



Analytiker trifft Architekt

Von Missverständnissen und guten Kooperationen

Carsten Pflug, Dirk Schüpferling

In jedem Entwicklungsprojekt kommt irgendwann der Punkt, an dem die Anforderungen an die Architekten übergeben werden. Und genau an diesem Schnittpunkt zwischen Analyse und Architektur können verschiedene Probleme entstehen, die den Projekterfolg negativ beeinflussen. Viele Faktoren bedingen, dass der Schnittpunkt zwischen diesen beiden „Welten“ nicht vollkommen konfliktfrei ist. In diesem Artikel werden wir einige dieser Probleme identifizieren und Anregungen geben, um den Übergang zwischen Analyse und Architektur so problemlos wie möglich zu gestalten.

Gründe für schlechte Kommunikation zwischen Analytiker und Architekt

► „Dafür können wir nichts, das haben die von der Analyse falsch spezifiziert.“ Oder „Es war ja wohl klar, dass bis zu 500 Personen gleichzeitig am System arbeiten. Was haben die Architekten hier bloß geplant.“ Solche oder so ähnliche Entschuldigungen bekommt man in vielen Projekten zu hören. Sie veranschaulichen den „Graben“ zwischen Analyse und Architektur recht deutlich. Doch woher kommt dieser Graben?

Es gibt viele Gründe und eine Vielzahl von Faktoren, die das Zusammenspiel zwischen Analyse und Architektur negativ beeinflussen. Wobei gerade diese Schnittstelle besonders wichtig ist, denn der Analytiker liefert die Anforderungen, die der Architekt und sein Team umsetzen müssen. Ein reibungsloses Zusammenspiel ist also für den Erfolg unerlässlich. Wir möchten an dieser Stelle die wichtigsten Faktoren kurz erläutern.



Abb. 1: Graben zwischen Analyse und Architektur

Unterschiedliche Ziele für die dokumentierten Anforderungen

Ein Problem ist, dass beide Parteien mit den dokumentierten Anforderungen unterschiedliche Ziele verfolgen. Ziel des Analytikers ist es, eine Abnahme/Freigabe durch den Kunden zu erhalten. Hier wird teils absichtlich noch Spielraum gelassen, damit die Freigabe leichter erlangt wird, oder weil man manches noch nicht vollständig durchdacht hat.

Der Architekt hingegen versucht, möglichst genaue Vorgaben zu bekommen, welche Systemfunktionen, wie (nichtfunktionale Anforderungen, Datenanforderungen, ...) realisiert werden sollen, denn diese Vorgaben bestimmen seine Arbeit. Das heißt, der absichtlich oder auch versehentlich gelassene Spielraum muss nun verschwinden. Zumeist geschieht dies, indem Annahmen seitens des Architekten getroffen werden. Dadurch



haben wir eine Kluft zwischen dem Wissen des Architekten über das System und dem Wissen des Analytikers.

Fazit: Analytiker und Architekt haben häufig unterschiedliche Erwartungen an das Anforderungsdokument.

Nichtfunktionale Anforderungen werden in der Analyse kaum oder zu ungenau dokumentiert

In den nichtfunktionalen Anforderungen stecken viele notwendige Informationen für die Architektur. Somit sind sie für den Architekten besonders wichtig. Leider werden aber diese Anforderungen in der Analyse meist sehr stiefmütterlich behandelt, oder zu ungenau spezifiziert. Die Realisierung dieser Forderungen wird dann überwiegend erhofft. Nichtfunktionale Anforderungen werden oft implizit mit den funktionalen Anforderungen umgesetzt. Denn jede Funktion weist z. B. ein Zeitverhalten oder eine Robustheit auf. Aber ohne deren Vorgaben ist das Risiko, dass das System im Nachhinein nicht den Kundenwünschen entspricht, besonders groß.

Unterschiedliche Sprache

Nein, wir wollen damit nicht behaupten, dass in der Analyse Deutsch und in der Architektur nur Russisch gesprochen wird. Vielmehr bedingen die unterschiedliche Ausbildung, gemachte Erfahrungen und die unterschiedlichen Welten, in denen die Personen unterwegs sind, dass derselbe Sachverhalt häufig mit verschiedenen Begriffen umschrieben wird. Hier spielen auch die jeweils andere Sichtweise auf den Sachverhalt bzw. die unterschiedlichen Ziele, welche von den beiden Parteien verfolgt werden, eine große Rolle. Diese uneinheitliche Sprache ist häufig der Grund für viele Missverständnisse zwischen Analyse und Architektur. Das besonders Kritische daran ist, dass diese Missverständnisse häufig gar nicht bemerkt werden.



Abb. 2: Unterschiedliche Sprachen



Unterschiedliche Sichten auf das System

Der Analytiker betrachtet das System als Ganzes und interessiert sich überwiegend dafür, wie Geschäftsprozesse mit dem System durchgeführt werden können. Die Architektur sieht das System als eine Komposition mehrerer Einzelteile, die in enger Kooperation übergeordnete Prozesse realisieren müssen. Der Zusammenhang auf der Ebene von Geschäftsprozessen ist für den Architekten nicht immer bekannt. Dadurch kann es zu Fehlinterpretationen und falschen Annahmen zu den Anforderungen auf der Seite der Architektur kommen.

Ungenauere Anforderungen

Dann gibt es auch noch das allzu bekannte Problem, dass aufgrund von zu wenig Zeit, zu wenig Ressourcen, zu wenig fachlicher Kenntnisse des Analytikers oder schlicht Unerfahrenheit mit dem Einsatz der passenden Requirement-Engineering-Methoden die in der Analyse entstandenen Anforderungen von zu schlechter Qualität sind. Der Designer hat dann die unruhliche Aufgabe, diese Anforderungen zu interpretieren, in der Hoffnung, dass er die richtige Interpretation wählt.

Informale Analyseergebnisse treffen auf formale Designbeschreibungen

Oft ist die Dokumentation der Analyseergebnisse eine Mischung aus einfachen formalen Modellen, wie sie die UML anbietet (Use-Case- und Aktivitätsdiagramme), und Prosaanforderungen, welche in Detailgrad und Umfang stark schwanken können. Der Architekt verwendet hauptsächlich streng formale Modelle. Die Übersetzung der Analyseergebnisse in eine Designbeschreibung birgt unter anderem das Risiko der Fehlinterpretation bei der Überführung und Ergänzung um architekturrelevante Informationen bzw. Verfeinerung der Anforderungen.

Zu viele Details in der Analyse

Das Problem der Detaillierung von Anforderungen kann aber auch in die andere Richtung gehen. Nämlich dann, wenn in der Analyse zu viele Details beschrieben werden, ohne deren Realisierungsmöglichkeiten abzuklären. Oft fallen dann andere Anforderungen unter den Tisch, da keine Zeit mehr zur Verfügung stand, diese zu analysieren. Ursache dieses Problems ist häufig die nicht vorhandene Absprache zwischen Analyse und Design und eine fehlende Aufgabenverteilung.

Räumliche und organisatorische Trennung

Nicht zu vergessen ist die räumliche Trennung zwischen den Analytikern und den Designern. Dies ist auch ein Faktor, der die Kommunikation zwischen den beiden „Lagern“ erschwert. Je örtlich verteilter Personen voneinander arbeiten, desto umständlicher und damit seltener wird kommuniziert und die Kommunikationsart weicht immer mehr auf indirektere Wege, meist in schriftliche Form, aus. Dadurch werden Missverständnisse oft erst spät erkannt und deren Beseitigung verlangt Ressourcen, welche eigentlich anderweitig benötigt werden.

Auch die organisatorische Trennung ist nicht zu verachten. Der Begriff „Lager“ in dem oberen Absatz ist hier gezielt eingesetzt worden. Probleme in einem Projekt werden gerne den „Anderen“ zugeschoben.

Lösungsansätze

Nicht alle Probleme sind an dieser Stelle aufgezeigt worden. Zum Beispiel ist die sozial-menschliche Komponente hier vollkommen ausgeblendet worden. Der wichtigste Faktor für

die Wahl der Gründe für die schlechte Kommunikation zwischen Analyse und Design ist die Möglichkeit, die Schwierigkeiten mit den hier vorgestellten Lösungsansätzen in den Griff zu bekommen.

Architekt früher in die Prozesse einbinden

Bereits in der Analysephase sollte der Architekt zu ausgesuchten Themen hinzugezogen werden. Besonders in der Endphase der Analyse bietet es sich an, dass Analytiker und Architekt gemeinsam an der Fertigstellung der Spezifikation arbeiten, um somit das Verständnis des Architekten für das Gesamtbild (System und Geschäftsprozessebene) zu verbessern.



Abb. 3: Teamwork

Ein voneinander Lernen, welche

Informationen

braucht der Analytiker bzw. der Architekt zu welchem Zeitpunkt und in welcher Form, verbessert die Zusammenarbeit für zukünftige Projekte.

Ist dieses gemeinsame Arbeiten nicht möglich, sollte zumindest ein Review der Spezifikation durch den Architekten stattfinden. Hierbei muss ihm die Zeit gegeben werden, sich intensiv mit der Dokumentation zu beschäftigen, um die für ihn nicht eindeutig oder ausreichend beschriebenen Punkte mit dem Analytiker zu klären. Eine reine Übergabe der Anforderungen, wir nennen dies auch „Spezifikation über den Zaun werfen“, ist auf keinen Fall ausreichend.

Gemeinsam ein durchgängiges Vorgehen bei der Systementwicklung entwickeln/gemeinsame Dokumentationsformen

Für ein gemeinsames Vorgehen müssen die gelebten Prozesse der Analyse und der Architektur beleuchtet werden. Dabei müssen folgende Fragen beantwortet werden: Wo werden ähnliche Aufgaben mit anderen Methoden oder Techniken durchgeführt? Wer übernimmt welche Aufgaben in der Systementwicklung? Gibt es Redundanzen? Wo kann man Verbesserungspotenzial finden? Wer wird wann in die Prozesse mit einbezogen? Kommen die Informationen zum richtigen Zeitpunkt in der richtigen Form? Und ganz wichtig, wer braucht welche Art von Information in welcher Form?

Mit den Ergebnissen aus dieser Betrachtung wird ein gemeinsames Vorgehen von Analyse und Design abgestimmt. Dabei ist es wichtig, dass alle Parteien in diesem Abstimmungsprozess mit einbezogen werden. Keine Partei sollte den Anderen deren Aufgaben vorgeben. Beim Festlegen des gemeinsamen Vorgehens werden auch die Ziele, die zu erstellenden Artefakte und die jeweiligen Stakeholder bestimmt, dokumentiert und allen Beteiligten klar kommuniziert.

Nicht nur das Anpassen der Prozesse ist ein Garant für eine gute Zusammenarbeit, auch die Dokumentationsformen haben darauf Einfluss. Idealerweise werden die Dokumentationsformen (natürlichsprachlich, UML-Diagramme usw.) in der Analyse und dem Design vereinheitlicht. So finden sich alle Parteien in den Dokumenten leichter zurecht, haben weniger Probleme, die Bereiche der jeweils anderen Parteien zu verstehen, und es bieten sich auch noch viele Möglichkeiten, eine vernünftige Nachvollziehbarkeit (Traceability) zwischen den

Artefakten aufzubauen. Auch die Gefahr einer redundanten Dokumentation gleicher Sachverhalte durch unterschiedliche Formen wird minimiert.

Für die Vereinheitlichung muss vor allem geprüft werden, welche Notationsformen aus welchen Gründen (Vorteile/Nachteile) in welcher Phase eingesetzt werden. Die gefundenen Formen werden normiert und für alle Projektbeteiligten dokumentiert.

Einheitliche Sprache verwenden (z. B. Glossar) oder Übersetzungsschnittstelle entwickeln

In den meisten Fällen wird die Kooperation zwischen Analyse und Design durch eine gemeinsame Sprache stark verbessert. Die einfachste Form, eine gemeinsame Sprache zu finden, ist die Erstellung eines gemeinsamen Glossars. In diesem werden nicht nur Fachbegriffe, sondern auch Prozesswörter (Verben) und alltägliche Begriffe, welche zwar von jedem Leser verstanden werden, aber leider nicht immer einheitlich bzw. deren Verwendung im speziellen Projektkontext anders als gewohnt ist, definiert. Dieses Glossar ist verbindlich für alle Projektbeteiligten.

Bekommt man nicht alle Beteiligten auf einen gemeinsamen Nenner, dann sollte zumindest eine Übersetzungsliste für die unterschiedlichen Begriffe verwendet werden. Diese Liste beschreibt die unterschiedlich verwendeten Begriffe und deren Entsprechung in der anderen Disziplin. Prinzipiell sollte man mit dieser Methode aber sehr vorsichtig sein, da sie ein gemeinsames Verständnis vorgaukelt. Bei den Beteiligten der Systementwicklung etabliert sich sehr schnell die falsche Annahme, dass jeder den anderen versteht.

Qualitygates im Vorfeld

Bevor die Ergebnisse der Analyse an die Architektur übergeben werden, müssen diese ein Qualitygate durchlaufen. In diesem werden die Anforderungen auf Herz und Nieren geprüft. Nur wenn die Anforderungen den festgelegten Qualitätskriterien genügen, dürfen diese an die Architektur übergeben werden.

Bezüglich der Zusammenarbeit zwischen Analytiker und Architekt hat dieses Vorgehen allerdings einen Nachteil. Erst beim Durchlaufen des Qualitygates und sogar manchmal erst danach setzt sich der Architekt intensiv mit den Anforderungen auseinander. Dies ist allerdings ein sehr später Zeitpunkt in der Systementwicklung.

Daher ist es sehr empfehlenswert, bereits in der laufenden Analyse Qualitygates einzuplanen. Diese Qualitygates müssen nicht den Umfang und Aufwand

des Abschluss-Qualitygates haben. Anstatt einer Inspektion wird hier eher ein Walkthrough als Prüftechnik zum Einsatz kommen. Auch die Anzahl der einzubeziehenden Stakeholder kann hier geringer sein. Oberstes Ziel muss es dabei sein, eine frühe Klärung, ob ein gemeinsames Verständnis der Anforderungen und des Vorgehens vorliegt, zu erlangen bzw. welche Maßnahmen getroffen werden müssen, um dies zu erreichen.



Abb. 4: Review

Fazit

Kommunikationsschwierigkeiten zwischen Analyse und Design lassen sich nicht von heute auf morgen beseitigen. Denn dies ist ein längerer Prozess, der von allen Parteien mitgetragen werden muss. Der Aufwand, der dafür betrieben werden muss, zahlt sich aber am Ende aus. Die Zusammenarbeit wird sich im Laufe der Zeit merklich verbessern. Dadurch kann das ganze Team effizienter arbeiten, was sich auf Dauer auf das benötigte Budget und die benötigte Zeit für ein Projekt positiv auswirkt.

Man darf aber nicht vergessen, dass die hier beschriebenen Lösungsansätze keine einmaligen Tätigkeiten beschreiben. Vielmehr muss die Art der Zusammenarbeit in regelmäßigen Abständen hinterfragt und falls nötig angepasst werden. Ein einmaliges Zusammenraufen führt aus kurzfristiger Sicht zu einer Verbesserung, auf lange Sicht aber werden die Schwierigkeiten immer wieder aufflackern.

Zu guter Letzt sollte auf jeden Fall immer der Teamgedanke bei allen Beteiligten gefördert werden, denn eine Systementwicklung ist nur in einem funktionierenden Team erfolgreich durchführbar.



Abb. 5: Gemeinsam ans Ziel



Carsten Pflug ist seit 2002 für die SOPHIST GmbH als Berater und Trainer unterwegs. Thematisch ist er dabei sehr vielseitig einsetzbar. Sei es die Erhebung, Analyse und Dokumentation natürlichsprachiger Anforderungen, die Entwicklung einer Systemarchitektur, die Etablierung und Verbesserung von Requirements-Management-Methoden, der Einsatz der Objektorientierung oder das umfangreiche Gebiet der Modellierungstechniken.
E-Mail: Carsten.Pflug@SOPHIST.de.

Dirk Schüpferling ist bereits seit 2001 ein SOPHIST und hat in den letzten Jahren die Erkenntnis gewonnen, dass Kommunikation meist der Schlüssel zur (Kunden-)Zufriedenheit ist. Aber ebenso Faulheit und Besserwisserie oder, wie der Fachmann sagen würde, „Wiederverwendung“ und „Aufzeigen von Verbesserungspotenzial“ sind Dinge, bei denen Dirk viel Spaß haben kann. Er forscht auf dem Gebiet der Dokumentationstechniken und löst hier Problemstellungen aus der Praxis, indem er Neuerungen und Anpassungen an etablierten Tools und Dokumentationsarten vornimmt.
E-Mail: Dirk.Schuepferling@SOPHIST.de.