

Fachbeitrag

Anforderungen an Softwareprodukte klar definieren

Requirements Engineering für Projektleiter

Teil 3: Anforderungen für die Software-Entwicklung ableiten

Die Merkmale der neuen Software sind festgelegt, der Funktionsumfang ist in Form von Use Cases modelliert. Als nächstes geht es darum, die Use Cases weiter zu detaillieren und daraus klare, für die Entwickler verständliche Anforderungen abzuleiten. Wie Sie dazu vorgehen, erklären Bettina Zastrow und Elisabeth Wagner in diesem dritten und abschließenden Teil der Artikelfolge.

Die Formulierung realistischer, eindeutiger und vollständiger Anforderungen hilft dem Projektleiter, der eine Software in Auftrag gibt, tatsächlich das System zu bekommen, das er will. Der erste Teil dieser Serie zeigt, wie die Software und die Rahmenbedingungen aus der Perspektive der Nutzer definiert werden. Der zweite Teil beschreibt die Formulierung der Use Cases: Was will welcher Akteur mit dem System tun? In diesem dritten und letzten Teil der Artikelfolge geht es darum, das Ziel der Softwareentwicklung aus Sicht der Entwickler zu beschreiben.

Sind die Use Cases fertig modelliert und zu Features zusammengefasst, lassen sich daraus die Anforderungen ableiten. Jedem Use Case werden dabei üblicherweise eine oder mehrere Anforderungen zugeordnet.

Mit Anforderungen sind hier standardisierte Sätze gemeint, die Ansprüche und Wünsche der Benutzer an die Funktionalitäten einer Software beschreiben. Auch Schnittstellen zu angrenzenden Systemen und eigenständige Systemprozesse sind zu berücksichtigen. Darüber hinaus können Vorgaben von außen existieren, z.B. aus firmeninternen Regelwerken, die ebenfalls als Anforderungen formuliert werden.

Definition: funktionale und nichtfunktionale Anforderungen

Ein Use Case beschreibt Funktionen, die ein Akteur mithilfe des Systems ausführen möchte. Daraus abgeleitete Anforderungen werden als "*funktionale* Anforderungen" bezeichnet. Sie beschreiben jeweils einen der folgenden drei Fälle:

- eine Interaktion zwischen dem Benutzer und dem System,
- eine Interaktion zwischen dem System und einem anderen System oder

Autor



Bettina Zastrow

Informatikerin; seit 2011
Geschäftsführerin Zastrow
information development

GmbH, Autorin von Software-
Handbüchern und Online-Hilfetexten

Kontakt: info@bettina-zastrow.de



Elisabeth Wagner

IPMA Level D, Kommuni-
kationsberaterin, Journa-
listin, seit 20 Jahren in der
Unternehmenskommunikation tätig

Kontakt:

wagner@projektmarketing-partner.de

Mehr Informationen unter:

› projektmagazin.de/autoren

ähnliche Artikel

› Die SOPHIST-Satzschablone: Funktio-
nale Anforderungen präzise formulieren

› Agile Softwareentwicklung mit Scrum
und User Stories

sowie in den Rubriken:

› [Anforderungsmanagement](#)

› [IT-Projekte](#)

- eine selbstständige Systemaktion aufgrund eines zeitlichen oder ereignisgesteuerten Auslösers.

Darüber hinaus ergeben sich weitere Anforderungen, z.B. rechtlicher, kultureller, sicherheitstechnischer Art, usw., die für die Anwendungsentwicklung bekannt sein müssen. Diese Anforderungen außerhalb der Use Cases werden als "*nichtfunktional*" bezeichnet (siehe Abschnitt "Nichtfunktionale Anforderungen berücksichtigen").

Funktionale Anforderungen ermitteln

Wünsche und Vorgaben der Stakeholder lassen sich am besten in Einzelinterviews ermitteln. Als Diskussionsgrundlage kann eine Liste der Use Cases dienen. Projektleiter und Stakeholder nehmen dabei die gleiche Perspektive auf das System ein und erarbeiten gemeinsam die Anforderungen aus der Nutzerperspektive, z.B. indem sie die Workflows zu den jeweiligen Use Cases durchgehen: Was sind z.B. für den Use Case "Nutzer registriert sich am System" die Voraussetzungen, was passiert im Hintergrund und was ist das Ergebnis?

Die Schwierigkeit ist hier häufig, exotische Anforderungen einzudämmen und die maßgeblichen Aspekte im Auge zu behalten, wie z.B. bei obigem Beispiel "Systemregistrierung" die Regeln für die Kennwortsicherheit.

Das Arbeitsergebnis ist eine Liste von Anforderungen aus dieser Nutzerrolle. Soweit möglich, sind die Anforderungen direkt im Format der SOPHIST[®] Satzschablone zu erfassen (siehe Abschnitt "Qualitätskriterien"). Anderenfalls reichen auch komplette Sätze aus, welche die Rolle, die Aktion und ggf. die Bedingung für die Ausführung beschreiben. Das Augenmerk ist dabei auf die vollständige Erfassung aller Anforderungen zu richten, die sich aus dem jeweiligen Use Case ergeben. Geschulte SW-Entwickler können hier gut unterstützen. Prüfung, Konsolidierung und Ergänzungen erfolgen in einem späteren Schritt.

! Um die Nachvollziehbarkeit zu gewährleisten, ist es ratsam, jede Anforderung bei der Erfassung mit einer Quellenangabe zu versehen. Dies ist auch später nützlich bei der Vergabe von Prioritäten.

Ein typischer Stolperstein ist die Schwierigkeit, Anforderungen am Rande des Spektrums zu erkennen und einzubeziehen. Am unteren Ende sind dies die banal erscheinenden Anforderungen, wie z.B. die Druckfunktion oder die Onlinehilfe, die so selbstverständlich sind, dass vergessen wird, sie explizit zu erwähnen. Am oberen Ende steht die Bedienbarkeit, auch als "Usability" bezeichnet. Zur Usability gehört es, die Benutzeroberfläche nach den Gesetzen der visuellen Wahrnehmung aufzubauen und effiziente Funktionen zu gestalten, die mit wenigen Schritten ausführbar sind.

Als Königsdisziplin der Benutzerfreundlichkeit gilt der "Joy of Use". Dieser Fachbegriff bedeutet, dass die Bedienung nicht nur angenehm sein, sondern Freude bereiten soll. Soll die Software diese Bedingung erfüllen, ist frühestmöglich der Rat von speziell geschulten Usability-Experten einzuholen. Diese kommen aus unterschiedlichen Fachrichtungen und sind entweder im Bereich der Psychologie oder im Umfeld von Web- und Grafikdesignern zu finden.

Wenn mit allen Stakeholdern ein Gespräch geführt wurde, entsteht als **Ergebnis** dieses Schritts eine **Liste von Anforderungen**, die den einzelnen **Use Cases zugeordnet** sind.

Der nächste Schritt umfasst eine Präzisierung der vorliegenden Anforderungen. Sie werden mit Eigenschaften wie Vorbedingungen und einer Priorität versehen. Die Erfassung kann in einer tabellarischen Form erfolgen.

Merkmale funktionaler Anforderungen

Tabelle 1 zeigt die typischen Bestandteile einer Anforderungsdefinition. Die in der linken Spalte aufgeführten Merkmale – von der Zuweisung einer eindeutigen Kennung für jede Anforderung bis hin zu dem Datum der Aufnahme einer Anforderung – sind ein häufig eingesetzter Standard im professionellen Requirements Engineering.

Element	Beschreibung
Nr.	Eindeutige Kennung für die Identifizierung der Anforderung
Name	Kurzname der Anforderung
Beschreibung und Zielsetzung	Langtext der Anforderung
Priorität	Rangfolge für die Umsetzung
Vorbedingungen	Bedingung, unter der die beschriebene Aktion ausgeführt werden soll
Fallbeispiel	Beispiel zur Veranschaulichung, wie diese Anforderung in den Gesamtkontext einzubetten ist
Offene Punkte	Sachverhalte, die noch zu klären sind
Quelle	Namenskürzel des Anforderers
Hinzugefügt am	Datum der Erstellung

Tabelle 1: Elemente einer funktionalen Anforderung.

Qualitätskriterien

Um bei Anforderungen eine hohe Qualität zu erreichen, kann auf den internationalen Standard ISO/IEC/IEEE 29148:2011 zurückgegriffen werden.

Dieser definiert für Anforderungen folgende Qualitätskriterien:

- **eindeutig:** frei von unklaren Formulierungen und Fehlern
- **vollständig:** enthält alle relevanten Sachverhalte
- **bewertet:** nach Priorität und Verbindlichkeit geordnet
- **konsistent:** in sich widerspruchsfrei sowie ohne Konflikte zu anderen Anforderungen
- **prüfbar:** enthält konkrete, messbare Daten und Eigenschaften, testbar
- **verfolgbar:** nachvollziehbar, wer die Anforderung eingebracht hat

Die ersten drei Kriterien lassen sich durch Verwendung der SOPHIST® Satzschablone erfüllen (siehe "**Die SOPHIST-Satzschablone: Funktionale Anforderungen präzise formulieren**", Projekt Magazin, Ausgabe 15/2014). Diese stellt eine Methodik zur Verfügung, um Anforderungen in natürlicher Sprache zu formulieren. Ein Satz wird strukturiert in sechs Schritten aufgebaut. Dabei werden das System, die rechtliche Verbindlichkeit, das Prozesswort, die Prozessart, das Objekt und die Bedingungen berücksichtigt.

Die Anforderung "Es soll Cross-Selling-Optionen geben" erfüllt z.B. die oben genannten Qualitätskriterien nur unzureichend (siehe Tabelle 2). Eine präzisere Formulierung, in der alle enthaltenen Fehler korrigiert sind, zeigt Tabelle 3.

Kriterium	Bewertung der Anforderung "Es soll Cross-Selling-Optionen geben"
bewertet	Es liegt keine Angabe zur Verbindlichkeit und zur Priorisierung der Anforderung vor ("soll")
eindeutig	Der Begriff "Cross-Selling" ist nicht näher definiert
konsistent	Ja
prüfbar	Es fehlt die Angabe, wem wo und wann Cross-Selling-Optionen angeboten werden sollen
verfolgbar	Es liegt keine Angabe zur Urheberschaft und zum Erstellungsdatum der Anforderung vor
vollständig	Es fehlt die Angabe, welche Produkte als Cross-Selling-Produkte klassifiziert werden sollen

Tabelle 2: Bewertung einer Anforderung nach der Norm ISO/IEC/IEEE 29148:2011.

Merkmal	Inhalt
Nr.	Req-11
Name	Cross-Selling-Option: "Andere Kunden haben auch angesehen"
Beschreibung und Zielsetzung	Der Kunde soll die Möglichkeit haben, ausgehend von einem Produkt weitere Produkte nach dem Kriterium "Andere Kunden haben auch angesehen" aufzurufen.
Priorität	4 - niedrig
Vorbedingungen	Benutzer befindet sich auf der Detailansicht eines Produktes
Fallbeispiel	Keine
Offene Punkte	Keine
Quelle	SMI
Hinzugefügt am	19.05.2015

Tabelle 3: Korrekte Anforderungsbeschreibung über Standardmerkmale.

Anforderungen, die mehrere Aspekte beinhalten, sollten in mehrere einzelne Anforderungen aufgeteilt werden. Das vereinfacht die Aufgaben der Entwickler wie auch der Tester. Erkennbar sind solche Anforderungen an dem Wort "und", z.B.: "Der Kunde muss bei der Registrierung AGB *und* Datenschutzerklärung bestätigen." Hier ist es sinnvoll, zwei separate Anforderungen zu formulieren. Dadurch können verschiedene Teammitglieder mit der Umsetzung betraut werden und diese können die Aufgaben einzeln abarbeiten und als erledigt markieren. Dies erleichtert die Kontrolle des Projektfortschritts.

Beispiel Jingleshop: Funktionale Anforderungen

Zur Verdeutlichung wird am Beispiel "Jingleshop" gezeigt, wie der Use Case "Anmelden" in einzelne technische Anforderungen aufgeschlüsselt wird, bis Missverständnisse nahezu ausgeschlossen sind. Für jeden Teilaspekt des Use Case wird dabei eine eigene Anforderung formuliert. Folgende fünf Anforderungen wurden ermittelt:

- Anmeldung mit Benutzername und Kennwort
- Eigenschaften Feld Benutzername: Länge
- Eigenschaften Feld Kennwort: Länge
- Eigenschaften Feld Kennwort: Darstellung
- Eigenschaften Feld Kennwort: Zwischenablage

Tabellen 4 bis 8 zeigen die vollständigen Anforderungsbeschreibungen.

Merkmal	Inhalt
Nr.	Req-21
Name	Anmeldung mit Benutzername und Kennwort
Beschreibung und Zielsetzung	Der Kunde muss jederzeit in der Lage sein, sich mit Benutzername und Kennwort am System anzumelden.
Priorität	1 - kritisch
Vorbedingungen	Anmeldung ist noch nicht erfolgt. Benutzer ist registriert
Fallbeispiel	Der Kunde befindet sich in einem Bestellprozess und möchte sich anmelden.
Offene Punkte	Keine
Quelle	SMI
Hinzugefügt am	19.05.2015

Tabelle 4: Beispiel für funktionale Anforderung "Anmeldung".

Merkmal	Inhalt
Nr.	Req-22
Name	Eigenschaften Feld Benutzername: Länge
Beschreibung und Zielsetzung	Das System muss fähig sein, einen Benutzernamen mit der Länge von 130 Zeichen zu verarbeiten
Priorität	3 - niedrig
Vorbedingungen	Keine
Fallbeispiel	Keine
Offene Punkte	Keine
Quelle	SMI
Hinzugefügt am	19.05.2015

Tabelle 5: Beispiel für funktionale Anforderung "Länge des Felds Benutzername".

Merkmal	Inhalt
Nr.	Req-23
Name	Eigenschaften Feld Kennwort: Länge
Beschreibung und Zielsetzung	Das System muss fähig sein, ein Kennwort mit der Länge von 20 Zeichen zu verarbeiten
Priorität	3 - niedrig
Vorbedingungen	Keine
Fallbeispiel	Keine
Offene Punkte	Keine
Quelle	SMI
Hinzugefügt am	19.05.2015

Tabelle 6: Beispiel für funktionale Anforderung "Länge des Feldes Kennwort".

Merkmal	Inhalt
Nr.	Req-24
Name	Eigenschaften Feld Kennwort: Darstellung
Beschreibung und Zielsetzung	Das System muss fähig sein, ein Kennwort nach einer Verzögerung von 1 Sekunde verdeckt anzuzeigen
Priorität	2 - hoch
Vorbedingungen	Keine
Fallbeispiel	Keine
Offene Punkte	Keine
Quelle	SMI
Hinzugefügt am	19.05.2015

Tabelle 7: Beispiel für funktionale Anforderung "Darstellung des Feldes Kennwort".

Merkmal	Inhalt
Nr.	Req-25
Name	Eigenschaften Feld Kennwort: Zwischenablage
Beschreibung und Zielsetzung	Das System muss verhindern, dass der Benutzer den Inhalt des Feldes "Kennwort" in die Zwischenablage kopiert
Priorität	2 - hoch
Vorbedingungen	Es wurde eine Eingabe im Feld Kennwort getätigt. Benutzer kopiert den Inhalt in die Zwischenablage per Maus oder Tastatur
Fallbeispiel	Keine
Offene Punkte	Keine
Quelle	FRT
Hinzugefügt am	21.06.2015

Tabelle 8: Beispiel für funktionale Anforderung "Kopierschutz für das Feld Kennwort".

Als **Ergebnis** entsteht in diesem Schritt eine Zusammenstellung **aller funktionalen Anforderungen**. Für eine leichtere Handhabung können die Zeilen und Spalten der Einzeltabellen getauscht und in einer Gesamttabelle zusammengeführt werden. Der Umfang des Projektdokuments hängt von der Größe des Projekts und der Anzahl der Anforderungen ab. Zugleich können die Zusammenhänge zwischen den Use Cases und den Anforderungen hergestellt werden. Dies geschieht entweder mit der Platzierung der Anforderungen bei den zugehörigen Use Cases oder durch Querverweise.

! Es hat sich als praktisch erwiesen, die einmal vergebene Nummerierung für Anforderungen beizubehalten, d.h. einmal vergebene Nummern nicht mehr zu ändern. Das erleichtert das Erkennen von neuen Anforderungen. Weiterhin sind anhand dieser Nummern eindeutige Querbezüge zwischen den Anforderungen untereinander sowie zwischen den Anforderungen und Use Cases möglich. Da der Text einer Anforderung im Lauf des Projekts ggf. verändert wird, bleibt der Bezug über die Nummerierung, die eine eindeutige Kennung darstellt, erhalten. Im Verlauf des Projekts neu hinzugekommene Anforderungen können fortlaufend weiternummeriert werden.

Nichtfunktionale Anforderungen berücksichtigen

In den nichtfunktionalen Anforderungen wird vorgegeben, welche Eigenschaften die Funktionen der Software haben sollen, um die gegebenen Randbedingungen zu erfüllen. Hierbei handelt es sich um Eigenschaften wie Benutzbarkeit, Systemstabilität, Leistung, Betreibbarkeit usw. Betrachtet wird das Umfeld der Software, wie die Daten, die Benutzer, die Systemumgebung sowie begleitende Prozesse wie Rollout, Dokumentation und Schulung.

Die nichtfunktionalen Anforderungen lassen sich aus dem Systemkontext und aus der Norm für Softwarequalität nach ISO/IEC 2500 ableiten. Eine mögliche Gliederung ist wie folgt:

- **Systemtechnische Anforderungen:** Anforderungen aus bestehender oder genutzter Hardware, Software und Infrastruktur (z. B. Netzwerk)
- Anforderungen der **Erfüllung bestimmter Vorschriften:** Anforderungen aus externen und internen Regelwerken
- **Softwarequalitäts-Anforderungen:** Anforderungen, die die Performanz, Zuverlässigkeit, Usability, Stabilität und Angemessenheit betreffen
- **Datenanforderungen:** Anforderungen technischer Natur zur Datenspeicherung und -verarbeitung
- **Archivierungsanforderungen:** Anforderungen, die aus Vorgaben zur Archivierung hervorgehen
- **Sicherheitsanforderungen:** Anforderungen aus Datenschutz und Datensicherheit, sowie Arbeitssicherheit
- **Betriebsanforderungen:** Anforderungen für die Betriebsführbarkeit, beispielsweise Vorgaben zur Datensicherung und -wiederherstellung
- **Lizenzierungsanforderungen:** Anforderungen, die darauf abzielen, das Softwareprodukt lizenzierbar zu machen
- **Portierungs- und Migrationsanforderungen:** Anforderungen zur Plattformunabhängigkeit
- **Rollout- und Schulungsanforderungen:** Anforderungen, die die Einführung der Software in den Regelbetrieb ermöglichen oder erleichtern

- **Dokumentationsanforderungen:** Anforderungen an Entwicklungs-, Benutzer-, Administrations- und Betriebsdokumentation
- **Sonstige Anforderungen:** branchen- oder unternehmensspezifische Anforderungen, die nicht von den oben genannten Themengebieten abgedeckt wurden

Diese Themengebiete sind für alle Softwareprojekte relevant und erfordern tiefer gehende Kenntnisse der IT, der Branche und der anwendbaren Regelwerke. In größeren Unternehmen kann in der Regel auf existierende Vorgaben zurückgegriffen werden. Falls nicht, können Sie die nichtfunktionalen Anforderungen auf gleiche Weise wie die funktionalen Anforderungen mittels eines Interviews erheben. Mit Vorlagen für die Anforderungsdefinition, sog. "Requirements Specification Templates", die auf dem Markt erhältlich sind, ist der Projektleiter gut gerüstet, um eine Vollständigkeit der nichtfunktionalen Anforderungen herzustellen.

In der Word-Datei, die Sie zusammen mit dem Artikel herunterladen können, finden Sie einen Gesprächsleitfaden sowie eine Liste der Ansprechpartner (Rollen) für die einzelnen Interviews sowie Empfehlungen, zu welchen Themen diese jeweils Auskunft geben können. Die ebenfalls beigefügte Excel-Datei dient zur Dokumentation der erhobenen Anforderungen und enthält zu jeder der oben genannten Themenkategorien ein exemplarisches Beispiel. Die Beispiele können Sie genau so oder mit geringfügigen Änderungen für jedes beliebige Softwareprojekt übernehmen. Überdies stellen die Beispiele eine Schablone dar, um mit den genannten Ansprechpartnern die relevanten nichtfunktionalen Anforderungen für jeden der genannten Bereiche zu erarbeiten und zu formulieren. Die Systematik für die Dokumentation gleicht jener der funktionalen Anforderungen.

Typische Stolpersteine bei diesem Schritt sind, Anforderungen festzulegen, die die Zukunft betreffen, z.B. "Unterstützung aller Betriebssysteme ab Windows 10.x aufwärts". Hier hilft eine einfache Frage weiter: "Ist die Anforderung testbar?" Die Antwort in diesem Fall würde "nein" lauten, da zum Zeitpunkt der Publikation dieses Artikels noch kein Nachfolger von Windows 10 verfügbar ist.

Als **Ergebnis** entsteht in diesem Schritt eine Zusammenstellung **aller nichtfunktionalen Anforderungen**. Diese können in das bestehende Projektdokument unter einer separaten Überschrift nach den funktionalen Anforderungen eingefügt werden.

Das dokumentierte und genehmigte **Ergebnis aller Schritte** dieser Artikelserie **entspricht dem Fachkonzept**. Um formale und organisatorische Kriterien ergänzt, entsteht daraus ein Lastenheft.

Wurde im Unternehmen bereits ein Softwareprojekt durchgeführt, ist es von großem Nutzen, die nichtfunktionalen Anforderungen zentral abzulegen und für jedes weitere Softwareprojekt wiederzuverwenden. Ist das noch nicht geschehen, kann der betroffene Projektleiter das beim Leiter der IT-Abteilung anregen (siehe "**Nichtfunktionale Anforderungen strukturiert erfassen und wieder verwenden**", Projekt Magazin, Ausgabe 8/2007)

Es zahlt sich aus, den nichtfunktionalen Anforderungen einen ebenso hohen Stellenwert einzuräumen wie den funktionalen Anforderungen. Erfüllt das fertig entwickelte Produkt z.B. die rechtlichen Vorgaben nicht, kann das Produkt im schlimmsten Fall nicht auf den Markt gebracht werden. Bleiben bestimmte Vorgaben unberücksichtigt, z.B. zur Architektur, zu möglichen Clients, zu Antwortzeiten und zur Bedienbarkeit, kann dies zur Folge haben,

dass das Produkt nur unter hohem Aufwand eingeführt, betrieben, lokalisiert, portiert, migriert oder weiterentwickelt werden kann, oder das Produkt nicht bedienerfreundlich ist und der finanzielle Erfolg ausbleibt.

Fazit

Dieser dreiteilige Beitrag zeigt, wie auch Projektleiter, die keine Ausbildung oder Erfahrung im professionellen Requirements Engineering haben, mithilfe eines systematischen Prozesses zu eindeutigen und vollständigen Anforderungen für eine neue Software gelangen. Der Prozess folgt zwei grundsätzlichen Vorgehensweisen: Zuerst verläuft die Denkweise vom Allgemeinen zum Konkreten. Damit wird gewährleistet, dass das große Bild stimmt, bevor man – sorgfältig und umfassend – die Details beschreibt. Danach verläuft der Prozess von der Nutzerorientierung zur IT-Orientierung, von der Anwendersprache zur IT-Notation. So gelingt die Kommunikation zwischen Technikern und Nichttechnikern, obwohl sie grundsätzlich unterschiedliche Sprachen sprechen.

Ausblick

Professionelles Requirements Engineering hat neben der Formulierung von Anforderungen viele weitere Facetten. Zum Beispiel gibt es professionelle Methoden, um Anforderungen auf Konsistenz zu prüfen und Widersprüche aufzulösen oder nach Wichtigkeit und Zeit zu priorisieren. Darüber hinaus ist es notwendig, auf Ereignisse während der Umsetzungsphase zu reagieren, die zu einer Änderung, Ergänzung oder Umpriorisierung von Anforderungen führen. Ein weiterer Artikel zu diesen Themen ist geplant.

Hat Ihnen dieser Artikel gefallen?

Bewerten Sie ihn im Projekt Magazin online und teilen Sie so Ihre Meinung anderen Lesern mit. Wählen Sie dazu den Artikel im Internet unter <http://www.projektmagazin.de/ausgaben/2016> oder klicken Sie [hier](#), um direkt zum Artikel zu gelangen.