

# Leadership @ Requirements Engineering: Wie dir Requirements-Management hilft Analysten-Teams zu organisieren

14 September 2016 | Insight Zühlke | [Manuel Hachem](#), [Martin Habicht](#)

**Lesezeit:** 4 Minutes

Oft konnte ich in den vergangenen Jahren beobachten, dass Business-Analysten oder Requirements-Engineers ad-hoc mit ihrer Arbeit beginnen. Das Modell nach dem sie arbeiten ist schnell gefunden, entweder agil mit User Stories oder sie sollen ein richtig dickes *Business-Specification-Document* erstellen, das zu genehmigen ist.

Das ad-hoc Vorgehen skaliert wunderbar, solange alleine gearbeitet wird. Je mehr Teammitglieder hinzustossen desto dringender stellt sich die Frage, wie wollen wir zusammenarbeiten?

- Wer bearbeitet welches Themengebiet?
- Wie stimmen wir uns im Projektverlauf untereinander ab?
- Wie bewirtschaften wir unseren Arbeitsvorrat?
- Welche Dokumente erstellen wir und welche sollten wir aufbewahren?

In grossen Projekten wird oft zuerst ein Lead-Business-Analyst eingesetzt. Der soll mit der Arbeit beginnen und später dem Analysten-Team erklären, wie alles funktioniert.

Im ungünstigen Fall passiert folgendes: Der Lead-Business-Analyst beginnt mit der Arbeit alleine. Er startet mit der Dokumentation von fachlichen Anforderungen auf Basis, was die Business-Stakeholder wollen. Und die Stakeholder wollen viel. Sie wollen umso mehr, je öfter mit ihnen geredet wird. Leider ist auch der Solution-Architect keine Hilfe. Er sitzt dem Lead-Business-Analysten bereits im Nacken, denn ohne Anforderungen kann er keine Schätzung abgeben, auf die alle warten.

Jetzt starten neue Mitglieder im Analysten-Team, die gespannt auf die Einarbeitung und Vorarbeiten des Lead-Business-Analysten sind.

*Was wird der Lead-Business-Analyst zuerst tun?*

Er delegiert Arbeit, die bisher liegen geblieben ist: «Bitte erhebt die nicht-funktionalen Anforderungen resp. Qualitätsanforderungen nach.»

*Doch was hilft dem Team, damit es abhebt?<sup>1</sup>*

- Das Team braucht ein gemeinsames Erfolgsziel und Aufgabe, die sie nur als Team lösen können.
- Das Team braucht Transparenz und Kommunikation, um bei der Zusammenarbeit abzuheben.

Ohne gemeinsames Ziel und Aufgabe wird aus einer Gruppe kein Team und aus einem Einzelgänger kein Teammitglied, denn er wird ohne das Team erfolgreich sein können. Wer als Leader mit negativen Konsequenzen für die Karriere droht<sup>2</sup>, wird niemanden begeistern. Besser ist das Erfolgsziel und die Etappensiege jedem im Team aufzuzeigen und regelmässig explizit zu kommunizieren. Beispielsweise beim Kick-Off Meeting, der individuellen Projekteinführung neuer Mitglieder und Team-Events.

Anstatt als Führungskraft «nur» zu delegieren hilft es dem Team mit Transparenz und Kommunikation aufzuzeigen, wohin jedem seine Reise geht:

### **Wer im Team bearbeitet welches Themengebiet?**

Transparenz und Kommunikation heisst, jedem zu zeigen, welche Arbeiten bereits gestartet wurden, welche Arbeiten als nächstes gestartet werden können und welche Arbeiten noch nicht bereit sind. Dabei hilft «priorisierte Landkarten» einzusetzen. Landkarten hängen an der Wand. Landkarten können bspw. mit RACI-Chart, Aktivitätsdiagrammen oder Gantt-Chart aufzeigen, welche Rollen und Aufgaben existieren. Ebenfalls können Know-how Matrizen aus Rollen gekreuzt mit Business Capabilities, Services, Produkte oder Prozesse dem Analysten-Team helfen sich zu orientieren und selbst zu organisieren. Delegiert wird ans Team und jeder im Team macht transparent, wofür er verantwortlich ist, z.B. mit seinem Namenseintrag in der Know-how Matrix oder RACI-Chart.

### **Wie stimmen wir uns im Projektverlauf im Team untereinander ab?**

Transparenz und Kommunikation heisst, jedem zu zeigen, wann ich verfügbar bin und welche Teilaufgabe ich gerade bearbeite. Dafür gilt es Metriken für Kapazitätsplanung, Rotationsplanung und Fortschrittsplanung (Burn-Up-Charts und Velocity) zu erstellen. Während sich Entwickler oft im Daily-Scrum Meeting untereinander austauschen, können wöchentliche Meetings im Analysten-Team genügen, um sich fachübergreifend abzustimmen. Nicht zu vergessen ist die Kommunikation zwischen Teams, wenn bspw. die Velocity des Entwickler-Teams steigt, dann können plötzlich mehr fertig spezifizierte Anforderungen nachgefragt werden. Ein institutionalisierter Austausch zwischen Analysten und Entwicklern, z.B. durch Refinement-Meetings, fördert die Kommunikation zwischen Teams.

### **Wie bewirtschaften wir unseren Arbeitsvorrat?**

Transparenz und Kommunikation heisst, den Arbeitsvorrat sichtbar zu machen und den Bearbeitungsstatus (bspw. mit einem Kanban-Board) zu kommunizieren. Anforderungen

müssen eine gewisse Güte haben, bis sie für alle erhoben, verstanden, widerspruchsfrei, machbar, priorisiert und finanziert sind. Die Stationen im Kanban-Board helfen zu verfolgen in welchem Gütestadium sich ein Feature befindet und Engpässe im Analyseprozess zu kommunizieren. Die Gesamtübersicht über alle Anforderungen im Arbeitsumfang (Scope) kann bspw. mit einer Story-Map gezeigt und priorisiert werden. Die Story-Map hilft einen Releaseplan sowie Scope- und Change-Management zu kommunizieren.

### **Welche Dokumente erstellen wir und welche sollten wir aufbewahren?**

Transparenz und Kommunikation heisst, Produkt- und Team-Standards im Requirements-Management-Plan (RMP) u.a. mit dem Requirements-Information-Model (RIM) zu beschreiben und jedes neue Teammitglied damit einzuarbeiten. Im RIM wird u.a. beschrieben, welche Anforderungsart, in welcher Tiefe und mit welcher Notation dokumentiert wird. Darüber hinaus kann im RMP definiert werden, was langlebige Produkt- und was kurzlebige Projektdokumentation ist; bspw. bei der Beschreibung der Verfolgbarkeit von Anforderungen von der Spezifikation, zum Softwarecode und Test in der Medizinalproduktentwicklung.

### **Weiterbildung zum Thema**

Wer mehr Erfahrungen in der Führung von Analysten-Teams austauschen möchte, empfehle ich gerne zwei Kurse:

Zum einen den Kurs «[Requirements in der agilen Softwareentwicklung](#)», um agile Tools anzuwenden (z.B. Story Map) und zum anderen den Kurs «[Requirements Engineering Advanced Level - Requirements Management](#)», um den Methodenbaukasten für das Management von Anforderungen fürs agile sowie das klassische Vorgehen kennenzulernen.

---

<sup>1</sup>Die besten Antworten werden mit einem Job-Angebot belohnt! ☐

<sup>2</sup>Mit Konsequenzen «drohen» kann lautstark oder unbewusst und subtil passieren.