

Ein klaffendes Loch im agilen Requirements Engineering

25 Juli 2013 | **Software Engineering** | [Markus Flückiger](#)

Lesezeit: 3 Minutes

„Wer schreibt denn nun die User Stories? Das muss doch der Product Owner tun, oder nicht? Aber was ist, wenn der nicht will oder kann?“ Dies ist eine Frage von vielen, die elf Praktiker in agiler Software Entwicklung im Rahmen des Zühlke Trainings [User Stories for Practitioners](#) diskutiert haben.

Mit Product Owner ist dabei natürlich die entsprechende Rolle aus Scrum gemeint und diese ist immer mal wieder mit Personen besetzt, die – allen Gurus zum Trotz – diese Rolle nur beschränkt ausfüllen können. Schliesslich haben Product Owners gelegentlich auch ein Tagesgeschäft, welches sie eigentlich zu 100% ausfüllt (z.B. als Führungskraft), und ein Leben ausserhalb des Berufs. Dabei besagt eine Faustregel, dass ein Product Owner ca. 30% in diese Rolle investieren sollte.

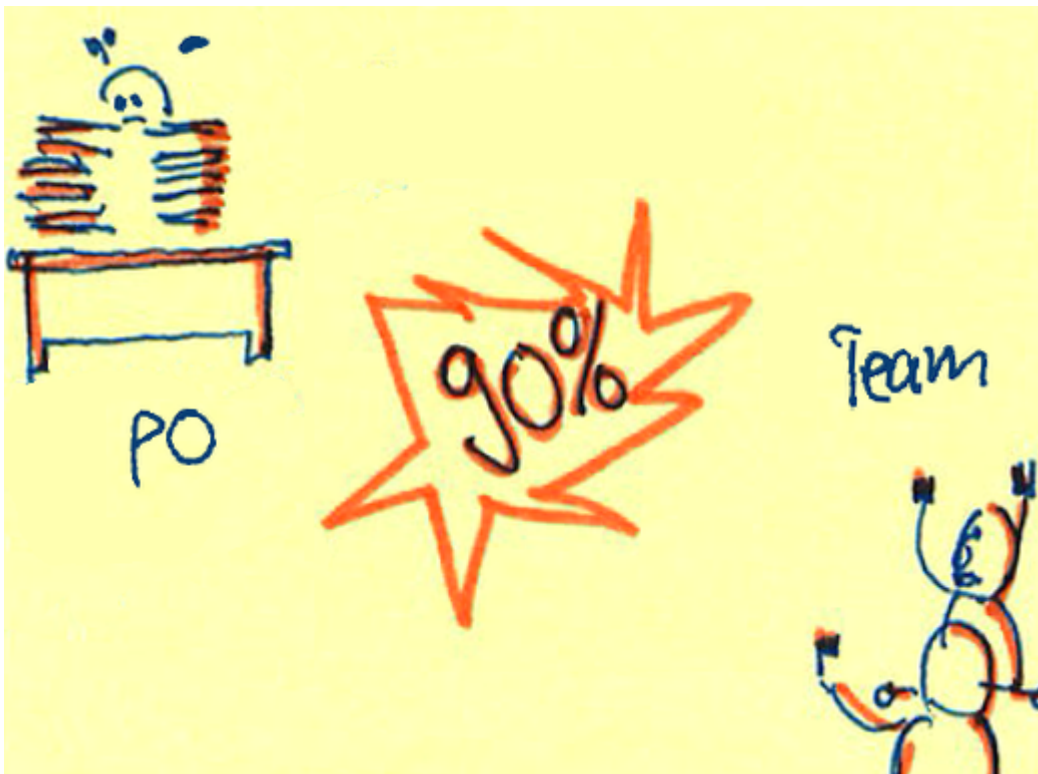
Nun kann sich natürlich das Team darüber ärgern und mit dem allseits beliebten Spruch „die Anforderungen sind unklar, so können wir nichts tun“ den Product Owner zur Mitarbeit zwingen. Dieses Bestreben ist allerdings unrealistisch, will das Team nicht gleich das ganze Unternehmen umbauen, und heizt den Konflikt nur weiter an.

Aber vielleicht braucht es hier etwas Klartext: Der Product Owner ist verantwortlich, dass Requirements Engineering gemacht wird. Es ist nicht seine Aufgabe, dem Team die Anforderungen als User Stories vorzukauen. Das wäre dann auch schnell eine 100% Aufgabe.

Zusammengefasst der geschilderte Systemkonflikt: Ein Product Owner hat ca. 10% Zeit für das Projekt. Die Faustregel besagt ca. 30% sind notwendig. Die Entwickler, die auf die vorverdauten Anforderungshäppchen warten, wollen 100%.

Genau wegen solchen Situation wollen wir agil sein! Denn dieser Deadlock ist Fakt und das Team muss einfach einen Weg herausfinden. Das agile Prinzip zur Auflösung ist Inspect und Adapt. Also zuerst die Situation verstehen und dann weg mit methodischen Dogmas, mit alten Hüten und anderen die Schuld in die Schuhe schieben. Hin zur Frage: Wie kriegen wir das bestmögliche Produkt hin?

Die Analyse hier zeigt ein klaffendes Loch im Requirements Engineering von sage und schreibe 90%!



Gap zwischen Product Owner und Team

100% der Requirements Engineering Aufgaben sind:

1. Gemeinsame Vision schaffen
2. VIPs ins Boot holen
3. Scope im Griff halten (Agile Slang: Backlog grooming)
4. Wertversprechen und Kosten optimieren (Agile Slang: Conversation. Im RE auch bekannt unter Requirements Negotiation)
5. Anforderungen erarbeiten und umsetzen (User Stories Slang: Card, Conversation, Confirmation)

Ein 10% PO wird mit (1) und (2) bereits gut ausgelastet sein und kann im Groben bei (3) mitdiskutieren. Doch schon ein PO, der (1) und (2) leisten kann, bringt einen riesigen Gewinn für das Projektteam!

Hier ein paar von den Teilnehmern in der Praxis angewandte Lösungsmuster, wie der Rest gefüllt werden kann:

- Product Owner aus dem Tagesgeschäft zu 30% freischaufeln.
- Product Owner ersetzen und den Ersatz auch bevollmächtigen(!)
- Product Owner Proxy einsetzen, der vor allem bei den Aufgaben (3) und (4) mitträgt.
- Fachvertreter ins Team holen, insbesondere für (5).
- Business Analyst ins Team holen, insbesondere für (3) bis (5).

- Anderes Rollenmodelle verwenden, beispielsweise aus DSDM oder XP.
- Team übernimmt die 90% als cross-functional Team.

Abschliessend bringt ein Teilnehmer dann seine Erkenntnis auf den Punkt: RE sei im agilen Umfeld ja genauso essentiell wie im nicht agilen. Es werde nur nach agilen Prinzipien durchgeführt und beispielsweise mit User Stories gesteuert. Und als eine Quintessenz für den User Stories for Practitioners Workshop kann man diese Aussage auch getrost stehen lassen.

Vielen Dank an die Teilnehmer für die spannenden Diskussionen!