

Qualität beginnt im Kopf – Sechs Praxistipps zur Verbesserung von Anforderungen

Mit welchen Mitteln können wir im Projektalltag dafür sorgen, dass trotz Zeit- und Budgetdruck genügend Ressourcen ins Erheben und Präzisieren von Anforderungen investiert werden? Wie können wir unser Umfeld davon überzeugen, dass es sich lohnt, die Qualität von Anforderungen zu verbessern? Alain Hofer

In der Neu- und Weiterentwicklung von Software oder Produkten stehen wir oft im Konflikt zwischen dem Bedürfnis, eine gute, ausgereifte und vollständige Lösung zu entwickeln, und der Tatsache der beschränkt vorhandenen Ressourcen. Dies wirkt sich oft schon früh im Projekt dadurch aus, dass keine Zeit für das saubere und seriöse Erheben von Kundenanforderungen besteht. Seriös in dem Sinne, dass die Anforderungen vollständig, konsistent, widerspruchsfrei und testbar formuliert werden, und von den Stakeholdern priorisiert und für gültig erklärt sind.

Die mit dem Formulieren und Erheben von Anforderungen betrauten Personen, wie Businessanalysten, Product Owner oder Requirements Engineers (hier im Weiteren als Businessanalysten bezeichnet), sind dafür mitverantwortlich, dass die nachgelagerten Aktivitäten wie die Entwicklung und das Testing auf einer soliden Basis aufsetzen können. Dies gilt sowohl für Produktentwicklungen im agilen als auch im traditionellen Wasserfall-Umfeld. Im Folgenden werden sechs konkrete Ideen beschrieben, die in hektischen Zeiten erfolgreich angewendet werden können.

Eins sei vorweggenommen: Das wichtigste Instrument für die Umsetzung der Ideen durch den Businessanalysten sind neben dem analytischen Denken seine sozialen Fähigkeiten, wie Empathie, Argumentationsgabe und Standhaftigkeit!

1. Scoping: Zeit durch Reduktion des Umfangs gewinnbringend nutzen

In bestehenden Produkten werden im Schnitt bis zu zwei Dritteln der Funktionalität nicht oder kaum verwendet. Viele Produkte werden



Alain Hofer
leitet das Büro von SwissQ in Bern, ist Businessanalyst und IREB-Kurs-trainer.

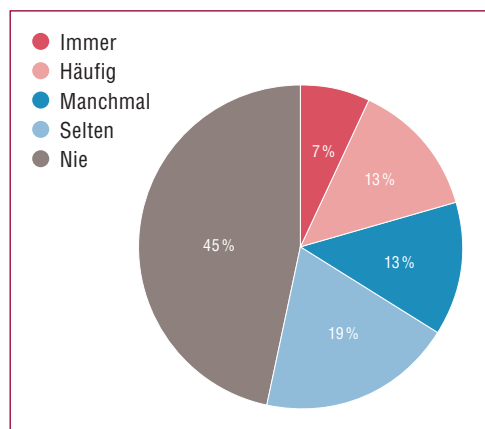


Bild 1: Knapp zwei Drittel aller Funktionalitäten eines Softwareprodukts werden nie oder sehr selten verwendet. Grafik: SwissQ, Daten: Standish Group, Chaos Report 2006

über Jahre weiterentwickelt, indem neue Funktionalitäten durch Projekte hinzugefügt werden, ohne bestehende Funktionen zu hinterfragen und entfernen. Welcher Businessanalyst kennt sie nicht, die Mutter aller Anforderungen, sinngemäss: Die neue Lösung soll mindestens alles, was bisher möglich war, unterstützen.

Bestehende Funktionalitäten und neue Anforderungen sollten also hinterfragt werden. Dazu können bei Erweiterung von bestehenden Produkten allenfalls auch Statistiken Aufschluss geben, welche Funktionalitäten selten oder nie verwendet werden.

Sind die Anforderungen für den geschäftlichen Erfolg und die Kundenzufriedenheit wirklich relevant? Und wie sehen es andere Stakeholder, wie beispielsweise Endbenutzer, die Security oder der Betrieb? Und sind diese Stakeholder auch bereit, für die Realisierung und den Betrieb einer Anforderung die Kosten zu übernehmen? Spätestens bei dieser Frage werden sich die Stakeholder wirklich überlegen, welchen Business Value sie einer Anforderung beimessen.

Je früher also im Prozess der Scope gesetzt werden kann, desto mehr können die Ressourcen auf das Wesentliche fokussiert werden. Eine Kundenanforderung kann zu vielen System- und Designanforderungen

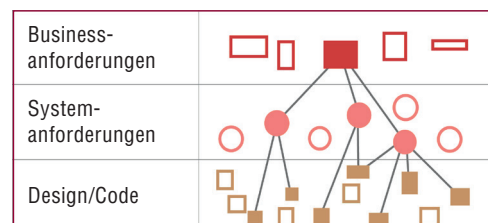


Bild 2: Hinter einer einzigen Kundenanforderung können viele System- und Designanforderungen stecken. Grafik: SwissQ

führen, und Auswirkungen auf viele Code-Elemente haben. Wenn eine solche Kundenanforderung früh verworfen oder präzisiert werden kann, hat dies unmittelbare Auswirkung auf die Ressourcenverteilung in den Folgeaktivitäten. Deshalb ist sicher eine der wichtigsten Aufgaben eines Businessanalysten, die Anforderungen zu hinterfragen und nach dem Warum einer Anforderung zu suchen. Eine beliebte, effektive und einfach anzuwendende Methode des Requirements Engineers ist, fünfmal die Warum-Frage zu stellen, um auf den Kern der Anforderung zu vorzustossen.

2. Priorisierung: das Wesentliche zuerst

In diesem Zusammenhang ist auch die Priorisierung zu sehen. Die Anforderungen werden gegeneinander nach wohldefinierten Kriterien priorisiert, wie zum Beispiel Business Value, Risiko oder Umsetzungskosten. Was in der agilen Softwareentwicklung erprobt ist und zum Backlog Management gehört wie das Gelbe zum Ei, kann in herkömmlichen Projektvorgehen genauso angewendet werden.

In einem ersten Schritt werden nur die Anforderungen umgesetzt, die für den geschäftlichen Erfolg unabdingbar sind. Nach diesem Schritt liegt bereits eine funktionierende Lösung vor, wenn auch mit eingeschränktem Funktionsumfang.

Falls noch Ressourcen übrig sind, können weitere Anforderungen zu einem späteren Zeitpunkt realisiert werden. Ganz nach dem Motto: lieber ein funktionierendes Produkt mit reduziertem Funktionsumfang als ein nicht funktionierendes Produkt mit vielen Funktionen. In ▶

der Praxis sehr bewährt hat sich zur objektiven und effizienten Priorisierung der Priority Poker (www.swissq.it/agile-de/priority-poker/).

3. Anforderungen auf Testbarkeit prüfen: Je früher ein Fehler entdeckt wird, desto besser

Es ist ein Trugschluss zu glauben, dass wir Zeit gewinnen könnten, wenn wir uns in frühen Projektphasen beeilen. Im Gegenteil: Jeder nicht entdeckte Fehler in einer Kunden- oder Systemanforderung kostet in späteren Projektphasen ein Vielfaches für dessen Behebung.

Was in der agilen Softwareentwicklung heute selbstverständlich ist, funktioniert sehr gut auch in einem wasserfallartig aufgesetzten Projekt. Früh werden Testpersonen zugezogen, die zu jeder Kunden- oder Systemanforderung einen Testfall definieren – oder mindestens eine Testidee skizzieren: Wie würde die Testperson die Anforderung testen oder abnehmen? Ist dies nicht möglich, ist die Anforderung allenfalls noch nicht präzise genug und lässt zu viel Interpretationsspielraum zu. Der IREB-Standard verlangt übrigens, dass jede Anforderung mit einem Abnahmekriterium hinterlegt wird.

Das frühe Anfügen eines Abnahmekriteriums ist umso wichtiger, weil als Spezifikations-sprache für Anforderungen heute immer noch mit über 50 Prozent Prosa am verbreitetsten ist. Und textbasierte Anforderungen lassen bekanntlich deutlich mehr Interpretationsspielraum zu als modellbasierte (Quelle: Trends & Benchmarks Report Schweiz, 2014).

4. Dokumentieren ja – aber was und wie?

Wer ist nicht schon mal zu einem laufenden Projekt dazugestossen, in dem zwar vieles dokumentiert war, aber einem trotzdem niemand sagen konnte, welche Anforderungen nun gültig sind, und weshalb andere zurückgestellt oder gelöscht wurden? Der Zusammenhang und die Historie zur entsprechenden Anforderung war nicht nachvollziehbar.

Die Dokumentation ist so eine Sache für sich. Man sollte grundsätzlich in Informationen denken und nicht in Dokumenten. Ein Dokument ist nichts anderes als eine Zusammenführung von Informationen zu einem bestimmten Zweck und zu einem wohldefinierten Zeitpunkt. So soll ein Anforderungsdokument mit Businessanforderungen beispielsweise dazu dienen, die Anforderungen des Kunden an die Lösung zu einem ganzen und für die Entwicklung verständlichen Bild zu führen. Beim Festhalten der Information kann man sich als Businessanalyst von den folgenden fünf Fragen leiten lassen:

- Wer interessiert sich für die zusammengestellten Informationen? (zB. Entwicklung, Testing, Betrieb oder Fach).

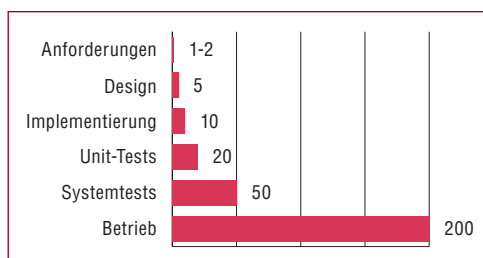


Bild 3: Kosten von Softwarefehlern, abhängig von der Projektphase. Grafik: Nach Stuart R. Faulk 1995

- Wie sollen die Informationen festgehalten sein? (Modell, Text, Kombination, elektronisch, auf Papier).
- Welche Informationen werden das Projekt nachhaltig überleben?
- Welche Vorgaben sind allenfalls einzuhalten?
- Welche Metainformationen sind mit festzuhalten? (Dazu gehören nach IREB.de sicher die Quelle, der Status, Verknüpfungen zu anderen Objekten und die Historie).

Requirement-Management-Werkzeuge sind eine wesentliche Unterstützung im Verwalten dieser Informationen und den dazugehörigen Metainformationen und erlauben es, entsprechende Dokumente daraus zu generieren. Herkömmliche Textverarbeitungs- und Tabellenkalkulationsprogramme unterstützen das Verwalten der einzelnen Informationsbausteine zu wenig.

Genauso wichtig ist es, festzuhalten, was nicht realisiert werden soll, und die Entscheidung dazu zu dokumentieren. Bei personeller Änderung muss nachvollziehbar sein, wann und weshalb gewisse Anforderungen gelten oder zurückgestellt wurden.

5. Agile Denk- und Handlungsweise leben

Nicht alle unserer Kunden haben die Gelegenheit, ein Projekt nach rein agilen Methoden durchzuführen. Doch auch in wasserfallartigen Vorgehensmethoden können agile Handlungsweisen übernommen werden. Es ist auch eine Frage der persönlichen Haltung und Denkweise.

So ist es oft möglich, Anforderungen so zu paketieren, dass in einem Projekt das Softwareprodukt in kleinere Realisierungseinheiten zerlegt wird. Jede dieser Einheiten kann dann in einem in sich funktionierenden Inkrement des Softwareprodukts realisiert werden. Beispielsweise wird in einem ersten Durchgang der Hauptfluss oder Durchstich aus Businesssicht realisiert, ohne Alternativen und Sonderfälle, dafür inklusive minimales GUI und Datenbankzugriff. Alternativflüsse werden anschliessend einer darauf folgenden Realisierungseinheit hinzugefügt. Nach jeder realisierten Einheit oder jedem Inkrement kann das Projekt auch ohne vorgesehenen

Meilenstein die Stakeholder oder das Business bereits um Feedback fragen und dies gleich mit aufnehmen. Auch ganz im Sinne eines evolutionären Prototyps (siehe IREB.de).

6. Erfahrungen weitergeben: Qualitätstipps

All die in der Projektarbeit gesammelten Erfahrungen in Bezug auf Qualitätsverbesserung können ihre Wirkung im Unternehmen erst entfalten, wenn sie für andere Mitarbeiter und Projekte nutzbar gemacht wird. Man erlebt oft, dass nach Abschluss des Projekts zwar eine Erfolgskontrolle und Feedbackrunde stattfindet (Stichwort Lessons Learned), die Erkenntnisse daraus jedoch nicht weiter genutzt werden.

Bewährt hat sich in der Praxis, die in den Projekten gemachten Erfahrungen an einer Stelle zu sammeln und gemeinsam mit einem Expertenzirkel aus Businessanalysten zu Best Practices oder Qualitätstipps zu verarbeiten. Diese Tipps müssen für Laien fassbar und einfach umsetzbar sein. Dabei gilt: Nur die maximal zehn wesentlichsten Tipps, auf einer A4-Seite kurz erklärt, zusammentragen. Die Tipps ändern sich im Laufe der Zeit ganz nach dem Prinzip der lernenden Organisation. Mit höherer Maturität werden auch die Tipps spezifischer. Wichtig ist, dass die pragmatischen Tipps in dieselbe Richtung weisen wie die allenfalls vorhandenen methodische Vorgaben.

Ziel der Tipps ist, dass Mitarbeiter mit weniger Erfahrung von den Fehlern der Kolleginnen und Kollegen profitieren und ihre Erfahrung selbst darauf aufbauen können. Die Voraussetzung dafür ist simpel, aber dennoch nicht trivial zu erreichen: Es braucht eine wertschätzende Fehler- und offene Kommunikationskultur, damit die Mitarbeiter auch offen und ehrlich über ihre gesammelten Erfahrungen berichten.

«Bist du in Eile, gehe langsam»

Um die Qualität von Anforderungen in einem IT-Projekt zu verbessern, und damit letztlich Zeit und Kosten zu sparen, liegt es oft nicht am fehlenden Werkzeug oder an unvollständigen Methoden, sondern an der Offenheit, bestehende Prozesse, Anforderungen und Funktionalitäten zu hinterfragen. Dazu braucht es neben den organisatorischen Voraussetzungen die präzise und emphatische Denkarbeit des Businessanalysten. Gerade in hektischen Projektphasen verfallen wir oft einem Aktivismus und nehmen Anforderungen auf, ohne sie wirklich zu hinterfragen, und kommen vor lauter Tun nicht mehr vom Fleck. Zu diesem Zeitpunkt ist es gut, sich auf eine alte Weisheit aus dem Zen-Buddhismus zu besinnen: «Bist du in Eile, gehe langsam». Also: kühlen Kopf bewahren, denn Qualität beginnt im Kopf.