

Funktionale und nicht funktionale Anforderungen

Der Unterschied zwischen funktionalen und nicht funktionalen Anforderungen ist nicht immer absolut klar zu definieren. Trotzdem in vielen Situationen kann man die beiden Arten der Anforderungen deutlich auseinander halten.

Die funktionalen Anforderungen definieren die Funktionalitäten, die Subsysteme, die Module, die zu realisieren sind. Anders gesagt, die funktionale Anforderungen schreiben vor, **was** zu tun ist.

Die nicht funktionale Anforderungen definieren die Qualitäten der einzelnen Module oder des ganzen Systems. Also, sie schreiben vor, **wie** es funktionieren soll.

Betrachten wir die beiden Anforderungen an einem Beispiel "Online Kontoführung in einer Bank".

Folgende funktionale Anforderung wird an das System (Modul) gestellt:

- Alle Buchungen müssen in einem Fenster angezeigt werden. Die Navigationswerkzeuge müssen dem Kunde zur Verfügung stehen.

Diese Anforderung beschreibt wirklich eine Funktionalität: es erscheint ein Fenster, in dem Fenster sind alle Buchungen aufgelistet, und man kann in diesen Buchungen navigieren/auswählen.

Diese Formulierung passt ganz gut in der Phase "Soll-Konzept" und auch vielleicht in der Phase "Pflichtenheft", weil hier im Groben klar ist, was von dem Modul zu erwarten ist.

Sie ist aber nicht deutlich genug für die Phase "Entwurf" formuliert: Was heißt "alle Buchungen" - die letzten 10 oder wirklich alle für alle Jahre? Wie soll es navigiert werden - nur mit der Maus oder auch mit den Cursor-Tasten?

Die 10 letzten Buchungen sind eindeutig zu wenig für viele Kunden, und alle Buchungen für alle Jahre wird kaum jemand brauchen, abgesehen davon, dass es relativ kompliziert und kostenspielig ist.

Diese Anforderung sollte dann präzisiert aussehen:

- Die letzten 30 Buchungen müssen in einem Fenster angezeigt werden. Der Kunde muss eine Option haben, die Anzahl der Buchungen selbst festzulegen, aber nicht mehr als 200. Die Navigation mit der Maus und mit der Tastatur muss dem Kunde gewährleistet werden.

So eine Formulierung verhindert schon viele Unklarheiten in der Phase "Realisierung". Sie passt selbstverständlich nicht in die früheren Phasen.

Nicht funktionale Anforderungen

Für dasselbe Beispiel wird folgende nicht funktionale Anforderung definiert:

- Das Buchungsfenster muss sicher sein.

Hier wird das Verhalten des Buchungsfensters beschrieben, also wie es funktionieren soll, also - die Qualität des Fensters. Was dabei passieren soll, welche Module es gewährleisten - das wird bewusst draußen gestellt.

Die Sicherheitsmaßnahmen werden selbstverständlich gefordert - sichere Anmeldung, Zertifikate, PIN/TAN-Verwaltung. Aber hier an dieser Stelle ist die Erwähnung von Sicherheit genau für dieses Buchungsfenster schon sehr wichtig. Ohne dieser Anforderung bleibt dieses Fenster ewig auf dem Bildschirm, und eine andere Person in Abwesenheit des Kunde kann auf seine Buchungen unerlaubten Zugriff haben.

Nicht funktionale Anforderungen

Somit soll diese nicht funktionale Anforderung in späteren Phasen etwa so umformuliert werden:

- Das Buchungsfenster muss sicher sein, d.h. wenn keine Aktivitäten seitens Kunde innerhalb 10 Minuten passieren, dann muss das Fenster automatisch schließen und die Verbindung zur Bank getrennt werden.

Die anderen Beispiele für nicht funktionale Anforderungen:

- Zuverlässigkeit/Verfügbarkeit des Systems (99,8 % im Jahr).
- Zugriffszeit/Aufrufzeit (500 ms).
- Antwortzeit (5 s)
- Schulungsdauer (3 Tage)

Nicht funktionale Anforderungen

Die nicht funktionale Anforderungen beschreiben die Eigenschaften des Systems oder der einzelnen Module. Man kann sie folgendermaßen unterteilen:

- Produkt-spezifische Anforderungen. Sie beziehen sich auf das Verhalten des Systems - Sicherheit, Zuverlässigkeit, Effizienz, Benutzbarkeit, Portierbarkeit.
- Unternehmen-spezifische Anforderungen. Hier sind die internen Standards, Richtlinien, Vorschriften, Vorgehensweise, Dokumentation, Umsetzung gemeint.
- Externe Anforderungen. Das sind die gesetzlichen, politischen, ethischen, interkommunikativen Anforderungen, sowie Datenschutzvorlagen und Sicherheitsrichtlinien.

Nicht funktionale Anforderungen

Die nicht funktionalen Anforderungen, wie schon gesagt, haben einen qualitativen Charakter, sie sind deshalb schwer zu überprüfen oder zu testen, wenn das Softwareprodukt in Betrieb übergeben wird. Man muss schon in der Phase "Pflichtenheft" die quantitativen Metriken einsetzen, um die Eigenschaften beim Testen messen und erkennen, ob die entsprechenden nicht funktionalen Anforderungen erfüllt sind. Beispiele für solche Metriken sind:

- KiB, MiB, GiB, TiB
- s, ms, Stunden, Tage
- Transaktionen pro Sekunde
- Wahrscheinlichkeit des Fehlers, Quote für Fehler

Problembereichsanforderungen

Das sind die funktionalen und nicht funktionalen Anforderungen, die aus dem Anwendungsbereich hervorgehen, also das sind ganz spezielle Anforderungen, die oft mit der Sicherheit verbunden sind. In dem Pflichtenheft werden sie oft nicht hervorgehoben, sondern einfach zu den funktionalen oder nicht funktionalen Anforderungen zugeordnet.

Beispiel:

- Das Geld von dem aktuellen Konto darf nur auf ein bestimmtes Konto überwiesen werden, aber nicht auf jedes beliebige Konto.

Anforderungen: externe Sicht

Anforderungen des künftigen Benutzers werden in deren (fachlichen) Sprache formuliert, so dass sie für den künftigen Benutzer verständlich sind. Diese Anforderungen legen das allgemeine, externe Verhalten des Systems fest. Hier sollten keine detaillierten technischen Begriffe, keine programm-technischen Strukturen, keine formalen Notationen vorkommen. Es werden einfache Ausdrücke, triviale Tabelle und nicht anspruchsvolle Diagrammen verwendet.

Diese Darstellungsform verursacht aber auch viele Probleme:

- Mehrdeutigkeit
- Ungenauigkeit
- Verschmelzung von Anforderungen

Anforderungen: externe Sicht

Um diese Probleme zu vermeiden, lohnt sich folgende Regeln aufzustellen:

- Standardformat für die Anforderungen
- Standardausdrücke
- Hervorhebungen im Text

Anforderungen: interne Sicht

Die Sicht des Systemanalytikers erweitert die externe Sicht, indem es beschrieben wird, wie die Benutzeranforderungen seitens künftigen Systems im Ganzen zu implementieren sind und welche Betriebseinschränkungen zu berücksichtigen sind. Dabei muss auf eine genaue Beschreibung des Entwurfs verzichtet werden.

Diese Sicht stellt vielmehr eine komplette und widerspruchsfreie Spezifikation des gesamten Systems und dient somit als Grundlage für den Vertrag zwischen Auftraggeber und Auftragnehmer.

Hier kann auch eine grobe Architektur des Systems dargestellt werden, d.h. die Aufteilung in Subsysteme, damit die Anforderungen besser zugeordnet werden können.

Am besten sollte eine zertifizierte Architektur (Peer-to-Peer, Client/Server, Serviceorientierte Architektur) verwendet werden.

Anforderungen: Sicht des Entwicklers

Für die Sicht sind die programm-technischen Strukturen, genaue Beschreibung von Funktionen und Datenflüssen, ausführliche Diagramme unentbehrlich. Diese Dokumente werden in der Phase "Entwurf" geschrieben.

Der Kunde bekommt diese Dokumente selten zu sehen, Entwurf ist oft die interne Angelegenheit des Auftragnehmers.