Nicht-funktionale Anforderungen

Michael Braun

Juristisches IT-Projektmanagement
Lehrstuhl für Programmierung und Softwaretechnik
Ludwig-Maximilians-Universität München
12. Januar 2016

Zusammenfassung Diese Ausführung gibt einen Überblick über die Charakteristiken *nicht-funktionaler Anforderungen* an Software Systeme und stellt eine übersichtliche Checkliste zur Anforderungsdefinition dar. Außerdem wird auf die Gewinnung von Anforderungen und die nötigen Qualitäten von geeigneten Metriken für Abnahmekriterien eingegangen.

Inhaltsverzeichnis

1 Einführung

In der Planungsphase eines Softwareprojektes befasst sich der Auftraggeber mit den Anforderungen, die das finale System zu erfüllen hat. Die resultierende Anforderungsdefinition beschreibt die Literatur als

> Ein Dokument, das eine vollständige Beschreibung dessen enthält, *was* ein Softwaresystem tun soll, aber nicht erklärt, *wie* es das tun soll. [1]

Die Anforderungen spiegeln die Wünsche der Systemnutzer wieder und bilden eine solides Grundgerüst für die Umsetzung eines Projektes. [2]

Diese Ausführung befasst sich mit der Unterscheidung von funktionalen und nicht-funktionalen Anforderungen, geht im Weiteren näher auf die verschiedenen Arten nicht-funktionaler Anforderungen ein und zeigt Möglichkeiten diese messbar zu machen.

1.1 Anforderungen

Eine minimale Anforderungsformulierung beinhaltet eine Beschreibung der geforderten Anforderung, eine Beschreibung des zu Grunde liegenden Problems oder Bedürfnisses und ein Abnahmekriterium, gegen das die Erfüllung der Anforderung objektiv geprüft werden kann. Außerdem empfiehlt es sich eine eindeutige ID zu vergeben und den Initiator der Anforderung festzuhalten. [3] Als Beispiel einer weit verbreiteten Formatvorlage dient uns eine sogenannte *Snow Card* aus dem Framework Volere. [4] (siehe Abbildung 1)

Des weiteren steht dem Verfasser der Aufbau von Anforderungen frei. Sinnvolle Bestandteile sind zum Beispiel Priorität, die aus Umsetzung bzw. Nichtumsetzung resultierende Änderung der Kundenzufriedenheit und -unzufriedenheit in numerischer Form, mögliche Konflikte mit anderen Anforderungen, Informationen zum Lebenszyklus, Status und betroffene Unternehmensprozesse. [schien]

1.2 Unterscheidung funktionale / nicht-funktionale Anforderungen

Anforderungen werden klassisch in zwei Kategorien unterteilt. Funktionale Anforderungen legen fest, *was* das System machen soll. [4] Ein Beispiel:

> Das System muss Emails versenden können.

Nicht-funktionale Anforderungen beschreiben *in welcher Qualität* das System die Leistung erbringen soll und welche Randbedingungen eingehalten werden müssen. [5] Ein Beispiel:

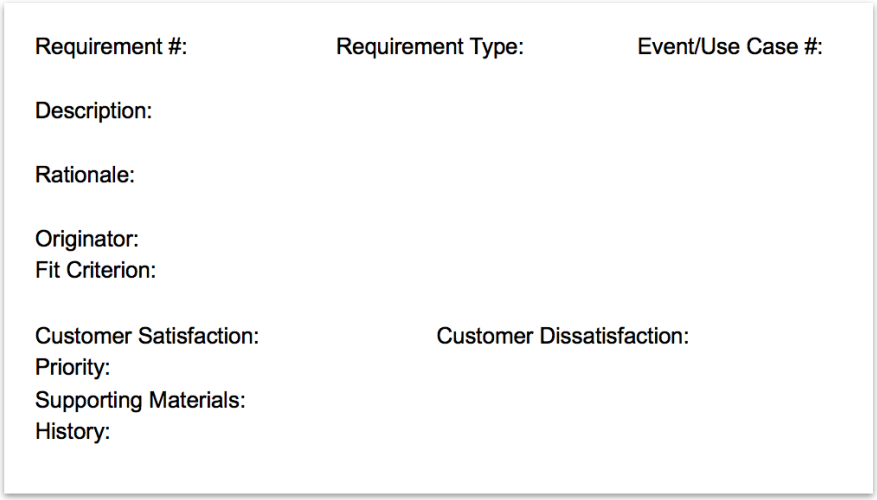

Abbildung 1. Snow Card zur Beschreibung einer Anforderung

Das System muss pro Sekunde mindestens 100 Emails verarbeiten können.

Definition: In der Literatur sind nicht-funktionale Anforderungen nicht eindeutig definiert, durch einen Vergleich der gängigsten Definitionen kann man sich jedoch einen guten Überblick verschaffen:

> Nicht-funktionale Anforderungen (non-functional requirements) – Anforderungen an die Umstände, unter denen die geforderte Funktionalität zu erbringen ist. [7]

Diese Definition wirkt auf den ersten Blick eindeutig, jedoch hängt die Entscheidung, ob es sich bei einer Sache um einen Umstand handelt vom Standpunkt des Betrachters und der resultierenden Formulierung ab. Als Beispiel dient hier die maximale Größe einer Email: Wenn die Anforderung besagt, dass eine Nachricht höchstens 10 MB groß sein darf, so ist das nach dieser Definition eine funktionale Anforderung. Formuliert man es jedoch so, dass Nachrichten nur verschickt werden können, wenn sie kleiner als 10 MB sind, dann ließe sich daraus eine Klassifizierung als nicht-funktionale Anforderung rechtfertigen. Aus diesem Grund gibt die gleiche Quelle eine weitere Definition an:

> Eine Anforderung bezeichnen wir als nicht-funktional, wenn das ihr zu Grunde liegende Bedürfnis eine nicht gegenständliche Eigenschaft ist. [7]

Diese Formulierung löst das Problem der Standpunktabhängigkeit, indem der Fokus auf das der Anforderung zu Grunde liegende Bedürfnis gelenkt wird. Jedoch wird damit auch die Entscheidung über die Gegenständlichkeit einer Eigenschaft eingeführt, was ebenfalls zu verschiedenen Interpretationen führen kann. In unserem Beispiel kann die maximale Größe einer Email als Einschränkung der Daten, also einer gegenständlichen Eigenschaft (siehe Abbildung 3) gesehen werden, oder aber die abstrakte Größe als zu Grunde liegendes, nicht gegenständliches Bedürfnis. Natürlich sind diese Fälle konstruiert und in der Realität selten, jedoch zeigen sie Ungenauigkeiten, die andere Ansätze durch ihre Allgemeinheit auszuschließen versuchen:

> Nicht funktionale Anforderungen sind Beschränkungen der durch das System angebotenen Dienste oder Funktionen. Dies schließt u.a. Zeitbeschränkungen, Beschränkungen des Entwicklungsprozesses, einzuhaltende Standards und Betriebsbedingungen ein. [8]

Am Beispiel der maximalen Emailgröße erkennen wir nach dieser Definition sofort eine nicht-funktionale Anforderung. Die Vorgabe, mindestens 100 Emails pro Sekunde verarbeiten zu können, welche bereits als Beispiel einer nicht-funktionalen Anforderung diente, fällt jedoch dieser Formulierung zum Opfer.

Diese Beispiele sollen nicht zur allgemeinen Verwirrung dienen, sondern in erster Linie zeigen, dass die Anwendung einer einzigen dieser Definitionen in vielen Fällen unzureichend ist. In Kombination ergeben die oben genannten Kriterien jedoch einen gutes Framework zur Unterscheidung von funktionalen und nicht-funktionalen Anforderungen.

2 Nicht-funktionale Anforderungen

Während funktionale Anforderungen bei den meisten Projekten definiert sind, werden nicht-funktionale Anforderungen oft vergessen, was im Laufe des Projekts zu Uneinigkeiten der Vertragspartner führen kann. Im Streitfall kann die Mängelhaftung aus § 633 Abs. 2 BGB herangezogen werden, um festzustellen welche Leistungen der Auftragnehmer liefern muss:

> Das Werk ist frei von Sachmängeln, wenn es die vereinbarte Beschaffenheit hat. Soweit die Beschaffenheit nicht vereinbart ist, ist das Werk frei von Sachmängeln,
> 1. wenn es sich für die nach dem Vertrag vorausgesetzte, sonst

> 2. für die gewöhnliche Verwendung eignet und eine Beschaffenheit aufweist, die bei Werken der gleichen Art üblich ist und die der Besteller nach der Art des Werkes erwarten kann.

Der Auftraggeber kann also eine Ausführung einklagen, „die bei Werken der gleichen Art üblich ist“ und sich für die geplante Verwendung eignet. Ein Urteil des Bundesgerichtshofes lautet wie folgt:

> Haben die Vertragsparteien nicht im Einzelnen vereinbart, was das zu erstellende Programm zu leisten hat, schuldet der Unternehmer ein Datenverarbeitungsprogramm, das unter Berücksichtigung des vertraglichen Zwecks des Programms dem Stand der Technik bei einem mittleren Ausführungsstandard entspricht. Welche Anforderungen sich hieraus im Einzelnen ergeben, hat der Tatrichter gegebenenfalls mit sachverständiger Hilfe festzustellen. [9]

Um der Anwendung dieser doch recht generischen Paragraphen vorzubeugen sollten bei der Anforderungsanalyse unbedingt anwendbare Randbedingungen, Leistungs- und Qualitätsanforderungen definiert werden. (siehe Abbildung 2)

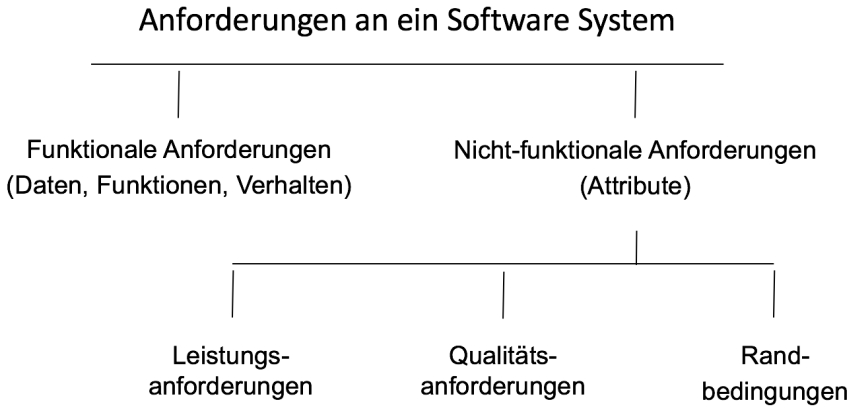

Abbildung 2. Anforderungen an ein Software System

2.1 Leistungsanforderungen

Unter Leistungsanforderungen versteht man im allgemeinen Anforderungen an die empirisch messbaren nicht-funktionalen Anforderungen eines Systems, also Anforderungen, deren zu Grunde liegendes Bedürfnis ein Leistungsmerkmal ist. [7] Ein Beispiel:

> Das System muss pro Sekunde mindestens 100 Emails verarbeiten können.

Leistungsmerkmale werden vom Benutzer explizit erwartet und sind verantwortlich für die als Performance wahrgenommenen Charakteristiken eines Systems. Die Erfüllung oder Nichterfüllung beeinflusst die Kundenzufriedenheit. [6] Leistungsmerkmale sind immer abhängig von einer messbaren Größe innerhalb des Systems. [7]

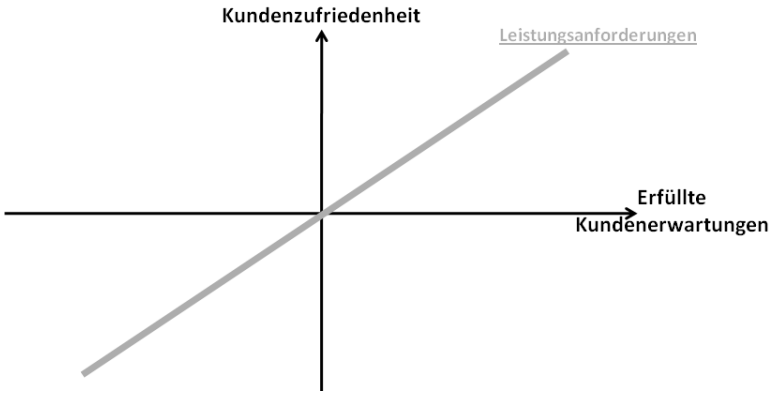

Abbildung 3. Erfüllung von Erwartungen bedingt Kundenzufriedenheit

Anforderungen auf Zeitbasis

- Maximale Zeit für eine Reaktion
- Zeit im Mittel für die Ausführung eines Prozesse
- Verfügbarkeit innerhalb eines Intervalls

Anforderungen basierend auf Raten

- Datendurchsatz
- Transaktionen
- Häufigkeit der Verwendung

Anforderungen basierend auf Mengen

- Minimaler benötigter Datenspeicher
- Darstellung von Inhalten pro Screen

Anforderungen an die Genauigkeit

- auf x Stellen
- Gleitkomma / Ganzzahlen
- Rundungsvorschriften
- Genauigkeit von Parametern

Anforderungen an Ressourcen

- Prozessorleistung
- Datenverbindungen
- Energie

2.2 Qualitätsanforderungen

Qualitätsanforderungen werden vom Benutzer subjektiv in Form von Attributen wie User Experience oder Sicherheit wahrgenommen und sind definiert als Anforderungen, deren zu Grunde liegendes Bedürfnis ein Qualitätsmerkmal ist. [7] Um Qualitätsmerkmale messbar zu machen bedarf es Metriken, auf die wir in Punkt 2.4 näher eingehen. Da die Menge möglicher Qualitätsanforderungen sehr groß und schwer überschaubar ist, hat es im Laufe der Zeit mehrere Versuche gegeben, standardisierte Qualitätsmodelle zu verfassen, die als Frameworks zur Anforderungsbestimmung eingesetzt werden können. Der aktuelle Standard ist ISO/IEC 25000.

ISO/IEC 25000 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) ersetzt seit 2005 die Norm ISO/IEC 9126 und dient als Einführung in die zugehörigen Standards für die Qualitätskriterien und -bewertung von Softwareprodukten.

Die Standardfamilie beinhaltet insgesamt drei Qualitätsmodelle: das *Quality In Use Model*, das *Product Quality Model* und das *Data Quality Model*. Diese können kombiniert als Framework von der Anforderungsbestimmung bis zur Qualitätssicherung eingesetzt werden. Die ersten beiden dieser Qualitätsmodelle

werden im Standard ISO/IEC 25010 definiert, der Software-Qualitätseigenschaften beschreibt und somit einen großen Teil der Qualitätsanforderungen an ein Softwareprodukt abdeckt.

Das Quality In Use Model setzt an der Schnittstelle zwischen Mensch und Software an und beschreibt fünf Charakteristiken in Bezug auf die Interaktion mit einem System: [10]

- Effektivität
- Effizienz
- Zufriedenheit (Nutzen, Vertrauen, Spaß, Komfort)
- Freiheit von Risiken (Ökonomischer Schutz, Gesundheits- und Umweltschutz)
- Rahmenabdeckung (Vollständigkeit, Flexibilität)

Das Product Quality Model (Abbildung 3) beschreibt acht Charakteristiken für Qualitätsanforderungen in Softwareprojekten mit jeweils mehreren Subcharakteristiken.

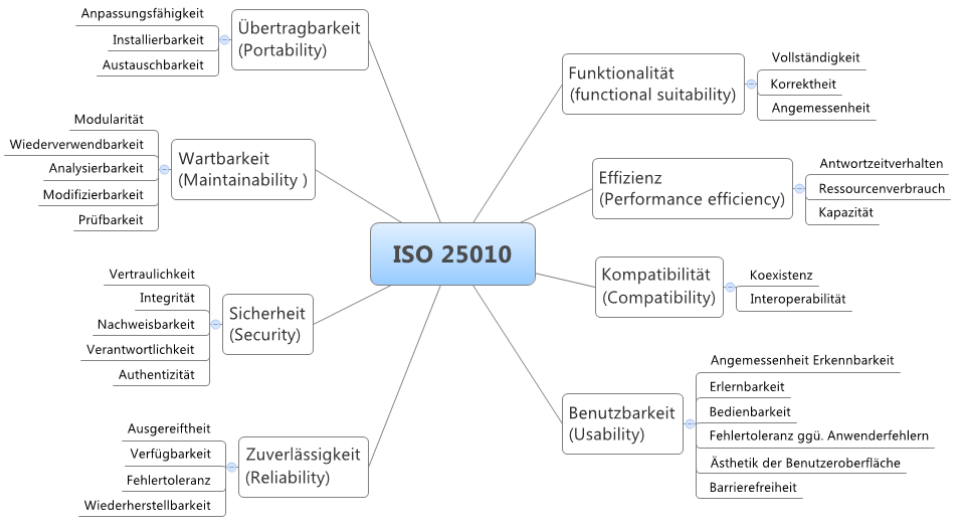

Abbildung 4. Qualitätsmodell aus ISO 25010 [11]

2.3 Randbedingungen

Unter einer Randbedingung versteht man eine Anforderung, deren Ursprung nicht von der Funktionalität des Systems ausgeht, sondern von externen Faktoren bestimmt wird. Bei der Anforderungsanalyse macht es Sinn, anfangs alle Randbedingungen zu sammeln und zu hinterfragen, ob diese nicht doch beeinflussbar sind, da später definierte Anforderungen darauf aufbauen. Unveränderliche Vorgaben des Auftraggebers mit einschränkendem Charakter bilden eine Kategorie von Randbedingungen, weitere Kategorien sind: [7]

Technische Randbedingungen

- Plattformen in Betrieb
- Bereits implementierte Schnittstellen
- Know-How der Belegschaft

Organisatorische Randbedingungen

- bestehende Prozesse
- Organisationsaufbau

Normative Randbedingungen

- Gesetze oder Verordnungen
- Normen

Kulturelle Randbedingungen

- Sprachen
- Gebräuche und Traditionen

2.4 Gewinnung

Die drei zuvor genannten Kategorien können als Leitfaden zur Gewinnung der nicht-funktionalen Anforderungen eines Projektes verwendet werden. Bei der Gewinnung von Leistungs- und Qualitätsanforderungen ist die Anforderung die Antwort auf eine Frage nach dem Attribut, zum Beispiel: [7]

Wie anpassungsfähig soll das System sein?

Die Antwort muss so formuliert werden, dass sie auch überprüft werden kann. Es reicht also nicht ein anpassungsfähiges System zu fordern, es müssen auch Metriken definiert werden, an denen gemessen werden kann, ob das System den erforderlichen Grad and Anpassungsfähigkeit liefert. Dieses Beispiel ist – im Gegensatz zu etwa Speicheranforderungen – schwer quantitativ zu messen, darum empfiehlt es sich hier Testfälle zu definieren, um reproduzierbare Ergebnisse und Metriken zu gewährleisten.

Bei der Gewinnung von Randbedingungen macht es ebenfalls Sinn, die oben genannten Punkte durchzugehen und die Anforderungen zu quantifizieren. Zuvor sollten die Vorgaben jedoch genau überprüft werden, ob sie wirklich als Randbedingungen einzustufen sind oder doch in eine andere, weniger starre Kategorie von Anforderungen fallen.

2.5 Metriken

Die genaue Formulierung von Anforderungen macht nur Sinn, wenn diese auch messbar, vergleichbar und reproduzierbar sind. [Weick] Deswegen sollen Abnahmekriterien an quantifizierbare Metriken geknüpft sein, mit denen entsprechende Ergebnisse erzielt werden können. Das IEEE Glossar (IEEE Std 610.12-1990) definiert Metriken wie folgt:

> metric – a quantitative measure of the degree to which a system or component possesses a given attribute.
> quality metric – a quantitative measure of the degree to which an item possesses a given quality attribute.

Metriken müssen also außerdem quantifizierbar, das heißt als Nummern ausdrückbar sein. Dadurch entstehen vertrauenswürdige Maßstäbe, die bei der Beobachtung verschiedener Zustände vergleichbare Werte generieren. Das kann die Zeit, die ein Proband für eine Aufgabe benötigt sein, eine Aussage über den Erfolg eines Prozesses, oder ein prozentualer Wert, zu welchem Anteil das System den Erwartungen entspricht. [12]

Besonders bei nicht-funktionalen Anforderungen besteht die Gefahr, in der Definition generisches Vokabular zu verwenden, das für jeden Leser eine andere Bedeutung hat. Anstatt beispielsweise eine „intuitive Benutzeroberfläche“ anzufordern, sollte sich der Verfasser Gedanken machen, wodurch die Intuitivität des Systems zu messen ist. Eine passende Metrik für dieses Beispiel wäre etwa „Ein Benutzer sollte nach einem Tag eigenständiger Einarbeitung mindestens 90% der Arbeitsschritte erfolgreich abschließen können.“

3 Fazit

Abschließend ist zu sagen, dass nicht-funktionale Anforderungen den funktionalen nicht an Wichtigkeit unterzuordnen sind, obwohl sie bei der Anforderungsdefinition oft vergessen werden. Die aufgeführten Kategorien und der Standard ISO/IEC 25000 können als Hilfestellung verwendet werden, um alle wichtigen Attribute eines Systems zu erfassen.

Bei der Gewinnung einer Anforderung kommt es weniger darauf an, ob es sich eindeutig um eine Leistungs- oder Qualitätsanforderung handelt, als viel mehr darum, passende Metriken zu definieren, die reproduzierbare Ergebnisse liefern und sich miteinander vergleichen lassen. Außerdem empfiehlt sich die Bestimmung von Prioritäten und Auswirkungen auf die Kundenzufriedenheit für jede Anforderung.

Rechtlich gesehen unterscheiden sich nicht-funktionale Anforderungen bis auf die schwierigere Bestimmung des Stands der Technik nicht von funktionalen Anforderungen und sollten darum auch ebenso ernst genommen werden.

Literatur

1. Davis, A. M. (1990). Software requirements - Analysis & Specification.
2. Hußmann, H. (1994). Zur formalen Beschreibung der funktionalen Anforderungen an ein Informationssystem.
3. Schienmann, B. (2002). Kontinuierliches Anforderungsmanagement.
4. Robertson, S., Robertson, J. (2006). Mastering the Requirements Process.
5. Baranowski, C. (2015). Software Qualitätssicherung.
6. Matzler, K., Bailom, F. (2004). Messung der Kundenzufriedenheit; in Kundenorientierte Unternehmensführung.
7. Glinz, M. (2006). Nicht-funktionale Anforderungen.
8. Weicker, K., Weicker, N. (2003). Softwaretechnik – Nicht funktionale Anforderungen.
9. BGH. (2003). Urteil vom 16.12.2003 – X ZR 129/01 – NJW-RR 2004, 782 – Herausgabepflicht bei Quellcode ohne ausdrückliche Vereinbarung
10. BSI. (2011). ISO/IEC 25010: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.
11. Comai, A. (2012). Non-functional requirements with ISO 25010.
12. Albert, W., Tullis, T. (2013). Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics.