

Vermeidung von Fallstricken bei der Anforderungsermittlung

(©2014 Peter Hartauer)
info@peterhartauer.de

Bei der Erhebung und Sammlung von Anforderungen für ein Produkt gibt es eine große Anzahl an Fallstricken, welche das effektive und vollständige Arbeiten behindern. Hier möchte ich Ihnen ein paar Einblicke in die Fallstricke geben.

Die Einkaufswagenmentalität

Diese Technik ist auch nach unzähligen Projekten allgegenwärtig. Sie kennzeichnet sich dadurch, dass jedes Projekt unter einer falschen Erwartungshaltung eines oder mehrerer Interessenvertreter leidet. Das Sammeln von Anforderungen verläuft wie ein Einkauf im Supermarkt.

Beispiel:

RE: Benötigen wir eine Lokalisierung?

IV: Hört sich gut an. Wir sollten uns nicht vor neuen Märkten sperren.

RE: Haben Sie sich Gedanken zur Sicherheit gemacht?

IV: Sicherlich! Unser Produkt sollte sicher sein.

RE: Was erwarten Sie an Zuverlässigkeit?

IV: Auf jeden Fall rund um die Uhr! Kein Produktionsausfall! DAS ist es, was wir brauchen um die Mitbewerber auszustechen.

Und so weiter und so fort. Man kann einem Interessenvertreter alles als Lösung verkaufen. Wenn ihm eine Lösung zu Beseitigung der Rohölabhängigkeit gegeben wird, so will er dies haben wollen. Aber das alles kostet Geld und die Interessenvertreter müssen verstehen, dass jeder ihrer Wünsche auch Kosten nach sich zieht.

Der Fallstrick ist hier ganz klar das Anbieten von Lösungen wie auf einer Einkaufsliste oder einer Speisekarte. Des Rätsels Lösung ist die Lösungen bei der Anforderungsanalyse nicht vorweg zu nehmen und dem Interessenvertreter verständlich zu machen, dann jeder seiner Wünsche Auswirkungen auf das Produkt haben wird und hiermit auch Mehrkosten verbunden sind.

Die „Das-ist-mit-zu-technisch“-Einstellung

System- und Architektur Anforderungen werden häufig durch Spezialisten und Techniker geschrieben. Dadurch hängt ihnen eine hohe Fachbegrifflichkeit und Technikverliebtheit nach, welche sich auch andere Interessenvertreter als abschreckend herausstellt. Somit lassen diese Interessenvertreter

Anforderungen, welche das System oder die Umgebung betreffen, gerne weg, verschweigen diese oder wehren sich aktiv dagegen diese Anforderungen zu spezifizieren.

Ein weiterer Hinderungsgrund für das erfolgreiche Sammeln solcher Anforderungen ist die geringe Möglichkeit zur Kommunikation. Hier treffen häufig zwei Welten, die technische und die fachliche, aufeinander. Ein guter Requirements Engineer hat nun die Aufgabe diese beiden Gruppen zusammen zu bringen. Er hat dafür zu sorgen, dass nicht nur Use-Cases für die domänen-spezifischen Anforderungen erstellt werden, sondern auch alle anderen Anforderungsdokumente beachtet werden.

Alles in Allem geht es wieder nur um das Eine: Der Interessenvertreter muss motiviert werden seine Anforderungen vollständig abzugeben. Dazu hilft ein Fragebogen oder eine Checkliste. Der Interessenvertreter muss verstehen, dass sich die Zeit, die er sich für das Beantworten der Fragen nimmt, ihm selbst wieder zu Gute kommt. In Form eines besseren, weniger fehlerbehafteten und qualitativ hochwertigeren Produktes. Der leichteste Weg dies zu beweisen sind Beispiele mit fehlgeschlagenen Projekten, bei denen nicht alle Fragen der Requirement Engineers vollständig beantwortet wurden.

Der „Alle-Anforderungen-sind-gleich“-Irrglaube

Nichts ist einfacher als allen Anforderungen dieselbe Priorität zu geben. Es ist der einfachste Unterschied, nämlich keiner. Alle Anforderungen sind gleich wichtig oder unwichtig. Wobei – in der Praxis werden alle Projekte, welche in diesen Irrglauben verfallen, alle Anforderungen als hochwichtig und hochpriorisiert einstufen. Ohne Ausnahme!

Doch warum sollte man priorisieren? Warum sind nicht doch alle Anforderungen wichtig? Schließlich zahlt der Kunde Geld dafür. Warum muss man dann noch Unterteilungen bei den Anforderungen machen?

Nehmen wir als Beispiel Anforderungen an die Architektur. Diese müssen priorisiert werden um den Business Architect, oder auch jedem anderen, zu zeigen, welches die wichtigsten Komponenten für ein System sind. Es können im Notfall keine Abstriche im Design gemacht werden, wenn alle Anforderungen in dieselbe Kategorie eingeordnet wurden. Wo soll nun der Rotstift angesetzt werden, wenn es darum geht das Projekt durch Verkleinerung des Funktionsumfangs noch zu retten?

Sollten Sie bei der Priorisierung Probleme bekommen und können sich nicht entscheiden, wie Sie diese vornehmen sollen, dann greifen Sie zu einem Trick: Vergleichen Sie immer zwei Anforderungen miteinander. Auf diese Weise steigen die wichtigsten Anforderungen an die ersten Stellen Ihrer Liste.

Das „Stell-es-in-das-Regal“-Problem

Einige Firmen sind sorgsam darauf bedacht, dass alle Anforderungen gesammelt und priorisiert werden, weil ein Berater oder Analyst es ihnen so erklärt hat. Doch nach einiger Zeit werden Anforderungen in ein Regal geschoben und danach nie wieder verwendet.

Das ist ein großes Problem, denn man erstellt sich eine ungeheure Menge an Anforderungen, welche nur noch sehr schwer zu verwalten sind. Auch geht der Wert der Anforderungen mit der Zeit zunichte, da kein Stakeholder genau verstehen kann, warum noch Anforderungen erhoben werden und welchen spezifischen Mehrwert diese haben.

Genau hier ist es Aufgabe der Requirement Engineers regelnd einzugreifen. Es muss den Stakeholdern verständlich gemacht werden, warum ihre Anforderungen wichtig sind und wie diese im weiteren Entwicklungsprozess verwendet werden.

Das „Anforderungen-können-nicht-gemessen-werden“-Syndrom

Dies kann ein Fallstrick sowohl für funktionale als auch für nicht-funktionale Anforderungen sein. Hierbei geht es in erster Linie um die Testbarkeit. Keine Anforderung ist sinnvoll, wenn sie nicht durch eine objektive Kontrolle überprüft werden kann.

Somit ist es wichtig, dass eine Anforderung zwei Merkmale besitzt: Eindeutigkeit und Messbarkeit.

Diese beiden Attribute sorgen dafür, dass eine Anforderung von jedem verstanden wird und auch durch Dritte getestet werden kann. Somit zählen diese Eigenschaften zu den Kriterien für gute Anforderungen.

Hier ein Beispiel für eine Anforderung:

„Das System muss ein modernes Benutzerinterface besitzen.“

Diese Anforderung trifft keines der beiden geforderten Merkmale. So ist der Ausdruck „modern“ im höchsten Maß subjektiv und kann von jeder Person unterschiedlich verstanden werden. Des Weiteren ist dies für jeden Tester ein wahrer Alptraum. Wann ist der Testfall für diese Anforderung nun bestanden? Wie sollte „modern“ gemessen werden?

Die „Nicht-genug-Zeit“-Beschwerde

Anforderungen zu entwickeln und diese auf hohem qualitativem Niveau zu spezifizieren ist eine schwierige, anspruchsvolle und komplexe Tätigkeit. Man kann dies nicht in kurzer Zeit erledigen. Oft schrecken Stakeholder davor zurück einen angemessenen Zeitraum dafür zur Verfügung zu stellen oder werden während der Erhebungsphase ungeduldig.

Der Grund liegt nahe, wenn man nur versucht sich einmal in diese Menschen hineinzusetzen. Sie wollen am Ende ihr Produkt haben. Während der Phase der Anforderungserhebung und Spezifikation entsteht nichts, was sie später verwenden können.

Vergleichen Sie es mit einem Hausbau. Die Phase in der die Baupläne erstellt werden zieht sich oft lange hin. Es werden Berechnungen durchgeführt und immer wieder Anpassungen vorgenommen. Sie wollen sich ja schließlich in ihrem Haus wohl und sicher fühlen. Da gehört es sich, dass die Küche nicht im Keller steht, während sich das Esszimmer im Obergeschoss befindet. Sie wissen ja, dass sobald der erste Stein liegt eine Änderung sehr teuer wird, wenn sie denn überhaupt möglich ist.

Erinnern Sie also auch ihre Stakeholder daran, dass die Anforderungen die Basis, quasi das Fundament, ihres Produktes sind und sie genügend Zeit dafür brauchen um ein stabiles und sicheres System entwickeln zu können. Auch Software lässt sich nicht beliebig verändern. Verhindern sie, dass Stakeholder aus Ungeduld ihre Ergebnisse sabotieren, indem sie direkt zu Entwicklern gehen oder andere Methoden ergreifen um vermeintlich schneller ans Ziel zu gelangen.

Die „Alle-Stakeholder-sind-gleich“-Fehlinterpretation

Während der Erhebung der Anforderungen bekommen Sie Eingaben von vielen Stakeholdern. Ihre Aufgabe ist es nun die richtigen Fragen an die richtige Person zu stellen. Und das erweist sich oft schwieriger als erwartet.

Zum Beispiel ist der Produktverantwortliche unter Umständen der richtige Ansprechpartner, wenn es um Fragen der Lizenzierung der Software oder Teilen des Systems geht. Aber fragen Sie ihn auch, wenn es sich um Benutzbarkeit dreht? Hier sollten Sie Ihre Anforderungen besser bei den Endbenutzern suchen.

Hierfür sollten Sie beachten, dass Sie sich vorher schon Gedanken über die einzelnen Stakeholder gemacht haben. Wer ist wichtig für das Produkt? Wer weiß mehr über bestimmte Aspekte?

Vergessen Sie aber bei aller Analyse der einzelnen Gruppen nicht, dass es noch weitere geben kann, die Interesse an ihrem Produkt haben könnten. Vergessene Stakeholder sind vergessene Anforderungen!

Die „Zu-Allgemein“-Tendenz

Ein Fragebogen zur Erhebung funktionaler und nicht-funktionaler Anforderungen sollte wie andere Erhebungstechniken behandelt werden. Der große Unterschied in einem Fragebogen besteht darin, dass man sich auf bestimmte Aspekte konzentriert.

Nehmen wir als Beispiel Anforderungen für die Architektur. Diese Anforderungen werden oft sehr allgemein gehalten. Es ist häufig auch schwierig sich genau festzulegen.

„Das System soll eine Antwortzeit von weniger als drei Sekunden haben.“

Diese Anforderung ist messbar, doch haben Sie sich überlegt, was davon alles betroffen ist? Jede Transaktion soll eine Antwort nach spätestens 3 Sekunden liefern. Das trifft auf einige ausgewählte Transaktionen zu, doch bei weitem nicht auf alle. Manche Transaktionen werden signifikant länger benötigen. Wenn Sie diese Anforderung nicht näher einschränken, so kann es passieren, dass Entwickler beginnen wirklich jede Transaktion innerhalb von 3 Sekunden zu einer Antwort zu bewegen. Hier besteht die Gefahr des „Over-Engineering“ um die Anforderungen zu erfüllen, die für eine spezielle Transaktion gar nicht vorgesehen waren.