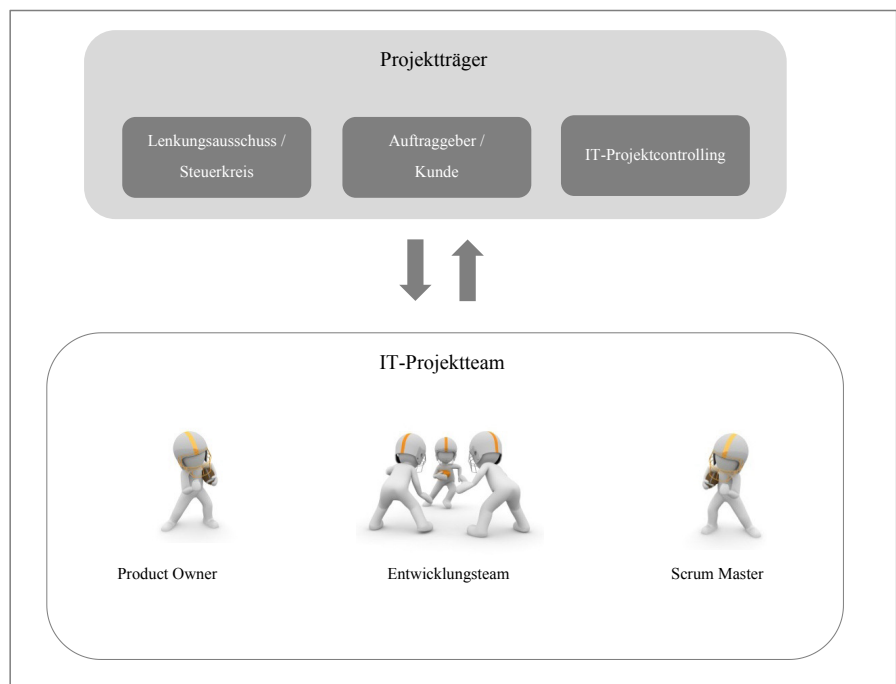


Abb. 2: IT-Projektumfeld in der agilen Softwareentwicklung

Kommunikation und Eigenverantwortlichkeit ohne Hierarchien charakterisieren das Scrum-Team. Einen Projektleiter bzw. Teilprojektleiter mit entsprechender Führungskompetenz gibt es, ebenso wie einen Requirements Engineer, nicht.

Anforderungen werden nicht zu Beginn möglichst exakt und vollständig beschrieben, sondern das Ausformulieren von klaren Funktionalitäten aus Anwendersicht steht im Vordergrund. Diese werden in zwei bis vier Wochen langen, sich wiederholenden Intervallen, iterativ und inkrementell umgesetzt.

Im Vergleich zum Wasserfallmodell wird bei Scrum mehr Wert auf funktionsfähige Software als auf eine umfassende Dokumentation gelegt. Anforderungen werden in direkter Zusammenarbeit mit den Anwendern erhoben.

Auch wenn die Rolle des Requirements Engineers als solche, wie sie in klassischen Vorgehensmodellen vorkommt, nicht definiert ist, so ist festzustellen, dass die Erhebung, Dokumentation, Validierung und Verwaltung von Anforderungen weiterhin erforderlich ist. Es stellt sich die Frage, ob der Product Owner diese Aufgaben wahrnehmen kann bzw. ob er ggf. Unterstützung zur Bewältigung dieser Aufgaben benötigt.

In der Praxis entsteht oft ein Vakuum, wenn Domänenwissen, Zeit sowie die Methodik des RE nicht ausreichend vorhanden sind. Nachfolgend wird kurz auf die Problematik eingegangen und es werden Handlungsanweisungen beschrieben, die in der Praxis zur Lösung beitragen können.

Handlungsanweisungen

Der Product Owner ist häufig überlastet oder nicht bevollmächtigt, die notwendigen Entscheidungen zu treffen (vgl. [Pic08]). Weiterhin ist das Maß an RE- und Domänenwissen, das für die Ausübung der Tätigkeiten erforderlich ist, häufig nicht ausreichend vorhanden. Die Rand- und Rahmenbedingungen des Geschäftsumfelds sind letztendlich entscheidend und beeinflussen die konkrete Besetzung der Rolle durch eine Person oder ggf. mehrere Personen.

Die nachfolgend beschriebenen Voraussetzungen und Handlungsanweisungen basieren auf Erfahrungswerten aus verschiedenen Softwareentwicklungsprojekten. So gehen die verschiedenen Anweisungen davon aus, dass ein Requirements Engineer eine sinnvolle Unterstützung für

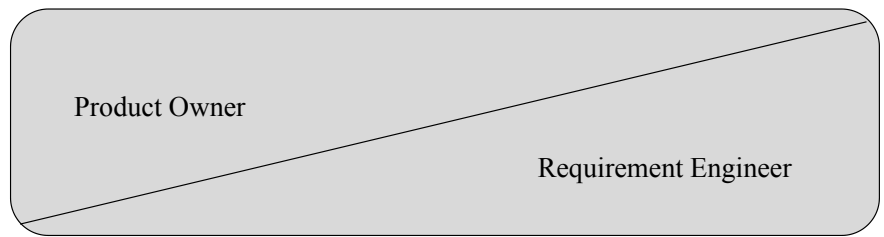


Abb. 3: Voraussetzungen zur Besetzung der Rolle

einen Product Owner darstellt, falls dieser nicht über ausreichend RE- und Domänenwissen bzw. Zeit verfügt.

Verschiedene kleine und mittelgroße mit Scrum durchgeführte IT-Projekte haben aufgezeigt, dass ein Requirements Engineer mittel- bzw. längerfristig unterstützen bzw. sogar die Rolle des Product Owners einnehmen kann. Letzteres sollte eine Ausnahme darstellen, falls für das angestrebte Projekt keine geeignetere Lösung zu finden ist. In diesem Fall sollte beachtet werden, dass der Requirements Engineer das vollständige „Mandat“ zur Erfüllung seiner Aufgaben benötigt, um seiner Verantwortung nachzukommen.

Besitzt der Product Owner ausreichend RE- und Domänenwissen sowie Zeit, so kann eine kurzfristige bis keine Unterstützung durch einen Requirements Engineer erfolgen. In diesem Fall ist es gelungen, eine Person zu identifizieren, die den Aufgaben und Verantwortlichkeiten dieser Rolle gerecht wird. Eine zusätzliche Unterstützung, um etwaige Schwachstellen auszugleichen, wird damit hinfällig.

Grundsätzlich kann erfahrungsgemäß festgehalten werden: Je geringer eine sinnvolle Besetzung möglich ist, desto eher ist eine Unterstützung durch einen (oder mehrere) Requirements Engineer anzuraten (siehe [Abbildung 3](#)). Ist die Besetzung der Product Owner-Rolle nicht sinnvoll möglich und es wird keine ausreichende Unterstützung durch einen Requirements Engineer gewährt, so ist der Projekterfolg i.d.R. in hohem Maße gefährdet.

Je nach Ausgangslage ergeben sich daraus die nachfolgenden vier Handlungsanweisungen:

(1) Der Product Owner nimmt die entsprechenden Aufgaben selbstständig, ohne Unterstützung eines Requirements Engineers, wahr. Im Regelfall ist die

Ausübung der Tätigkeiten eine Vollzeitaufgabe, insbesondere in komplexen bzw. innovativen IT-Projekten.

(2) Ein Requirements Engineer nimmt die Rolle des Product Owners ein. Der Requirements Engineer führt über die gesamte Projektlaufzeit die Aufgaben des Product Owners aus und trägt die damit verbundene Verantwortung. Analog zur vorherigen Handlungsanweisung, sollte er möglichst Vollzeit verfügbar sein und kann vom Entwicklungsteam unterstützt werden. Bei dieser Variante sollte insbesondere der Grad des erforderlichen Domänenwissens hinterfragt werden. In Projekten mit vielen und/oder verteilten internen oder externen Stakeholdern kann diese Variante, z.B. zur Neuentwicklung eines IT-Systems, erfolgversprechend eingesetzt werden.

(3) Der Requirements Engineer unterstützt temporär und eher kurzfristig den Product Owner bei seinen Aufgaben. Dies ist vor allem zu Projektbeginn sinnvoll, um den Product Owner frühzeitig bei der Ermittlung und Erfassung der Kundenbedürfnisse zu unterstützen und eventuellen Schwierigkeiten, z.B. bei der Erstellung des initialen Product Backlogs, vorzubeugen. Jegliche Entscheidungen trifft der Product Owner und verantwortet sie gegenüber seiner Organisation. Der Requirements Engineer steht dem Product Owner beratend zur Seite. Der Product Owner kann bestimmte Themengebiete an den Requirements Engineer delegieren, sollte aber der primäre Ansprechpartner für das Entwicklungsteam sein.

(4) Der Requirements Engineer unterstützt den Product Owner mittel- bzw. langfristig oder sogar über die gesamte Projektlaufzeit. Er positioniert sich entweder hinter dem Product Owner oder im Entwicklungsteam und übernimmt

des Öfteren kleinere Aufgabenbereiche eigenverantwortlich. Auch in dieser Konstellation liegt die Entscheidungsgewalt und Gesamtverantwortung beim Product Owner. Der Requirements Engineer wird jedoch vermehrt in Entscheidungsprozesse eingebunden und kann die Vertretungsfunktion für den Product Owner übernehmen. Insbesondere bei komplexen IT-Projekten mit vielen Stakeholdern kann diese Zusammenarbeitsstrategie sehr effektiv verlaufen.

Fazit

Der Requirements Engineer nimmt in klassischen Vorgehensmodellen eine wichtige Rolle ein. In agilen Entwicklungsmethoden ist die Rolle des Requirements Engineers zwar nicht explizit aufgeführt, die Aufgaben und Verantwortlichkeiten sind aber sehr wohl enthalten und primär dem Product Owner zugeordnet. Jedoch ist das Maß an RE- und Domänenwissen sowie Zeit, die der Product Owner für die Ausübung dieser Aufgaben benötigt, in der Praxis häufig nicht ausreichend.

Die vier beschriebenen Handlungsstrategien zeigen auf, dass der Requirements Engineer auch im agilen Umfeld mindestens unterstützen, wenn nicht sogar die Product Owner-Rolle einnehmen kann. Bevor eine der genannten Handlungsanweisungen ausgewählt wird, sollten die Rand- und Rahmenbedingungen des Geschäftsumfelds kritisch geprüft werden. Je nach Ausgangslage bieten sich anschließend verschiedenste Ansätze an. Jede Variante passt sich der entsprechenden Situation an, ohne die Werte von Scrum zu verändern. ■

Literatur

[Ada13] S. Adam, C. Wüsch, M. Koch, Ergebnisbericht RE-Kompass 2013, siehe:

http://www.re-kompass.de/fileadmin/project/re-kompass.com/uploads/documents/Ergebnisbericht_RE-Kompass_2013.pdf

[Fuc12] A. Fuchs, C. Stolze, O. Thomas, Von der klassischen zur agilen Softwareentwicklung: Evolution der Methoden am Beispiel eines Anwendungssystems, in: HMD - Praxis der Wirtschaftsinformatik, 290, 2012.

[Kom12] A. Komus, Agile Methoden in der Praxis – Studie zur Anwendung und Zufriedenheit, in: HMD - Praxis der Wirtschaftsinformatik, 290, 2012.

[Pic08] R. Pichler, Agiles Projektmanagement erfolgreich einsetzen, dpunkt 2008.

[Sch13] K. Schwaber, J. Sutherland, Scrum Guide, siehe: <https://www.scrum.org/Scrum-Guide>

[Ver13] VERSIONONE, 7th Annual State of Agile Development Survey 2013, siehe:

<http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf>

[Wir11] R. Wirdemann, Scrum mit User Stories, Hanser 2011.