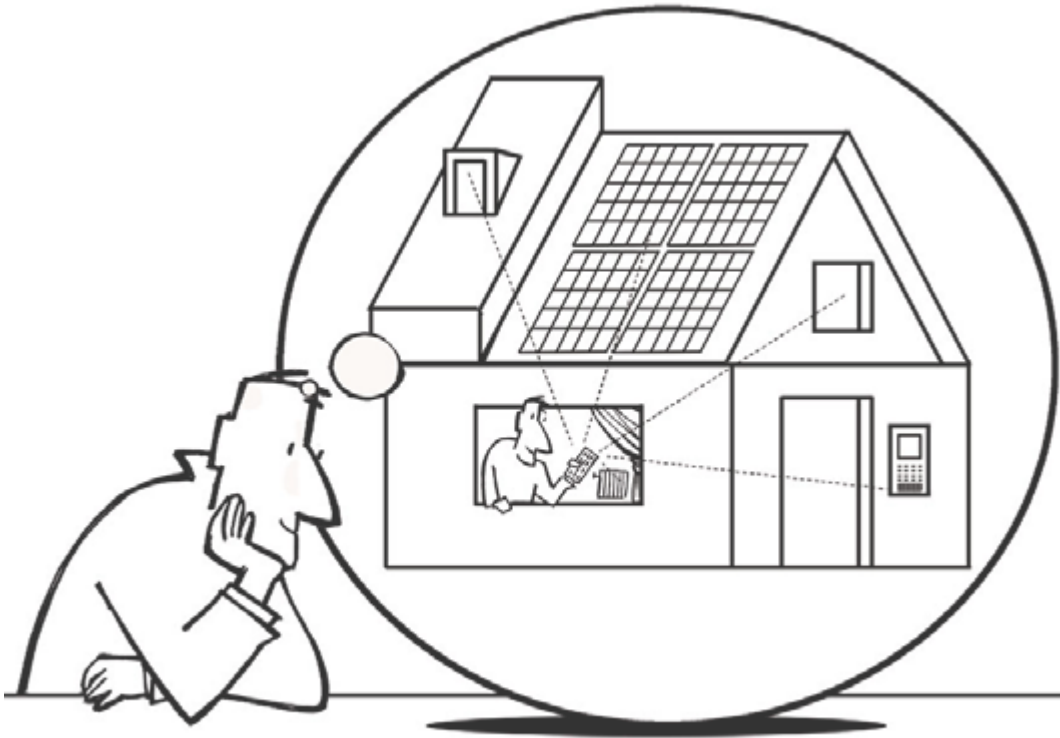


Anforderungen analysieren – vom Wunsch zur Absicht



12.1 Überblick über die Analyse von Anforderungen

Wie im vorigen Kapitel angesprochen, ist das Ziel der Analyse von Ursprungsanforderungen (also Anforderungen, die als Ausgangsbasis dienen), belastbare Anforderungen an den gewählten Betrachtungsgegenstand zu ermitteln. Diese Analyse soll angemessen durchgeführt werden, das heißt, es soll nicht mehr Aufwand in diesen Schritt investiert werden, als unbedingt notwendig ist. Leider zeigt der Projektalltag, dass auch dafür nicht immer die benötigten Ressourcen zur Verfügung stehen. Als Ausweg müssen die zur Verfügung stehenden Ressourcen möglichst gewinnbringend eingesetzt werden. Dazu sollten Sie sich zunächst Gedanken darüber machen, welche Qualitätskriterien (siehe Kapitel 3 „Requirements-Engineering im Überblick“) für Ihre Anforderungen im Vordergrund stehen sollten. In den wenigstens Fällen können Sie Anforderungen erzeugen, die zu 100 % allen Qualitätskriterien genügen. Des Weiteren sollten Sie sich immer wieder fragen, mit welchen Anforderungen Sie noch nicht zufrieden sind bzw. an welchen Stellen Ihrer Anforderungssammlung Sie als Nächstes arbeiten müssen, um die Konsumenten der Anforderungen zufriedenzustellen, damit diese die ihnen gestellten Aufgaben erledigen können.



Um diese beiden Gedanken zu unterstützen, wollen wir die Analyse von Anforderungen etwas formalisieren, indem wir in diesem Kapitel sechs unterschiedliche Analyseaufgaben angeben. Mit deren Hilfe könnten Sie eine Anforderungssammlung erzeugen, die zu 100 % den Qualitätskriterien genügt. Oder die – was in der Praxis eher benötigt wird – Ihnen erlauben zu entscheiden, an welcher Stelle Sie Ihre Spezifikation als Nächstes weiter verbessern sollten, sofern Sie noch Ressourcen zur Verfügung haben.

Dabei gilt es zu beachten: Auch wenn durch die Analysetätigkeiten neue Anforderungen entstehen, bedeutet dies nicht, dass Sie diese neuen Anforderungen auch dokumentieren müssen. Sie dienen primär als Ausgangspunkt für weitere Analyseschritte; die Entscheidung, ob Sie diese Anforderungen auch dokumentieren oder beispielsweise nur verbal weitergeben, unterliegt anderen Kriterien (siehe Kapitel 16 „Wegweiser: Anforderungen dokumentieren und vermitteln“).



Durch die Analysetätigkeiten entsteht zunächst im Kopf ein Bild der Anforderungen. Dieses Bild kann, muss aber nicht, vollständig in einer Anforderungsspezifikation oder einem Backlog einer agilen Entwicklung dokumentiert werden.

Auch wenn wir im Folgenden von Anforderungen oder Spezifikationen reden, so sind die Aussagen in diesem Kapitel nicht auf die klassischen Artefakte beschränkt. Die Artefakte einer agilen Entwicklung (Backlog-Items, User-Storys, Akzeptanzkriterien etc., siehe Kapitel 17 „Storytelling, User-Storys und Co.“) können ebenfalls durch die hier beschriebenen Aufgaben entstehen und analysiert werden. Das prinzipielle Vorgehen bei der Analyse von Anforderungen ist in **Abbildung 12.1** mit Beispielen für Übergänge zwischen Artefakttypen definiert.

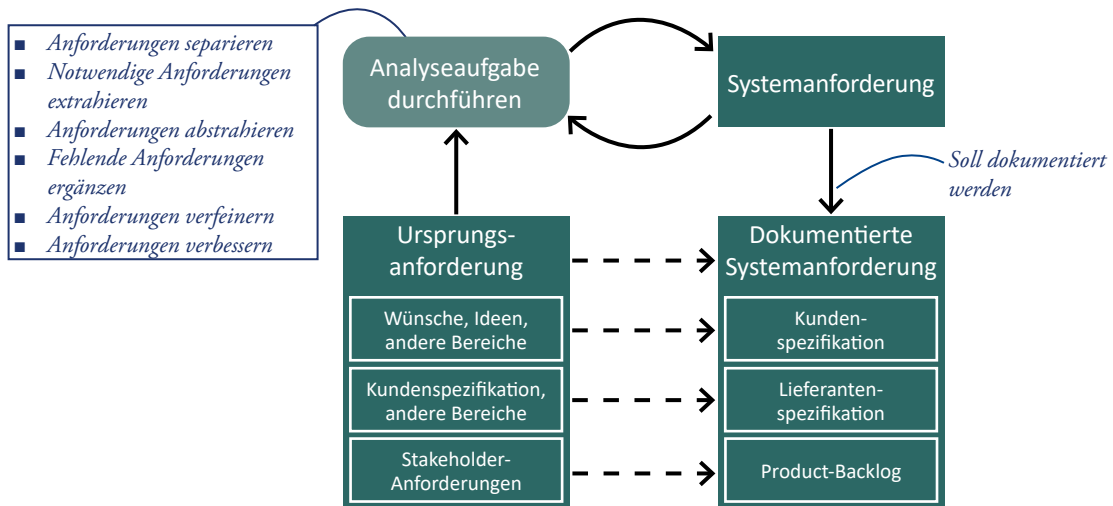


Abbildung 12.1: Vorgehen bei der Analyse von Anforderungen

Wir führen für eine Ursprungsanforderung eine der Analyseaufgaben durch und erhalten dadurch eine Systemanforderung. Unabhängig davon, ob Sie diese Anforderung in die dokumentierten Anforderungen aufnehmen möchten oder nicht, muss die neue Anforderung noch nicht Ihren Ansprüchen genügen. Somit können Sie für diese neu gefundene Systemanforderung eine weitere Analyseaufgabe durchführen, um sie zu verbessern oder daraus neue Anforderungen herzuleiten.

Bevor nun die einzelnen Aufgaben im Detail vorgestellt werden, müssen noch zwei übergreifende Themen geklärt werden. Zunächst möchten wir Ihnen das Gesamtbild der Anforderungen vorstellen, das mithilfe der hier beschriebenen Tätigkeiten ermittelt wird. Danach werden wir die einzelnen Tätigkeiten in Zusammenhang stellen, um dieses Gesamtbild zu erzeugen.

12.1.1 Den Wald trotz vieler Bäume sehen

Mithilfe der hier vorgestellten Aufgaben werden Sie unter anderem viele, eventuell bislang fehlende Anforderungen erzeugen. Dabei gehen wir davon aus, dass jede Anforderung eine andere Anforderung verfeinert (siehe auch Kapitel 3 „Requirements-Engineering im Überblick“). Eine Ausnahme bildet dabei die abstrakteste Ebene von Anforderungen.

Deren Anforderungen stehen nebeneinander und bilden die Basis für die verfeinernden Anforderungen. Somit ergeben sich viele Bäume (mathematisch als „Wald“ bezeichnet), wobei die Wurzeln durch die abstraktesten Anforderungen gebildet werden und die Blätter durch die Anforderungen repräsentiert werden, die nicht mehr verfeinert werden.

An dieser Stelle kommt die Unterscheidung zwischen nicht-funktionalen und funktionalen Anforderungen mit ins Spiel. Die entstehenden Verfeinerungshierarchien können in diese beiden Arten unterschieden werden, wobei Abhängigkeiten zwischen diesen beiden Teilen bestehen. So kann eine nicht-funktionale Anforderung durch eine funktionale Anforderung verfeinert werden bzw. kann umgekehrt eine nicht-funktionale Anforderung als Begründung für eine Funktion des Systems dienen (siehe [Abschnitt 12.2.5 „Anforderungen verfeinern“](#)). In [Abbildung 12.2](#) ist ein unvollständiges Beispiel für diese Zusammenhänge am Beispiel einer Use-Case-Verfeinerung gegeben.

Allerdings kann eine Anforderung auch aus zwei abstrakteren Anforderungen resultieren. Damit müsste man anstatt von Bäumen von Graphen reden, da in diesem Fall ein Knoten in dem Baum mehr als ein Elternelement besitzt. In dem Beispiel in [Abbildung 12.2](#) könnte dies die Forderung nach der Erkennung der Bewegung sein, die in anderen Use-Case-Abläufen zusätzlich benötigt werden könnte. Für die Vorstellung der Analyseaufgaben in diesem Kapitel wollen wir jedoch diesen Fall außer Acht lassen und weiterhin in Bäumen denken. Wie Sie in [Kapitel 21 „Strukturen und Zustände“](#) lesen werden, kann die Verfeinerungshierarchie zur Strukturierung einer Anforderungsspezifikation genutzt werden. Dabei muss der Fall mehrerer abstrakter Anforderungen als Begründung zu einer Anforderung gesondert betrachtet werden.

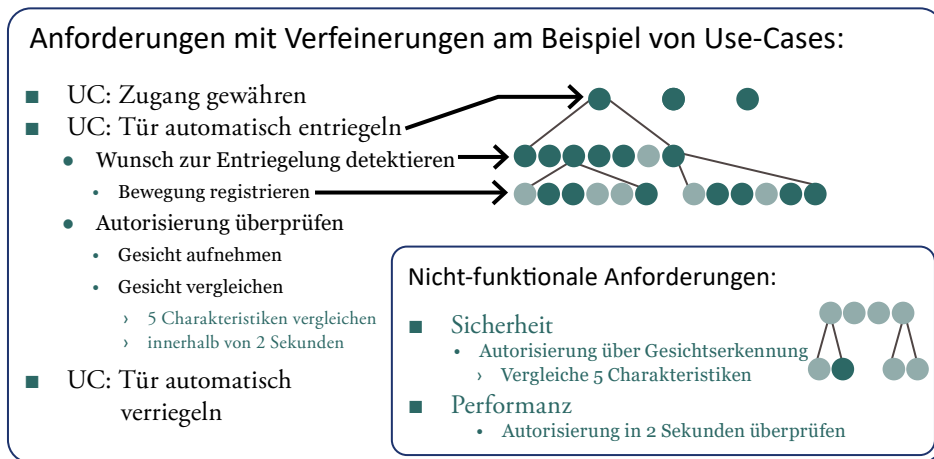


Abbildung 12.2: Zusammenhang zwischen Anforderungen

12.1.2 Der Ablauf bei der Anforderungsanalyse

In diesem Abschnitt möchten wir die Aufgaben im Überblick darstellen und sie in eine Reihenfolge bringen. Dies soll nur als Leseanleitung dienen, da die Aufgaben nicht

unbedingt in dieser Reihenfolge bearbeitet werden müssen. Manchmal kann eine Aufgabe für eine betrachtete Anforderung ausgelassen werden, manchmal muss sie zweimal durchgeführt werden. Wann es sinnvoll ist, eine der genannten Aufgaben auch wirklich durchzuführen, lesen Sie bitte in den folgenden Abschnitten.

Als Eingabe in die Analyse nehmen wir eine Anforderung an, die Ihnen von einem Stakeholder mitgeteilt wurde oder im Rahmen eines Lastenhefts bei Ihnen aufgeschlagen ist. In dem Schritt *Anforderungen separieren* sollten Sie als Erstes diese Ursprungsanforderung gegebenenfalls in mehrere Anforderungen zerlegen, um diese in den nächsten Schritten getrennt voneinander betrachten zu können. Im zweiten Schritt (*Notwendige Anforderungen extrahieren*) sollten Sie überprüfen, ob sich die separierten Anforderungen auch wirklich an das betrachtete System richten. Ist dies nicht der Fall, so müssen Sie den für Ihr System relevanten Anteil für die weitere Betrachtung identifizieren. Diese beiden Schritte sollten Sie auf alle Ursprungsanforderungen anwenden, um so einen guten Startpunkt für die weitere Analyse zu erzeugen.

Die nächste Aufgabe ist es, zu einer Anforderung abstraktere Anforderungen zu finden (*Anforderungen abstrahieren*), bis Sie die oberste Ebene ihrer Anforderungen, also die Wurzeln der zu erzeugenden Bäume, identifiziert haben. Dabei können Sie sowohl auf neue Anforderungen stoßen als auch bereits bestehende Anforderungen in eine Abstraktions-/Verfeinerungshierarchie einordnen. Die oberste Ebene wird dann in dem Schritt *fehlende Anforderungen ergänzen* vervollständigt, in dem Sie weitere Anforderungen auf dieser Ebene definieren können.

Damit haben Sie den Startpunkt für die vorletzte Tätigkeit gefunden (*Anforderungen verfeinern*), in der Sie nun entscheiden können, welche der bestehenden Knoten in den Bäumen weiter verfeinert werden sollen.

Zum Abschluss der Analyse können Sie die gefundenen Anforderungen auf ihre individuelle Qualität überprüfen, um sie z. B. eindeutig zu formulieren (*Anforderungen verbessern*).

In der Praxis werden Sie bei der Anwendung der einzelnen Tätigkeiten Wissen benötigen, das über die Informationen in den Ursprungsanforderungen hinausgeht. Um diese Wissenslücken auszugleichen, werden Sie Ihre Stakeholder bzw. Auftraggebenden fragen müssen oder Annahmen treffen, die im Nachhinein dann noch abgesichert werden müssen.

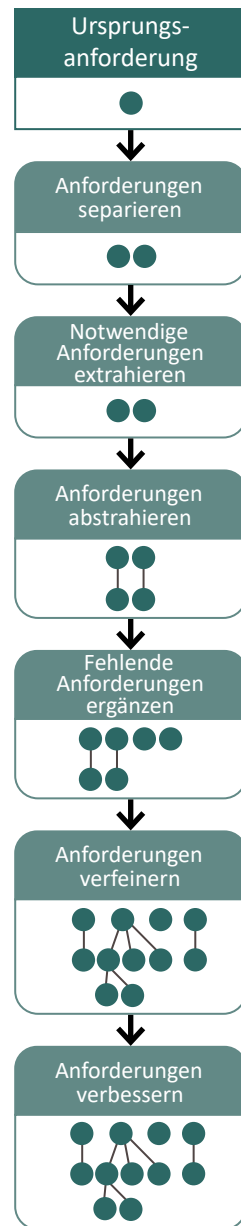


Abbildung 12.3:
Die Analyseaufgaben in der Übersicht

12.2 Die Aufgaben im Detail

Kommen wir nun zum Kern dieses Kapitels: Die Beschreibung der einzelnen Aufgaben. Sie werden in einer bestimmten Reihenfolge dargestellt, jedoch können die Schritte wiederholt oder in einer anderen Reihenfolge für eine Ursprungsanforderung durchgeführt werden.

Für die Beschreibung gehen wir von vier Ursprungsanforderungen an unser Smart-Home-System (SHS) aus, wie die Stakeholder sie Ihnen genannt haben könnten.

Das SHS muss die Erlaubnis zum Öffnen der Tür überprüfen und innerhalb von 2 Sekunden die Tür öffnen. (U1)

Das SHS muss jede Entriegelung einer Tür auf konfigurierten Handys und Tablets innerhalb von 2 Sekunden anzeigen. (U2)

Der Türsensor muss den Status einer Tür (offen/geschlossen) melden. (U3)

Die Nutzung des SHS muss so einfach wie möglich sein. (U4)

Durch die Bearbeitung der Ursprungsanforderungen gemäß der sechs Aufgaben werden sie sukzessive in belastbare Systemanforderungen, in die Ergebnisanforderungen, überführt. Dabei werden sie sowohl um neue Anforderungen ergänzt als auch in Richtung der Eindeutigkeit und Notwendigkeit verbessert. Hier kommen auch die in [Kapitel 9](#) „Das SOPHIST-REGelwerk“ vorgestellten Regeln ins Spiel, die Ihnen Hinweise geben, welche Defizite in einer betrachteten Anforderung existieren können.

12.2.1 Anforderungen separieren

Ihre erste Aufgabe bei der Analyse von vorgegebenen Anforderungen ist es, sie in ihre Bestandteile zu zerlegen. Betrachten wir dazu die folgende Beispielanforderung.

Das SHS muss die Erlaubnis zum Öffnen der Tür überprüfen und innerhalb von 2 Sekunden die Tür öffnen. (U1)

Es ist offensichtlich, dass in dieser Anforderung zwei Funktionen von dem System gefordert werden: Das SHS soll die Erlaubnis überprüfen und die Tür öffnen. Dabei ist nur implizit ausgedrückt, dass es die Tür natürlich nur bei erfolgreicher Prüfung öffnen soll. Daraus folgt ein Grund, warum wir in einer Anforderung nur eine Funktion (bzw. eine funktionale Anforderung) fordern wollen: Der Zusammenhang zwischen den Funktionen kann explizit ausgedrückt werden.

Die beiden Funktionen in der obigen Anforderung und die (nicht eindeutig dem Erlaubnis überprüfen oder dem Tür öffnen zuordenbare) Anforderung an die Performanz bringen uns zu einem weiteren Grund für die Zerlegung: Die nachfolgende Bearbeitung der Anforderungen kann getrennt voneinander erfolgen. Für die nachfolgenden Entwicklungsschritte kann eine getrennte Betrachtung sinnvoll sein. Hierzu einige Beispiele:

- Die Funktionen sollen getrennt voneinander getestet werden: Dabei würde die gesamte Ursprungsanforderung auch bei einem „Fail“ von einem Teil der Anforderung als „nicht bestanden“ gewertet werden müssen.

- Die Anforderung an die Performanz soll später entwickelt werden: Auch hier hätten Sie nur die Chance, die gesamte Anforderung als „realisiert“ zu bewerten, und nicht den tatsächlich entwickelten Teil. In einer agilen Entwicklung spricht man hier von einem Zerlegen einer User-Story.
- Nur ein Teil der Anforderung soll weiter verfeinert werden: Es sollte beim Lesen der Anforderung offensichtlich sein, welchen Teil der Anforderung Sie mit weiteren Anforderungen weiter beschreiben.

Wir können uns also die folgenden Fragen stellen, um zu entscheiden, ob eine Anforderung zerlegt werden sollte oder nicht.

- Beinhaltet die Anforderung unterschiedliche Funktionen des Systems?
- Existieren unterschiedliche Bedingungen für die einzelnen Anteile der Anforderung?
- Beinhaltet die Anforderung eine Qualitätsanforderung zu einer funktionalen Anforderung?

Sobald Sie eine oder mehrere der Fragen mit Ja beantworten, haben Sie gute Gründe, Ihre Anforderungen in ihre Bestandteile zu zerlegen und diese im Weiteren als eigenständige Anforderungen zu betrachten. Falls Sie sich unsicher sind, sollten Sie diese Separierung trotzdem durchführen, da im weiteren Verlauf der Entwicklung eher Gründe für eine getrennte Betrachtung hinzukommen werden als umgekehrt.

Für die Ursprungsanforderung U1 erhalten Sie so die drei Ergebnisanforderungen (E1 bis E3):

Das SHS muss die Erlaubnis zum Öffnen einer Tür überprüfen und innerhalb von 2 Sekunden die Tür öffnen. (U1)

Das SHS muss die Erlaubnis zum Öffnen einer Tür überprüfen. (E1)

Sobald das SHS die Erlaubnis zum Öffnen einer Tür überprüft hat und falls die Überprüfung positiv war, muss das SHS die entsprechende Tür öffnen. (E2)

Die Funktion Tür öffnen muss eine Ausführungszeit von maximal 2 Sekunden aufweisen. (E3)

Die Umformulierung beinhaltet einige Annahmen, die nicht explizit in der Ursprungsanforderung enthalten waren, z. B. die Zuordnung der zwei Sekunden zum Öffnen der Tür nach Überprüfung. Seien Sie vorsichtig bei solchen Annahmen und lassen Sie sich diese von Ihrem Stakeholder bestätigen.

Abbildung 12.4 zeigt die Ausgangsmenge und die Menge der durch die Separierung erzeugten Anforderungen. Der hier dargestellte, noch sehr kleine „Wald“ auf der rechten Seite wird durch die folgenden Aufgaben in der Analyse sukzessive ausgebaut.

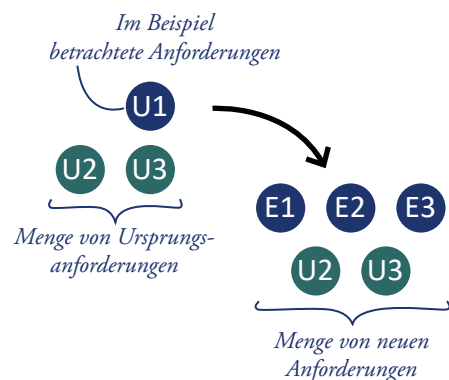
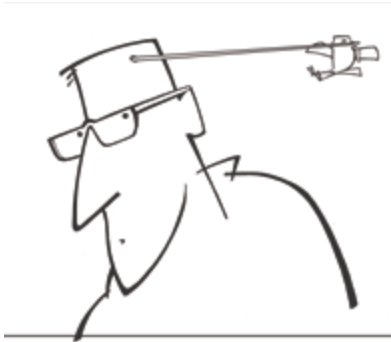


Abbildung 12.4: Ergebnis der Aufgabe Anforderungen separieren

12.2.2 Notwendige Anforderungen extrahieren



Häufig drücken die Anforderungen der Stakeholder nur Wünsche aus, die zum einen nicht alle erfüllt werden können und die zum anderen vielleicht nicht alleine von dem betrachteten System realisiert werden können.

Der wohl wichtigste Schritt in der Analyse von Anforderungen ist es demnach, diejenigen Anforderungen aus den Ursprungsanforderungen zu extrahieren, die tatsächliche Anforderungen an das System darstellen. Erst dann haben Sie nicht mehr und nicht weniger von Ihrem System gefordert, als es leisten soll. Bei dieser Tätigkeit unterscheiden wir zwei Fälle.

1. Fall: Das System realisiert nur einen Teil der Anforderung

Das wohl häufigste Defizit in den Ursprungsanforderungen besteht darin, dass dort mehr verlangt wird, als das betrachtete System wirklich leisten kann bzw. leisten soll. In der Praxis findet man immer wieder zwei Gründe dafür.

Entweder wünscht sich der Stakeholder etwas von dem neuen System, das von dem System mit den gegebenen Randbedingungen (Entwicklungszeit und -kosten bzw. Herstellungskosten des Systems) nicht realisiert werden kann. Oder der Wunsch muss in einer Zusammenarbeit zwischen dem betrachteten System und seinen Nachbarsystemen (z. B. von anderen Lieferanten verantwortet) realisiert werden. Dies deutet darauf hin, dass der Architekturschritt eine Ebene oberhalb zur Abgrenzung des Systemkontextes nicht vollständig durchgeführt oder bei der Formulierung der Anforderungen nicht korrekt berücksichtigt wurde (siehe auch [Kapitel 3 „Requirements-Engineering im Überblick“](#)). Stellen Sie sich also die Frage:

- Welchen Teil der geforderten Funktionalität soll das System realisieren?

Betrachten wir die Konsequenzen der Anforderung E3 an die Performanz in der Realisierung, so stellt sich heraus, dass dafür leistungsfähige Motoren existieren müssen, die Ihrem Kunden vielleicht zu teuer sind. Eine Alternative wäre, nur das Entriegeln der Tür und nicht das zusätzliche Öffnen der Tür zu unterstützen. Diese Änderung muss dann auch in den bereits bestehenden Anforderungen nachvollzogen werden, wodurch zuvor gefundene Anforderungen durch neue Versionen ersetzt werden.

Sobald das SHS die Erlaubnis zum Öffnen einer Tür überprüft hat und falls die Überprüfung positiv war, muss das SHS die entsprechende Tür öffnen. (E2)

Das SHS muss die Erlaubnis zum Entriegeln einer Tür überprüfen. (E1')

Sobald das SHS die Erlaubnis zum Entriegeln einer Tür überprüft hat und falls die Überprüfung positiv war, muss das SHS die entsprechende Tür entriegeln. (E2')

Die Funktion Tür entriegeln muss eine Ausführungszeit von maximal 2 Sekunden aufweisen. (E3')

Das nächste Beispiel beschäftigt sich mit der Einbeziehung von Nachbarsystemen. Hier gehen wir von der Anforderung U2 aus. Da das angesprochene Handy oder Tablet jedoch nicht Bestandteil des SHS sein soll (siehe Kapitel 6 „Ziele, Informanten und Fesseln“), folgen als tatsächliche Funktionen des Systems leicht geänderte Anforderungen. Dabei wurde auch für die implizite Performanzanforderung in U2 die Anforderung E5 definiert, um den geänderten Funktionalitäten gerecht zu werden.

Das SHS muss jede Entriegelung einer Tür auf konfigurierten Handys und Tablets innerhalb von 2 Sekunden anzeigen. (U2)

Sobald das SHS eine Tür entriegelt hat, muss das SHS die Entriegelung an die konfigurierten Handys und Tablets senden. (E4)

Das Senden muss eine Ausführungszeit von maximal 1 Sekunde aufweisen. (E5)

2. Fall: Die Anforderung bezieht sich auf eine Komponente

Ebenfalls häufig tritt die Situation auf, dass ein Stakeholder eine Anforderung stellt, die sich nur an eine Komponente des Systems richtet. Man macht sich das Leben leicht und nimmt an, dass es eine solche Komponente gibt, vergisst dabei jedoch, die Begründung für die geforderte Komponentenanforderung als Systemanforderung zu fordern. Ihre Aufgabe ist es, diese Systemanforderung zu formulieren, da jede Anforderung an eine Komponente durch eine Systemanforderung begründet werden sollte. Zusätzlich haben Sie so eine weitere Systemanforderung gefunden, die es wert sein kann, weiter beschrieben zu werden (siehe Abschnitt 12.2.5 „Anforderungen verfeinern“). Beantworten Sie für jede Anforderung an einen Teil des Systems die folgende Frage:

- Wegen welcher Systemanforderung existiert die gegebene Komponentenanforderung?

Auch hier haben wir wieder ein Beispiel. Die Anforderung U3 richtet sich an die Komponente Türsensor des Systems, woraus aber eine neue Anforderung an das System abgeleitet werden kann.

Der Türsensor muss den Status einer Tür (offen/geschlossen) melden. (U3)

Das SHS muss das Öffnen einer Tür erkennen. (E6)

Das SHS muss das Schließen einer Tür erkennen. (E7)

Die Aufgabe, die richtigen Anforderungen herauszufinden, ist auch in der agilen Entwicklung von entscheidender Bedeutung. Beim Formulieren der User-Stories sollte der Product-Owner sich schon auf das für das System Relevante konzentrieren. Dies erspart Nachfragen und Änderungen während der Refinement-Besprechungen mit dem Entwicklungsteam.

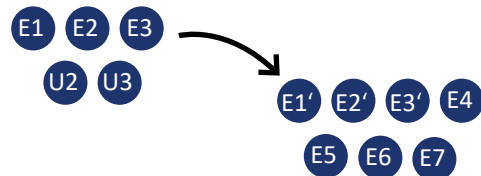


Abbildung 12.5: Ergebnis der Aufgabe
Notwendige Anforderungen extrahieren

12.2.3 Anforderungen abstrahieren

Mit dieser Aufgabe werden zu den Anforderungen, die bereits gefunden wurden, abstraktere Anforderungen definiert. Diese bilden den Ausgangspunkt für die nachfolgende Verfeinerung (siehe [Abschnitt 12.2.5 „Anforderungen verfeinern“](#)). Erst durch diesen Schritt der Abstraktion erhalten Sie eine vollständige, wenn auch grobe, Sicht auf die Anforderungen an Ihr System. Des Weiteren werden die bisher gefundenen Anforderungen dadurch in Zusammenhang gesetzt, dass die abstrakten Anforderungen eine gemeinsame Basis für sie bilden.

Für jede Anforderung können Sie sich die folgenden Fragen stellen:

- In welchem Ablauf soll das System die geforderte Funktionalität durchführen bzw. außerhalb des Systems zur Verfügung stellen?
- Bezieht sich eine geforderte Eigenschaft des Systems auf eine Funktion des Systems?
- Zu welchem Problem wird mit der geforderten Funktionalität oder Eigenschaft eine Lösung vorgegeben?

Die Antwort auf die erste Frage wird Sie zu einer neuen, abstrakteren Funktion des Systems bringen, während eine Antwort auf die zweite Frage zu einer Funktion auf einer beliebigen Abstraktionsebene führt. Eine Antwort auf die dritte Frage führt Sie zu einer nicht-funktionalen Anforderung, einer Qualitätsanforderung.

Für eine neu gefundene Anforderung können Sie nun diesen Schritt erneut durchführen, um auf eine noch abstraktere Anforderung zu kommen. Führen Sie diese Abstraktionen so oft durch, bis Sie:

- einen Use-Case erhalten haben oder
- eine der in [Kapitel 13 „Nicht-funktionale Anforderungen“](#) vorgestellten Kategorien von nicht-funktionalen Anforderungen erreicht haben.

Betrachten wir die mehrfache Durchführung dieser Abstraktion an einem Beispiel anhand der Anforderungen, die wir in den vorherigen Aufgaben gefunden haben:

Sobald das SHS die Erlaubnis zum Entriegeln einer Tür überprüft hat und falls die Überprüfung positiv war, muss das SHS die entsprechende Tür entriegeln. (E2')

Die Frage, in welchem Ablauf das SHS die Tür entriegeln soll, bringt uns zu folgender Anforderung:

Das SHS muss BewohnerInnen die Möglichkeit bieten, eine Tür ohne manuelle Bedienung zu entriegeln. (UC1)

Dies ist eine abstraktere Anforderung als E2', da das SHS mehr Schritte für die Anforderung UC1 durchführen muss, als nur die Tür zu entriegeln. Versucht man nun, die erste Fragestellung auf die neu gefundene Anforderung anzuwenden, so kann man sich viele Abläufe vorstellen, im Rahmen derer die BewohnerInnen das Haus betreten möchten

(sie kommen vom Einkaufen nach Hause, sie haben etwas zu Hause vergessen etc.). An allen diesen Abläufen wird das SHS aber „nur“ mit dem Entriegeln der Tür beteiligt sein, weswegen die gefundene Anforderung UC1 einen Use-Case und damit eine Wurzel für viele Anforderungen an das SHS darstellt.

In einem zweiten Beispiel betrachten wir als Ausgangspunkt wieder wie oben die Anforderung E2' und stellen uns nun die Frage, weshalb die entsprechenden Bedingungen dort angegeben sind. Als Antwort können wir die Anforderung E8 erhalten.

Nur Befugte dürfen eine Tür entriegeln. (E8)

Dies ist eine Anforderung, die man sofort als Sicherheitsanforderung identifizieren kann, und dementsprechend unterhalb dieser Kategorie (in der [Abbildung 12.6](#) als K1 bezeichnet) anordnen würde.

Als letztes Beispiel gehen wir von der Anforderung E3' aus:

Die Funktion Tür entriegeln muss eine Ausführungszeit von maximal 2 Sekunden aufweisen. (E3')

Die Frage nach der Funktion, auf die sich die gegebene Ausführungszeit bezieht, führt uns auf die schon vorher gefundene Funktion des Entriegelns der Tür, die in der Anforderung E2' beschrieben wurde. Hier erhalten wir also keine neue Anforderung.

Unser Tipp: Führen Sie als Product-Owner in einer agilen Entwicklung diese Aufgabe auf jeden Fall durch. Entweder Sie erreichen dann die Ebene von Epics oder Sie erhalten eine abstraktere User-Story, die als Motivation für die ursprüngliche User-Story dienen kann und die Sie danach in kleinere User-Stories zerlegen können (siehe [Abschnitt 12.2.5 „Anforderungen verfeinern“](#)). Durch die Verfeinerung erhalten Sie neue User-Stories, die die ursprüngliche User-Story sinnvoll ergänzen.

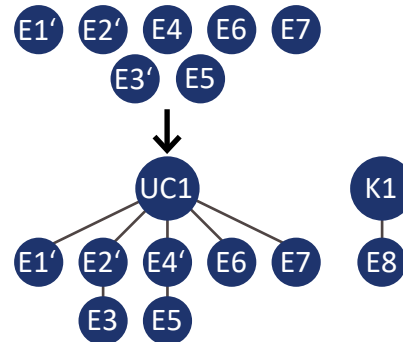


Abbildung 12.6: Ergebnis der Aufgabe Anforderungen abstrahieren

12.2.4 Fehlende Anforderungen ergänzen

Mithilfe der zuvor definierten Aufgaben wurden die Anforderungen gefunden, die sich direkt aus den Ursprungsanforderungen ableiten ließen. Diese werden nun um die Aspekte ergänzt, die nicht vom Stakeholder vorgegeben wurden, aber zum Teil implizit in den bisher gefundenen Anforderungen gefordert werden.

Auch hier werden wieder einige Fragen gestellt, die Sie zu den fehlenden Anforderungen führen:

- Mit welcher Funktion wird eine geforderte Eigenschaft eines Objekts oder eine Bedingung einer Anforderung sichergestellt?
- In welcher Funktion wird eine Ausgabe an einer Schnittstelle erzeugt?

- In welcher Funktion wird eine Eingabe an einer Schnittstelle verwendet?
- Welche weiteren Funktionen werden auf oberster Ebene benötigt?
- Welche weiteren Kategorien von nicht-funktionalen Anforderungen sind für Ihr System relevant?

Weitere Indikatoren finden Sie in Kapitel 9 „Das SOPHIST-REgelwerk“.

Während die ersten drei Fragen die impliziten Annahmen in den Anforderungen adressieren, dienen die letzten beiden Fragen zur Ermittlung der Basis- und zum Teil auch der Leistungsfaktoren Ihres Systems (siehe Kapitel 8 „Anforderungsermittlung“).

Auch hier wollen wir wieder ein paar Beispiele darstellen, um die Anwendung der Fragen zu verdeutlichen. Beginnen wir mit der Anforderung E4.

Sobald das SHS eine Tür entriegelt hat, muss das SHS die Entriegelung an die konfigurierten Handys und Tablets senden. (E4)

Wendet man die erste Frage von oben auf diese Anforderung an, so ergibt sich eine neue Anforderung, die aus der gegebenen Bedingung resultiert:

Das SHS muss eine Tür entriegeln.

Diese Anforderung entspricht einer Anforderung, die in ausführlicherer Version schon vorher definiert wurde (E2' aus Abschnitt 12.2.2 „Notwendige Anforderungen extrahieren“); Sie muss also nicht erneut aufgenommen werden. Aus der Bedingung in Anforderung E2' folgt jedoch die folgende Anforderung:

Das SHS muss die Erlaubnis zum Entriegeln einer Tür überprüfen.

Auch diese Anforderung wurde in einem vorherigen Schritt als E1' gefunden und muss dementsprechend ebenfalls nicht als neue Anforderung definiert werden.

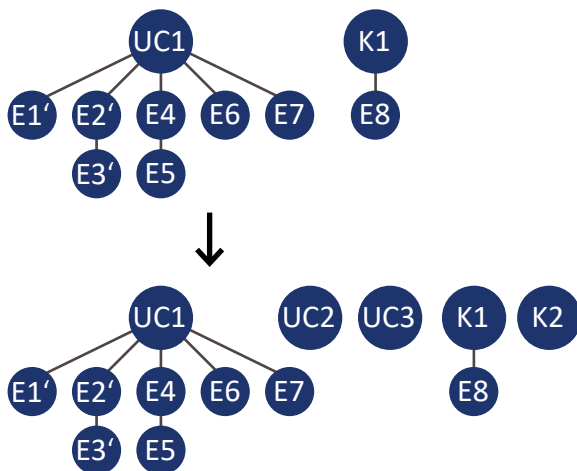


Abbildung 12.7: Ergebnis der Aufgabe
Fehlende Anforderungen ergänzen

Aus der in E4 erwähnten Konfigurierbarkeit folgt:

Das SHS muss berechtigten Personen die Möglichkeit bieten, Handys und Tablets als Empfangsgerät zu konfigurieren. (UC2)

Bitte beachten Sie, dass bei dieser Anforderung neue Informationen hinzugekommen sind, die natürlich wieder mit Ihrem Stakeholder abgestimmt werden müssen. Sie ist direkt als Use-Case definiert, da eine weitere Abstraktion aus dem Betrachtungsgegenstand des SHS herausführen würde (siehe Abschnitt 12.2.3 „Anforderungen abstrahieren“).

Als letztes Beispiel betrachten wir nun die Use-Cases, um die oberste Ebene von Anforderungen zu vervollständigen.

Aus der Anforderung, dass das SHS die Tür entriegeln soll, ergibt sich sofort die Frage, ob das SHS diese Tür nicht auch verriegeln sollte. Nach Absprache mit dem Stakeholder könnte man folgende Anforderung, wiederum als Use-Case, formulieren:

Das SHS muss eine Tür automatisch verriegeln. (UC3)

Ähnlich verhält es sich mit den Kategorien von nicht-funktionalen Anforderungen. Überprüfen Sie die bisher gefundenen Kategorien, ob nicht eine für Ihr System wichtige Kategorie bisher noch nicht gefunden wurde. Für unser SHS könnte man sich sicher Anforderungen an die Performanz oder Bedienbarkeit vorstellen.

Mit den hier erwähnten Beispielen ergibt sich zusammenfassend das Bild aus [Abbildung 12.7](#) für unsere bisher gefundenen Anforderungen. Gegenüber der [Abbildung 12.6](#) wurden lediglich zwei neue Use-Cases und eine neue Kategorie von nicht-funktionalen Anforderungen (K2: Performanzanforderungen) aufgenommen. Eigentlich könnten noch weitere Anforderungen, insbesondere Use-Cases, gefunden werden; wir haben diese aus Gründen der Übersichtlichkeit nicht weiter betrachtet.

Auch diese Aufgabe werden Sie, bewusst oder unbewusst, in einer agilen Entwicklung beim Definieren von User-Stories oder auch bei Refinement-Besprechungen durchführen. Damit identifizieren Sie z. B. neue User-Stories, die dazu dienen, in der ursprünglichen User-Story getroffene Annahmen implementieren zu lassen.

12.2.5 Anforderungen verfeinern

Kommen wir nun zu der Aufgabe mit den meisten Freiheitsgraden in der Analyse der Anforderungen, dem Verfeinern. Hier können Sie für jede bisher gefundene Anforderung (also für jeden Knoten in dem bisher entstandenen Baum) entscheiden, ob Sie diese Anforderung noch genauer beschreiben möchten. Sie sollten diesen Schritt für eine Anforderung in den folgenden Fällen nicht durchführen:

- Sie sind mit jeder fachlichen Lösung für die Anforderung zufrieden.
- Sie sind sich sicher, dass es zum allgemeinen Wissen gehört, was genau unter der Anforderung zu verstehen ist.
- Das Verb, das den in einer funktionalen Anforderung geforderten Prozess beschreibt, ist im Glossar definiert und muss deswegen nicht über eine Verfeinerung genauer beschrieben werden.

Wenn Sie sich nicht zu 100 % sicher sind, dass einer der angegebenen Gründe zutrifft, sollten Sie diese Anforderung verfeinern. Das wird jedoch zu einem sehr hohen Aufwand in der Analyse führen. Diesen Aspekt werden wir in [Abschnitt 12.3 „Angemessener Einsatz der Tätigkeiten“](#) betrachten. Falls Sie die erste Frage zwar verneinen müssen, Sie jedoch mit vielen, aber nicht allen Lösungen einverstanden sind, so können Sie als Verfeinerung einer Anforderung auch Lösungen mithilfe des entsprechenden Modaloperators ausschließen (siehe [Kapitel 3 „Requirements-Engineering im Überblick“](#)).

Stellen Sie sich die folgenden Fragen, um eine Anforderung zu verfeinern:

- Wie ist der Ablauf der Interaktion zwischen Personen oder Nachbarsystemen und dem betrachteten System bzw. welche Schritte muss das System für eine geforderte Funktionalität durchführen?
- Wie kann das System eine geforderte nicht-funktionale Eigenschaft sicherstellen?

Für das Finden einer Verfeinerung hat es sich als zielführend herausgestellt, zunächst ungeordnet alle die Anforderungen zu betrachten, die Ihnen bei der zu verfeinernden Anforderung wichtig sind. Bringen Sie die Anforderungen in eine Ordnung (z. B. einen Ablauf) und erweitern Sie diese um fehlende Anforderungen. So erhalten Sie eine vollständige Verfeinerung der eingangs betrachteten Anforderung. In der Praxis hat es sich bewährt, bestimmte Aspekte (z. B. eine Fehlerbehandlung) bei dieser Verfeinerung nicht zu betrachten, sondern gesondert und funktionsübergreifend zu beschreiben.

Als erstes Beispiel betrachten wir den identifizierten Use-Case UC1. Auch wenn schon einige Anforderungen als Verfeinerung gefunden wurden, so wird eine ausführliche Betrachtung weitere Anforderungen liefern. Als Ergebnis könnte ein Ablaufdiagramm wie in Kapitel 18 „Anforderungen modellieren“ entstehen, wobei die dort dargestellten Aktionen als Anforderung interpretiert werden. Für diese Anforderungen kann dann eine erneute Verfeinerung notwendig sein. Betrachten wir hierzu wieder die Anforderung E1’.



Das SHS muss die Erlaubnis zum Entriegeln einer Tür überprüfen. (E1’)

Falls Sie sich noch nicht auf eine Lösung festlegen wollen, aber eine Authentifizierung über einen PIN-Code ausschließen wollen, so können Sie formulieren:

Die Authentifizierung zum Entriegeln der Tür darf nicht über die Eingabe eines PIN-Codes gelöst werden. (E9)

Herausfordernder wird die Verfeinerung von nicht-funktionalen Anforderungen. Es ist im Allgemeinen ein iterativer Prozess, alle Anforderungen zu formulieren, die eine sehr abstrakte Anforderung verfeinern. Ein Ansatz ist es, die Auswirkungen einer nicht-funktionalen Anforderung bei der Verfeinerung von funktionalen Anforderungen zu betrachten. Zusätzlich können weitere technologische Anforderungen an das System folgen. Als Beispiel hierzu betrachten wir eine allgemeine Anforderung an die Bedienbarkeit.

Die Nutzung des SHS muss so einfach wie möglich sein. (U4)

Diese Anforderung lässt jede Menge Spielraum, den es festzulegen gilt. Eine Auswirkung dieser Anforderung haben wir schon in E9 angegeben, da wir keine Eingabe eines komplizierten Pin-Codes haben wollen. Darüber hinaus kann man eine allgemeine Anforderung als Vorgabe für die Realisierung formulieren:

Das Ein- und Ausschalten des Lichts muss in jedem Raum über einen Tastschalter realisiert werden. (E10)

In einer agilen Entwicklung entspricht dieser Schritt dem Aufteilen einer User-Story, z. B. ausgelöst durch die Diskussionen mit dem Entwicklungsteam während einer Refinement-Besprechung. Anstelle der einen ursprünglichen User-Story werden zwei oder mehr User-Stories in die Entwicklung übergeben (siehe auch Abschnitt 17.2.1 „Verschiedene Detaillierungsebenen von User-Stories“).

12.2.6 Anforderungen verbessern

Mit den bisher vorgestellten Aufgaben haben wir aus den vorgegebenen Anforderungen eine Vielzahl von Systemanforderungen abgeleitet. In dieser letzten Aufgabe geht es nun darum, jede der zuvor gefundenen Anforderungen zu überprüfen und gegebenenfalls zu verbessern, um die Qualität jeder einzelnen Anforderung sicherzustellen.

Betrachten Sie jede einzelne Anforderung und stellen Sie sich die folgenden Fragen:

- Ist es eindeutig, wann ein Test einer Anforderung positiv oder negativ bewertet wird?
- Sind die in einer Anforderung verwendeten Begriffe durchgängig verwendet und in einem Glossar definiert?
- Existieren überflüssige Anteile in der Anforderung?
- Sind die Bedingungen in einer Anforderung vollständig und korrekt?

Gerade die letzte Frage führt zu einer Änderung in einer Anforderung, wodurch es notwendig sein wird, die zuvor vorgestellten Aufgaben für diese Anforderung erneut durchzuführen. Betrachten wir hierzu als Beispiel eine zuvor ermittelte Anforderung.

Sobald das SHS die Erlaubnis zum Entriegeln einer Tür überprüft hat und falls die Überprüfung positiv war, muss das SHS die entsprechende Tür entriegeln. (E2')

Diese erscheint zunächst vollständig, jedoch sollte man als Bedingung mit aufnehmen, dass die Tür auch verriegelt ist, sodass die Entriegelung nicht zu einem Fehler führt. Es folgt die verbesserte Anforderung E2“:

Sobald das SHS die Erlaubnis zum Entriegeln einer Tür überprüft hat und falls die Überprüfung positiv war und falls die Tür verriegelt ist, muss das SHS die entsprechende Tür entriegeln. (E2“)

Für den neu hinzugefügten Teil in der Anforderung sollten Sie nun wieder die Aufgabe *Fehlende Anforderungen ergänzen* durchführen. Nur damit stellen Sie sicher, dass das SHS auch den Verriegelungsstatus überprüft.

Sowohl die Frage nach der Korrektheit der Bedingung als auch die Frage nach der eindeutigen Verwendung von Begriffen bringt uns dazu, noch einmal über die folgende Anforderung nachzudenken:

Das SHS muss BewohnerInnen die Möglichkeit bieten, eine Tür ohne manuelle Bedienung zu entriegeln. (UC1)

Mit dieser Anforderung würde es z. B. Dieter, dem Vater von Lina, nicht gestattet sein, diese Funktion des SHS zu verwenden. An dieser Stelle sollte man also nicht nur den BewohnerInnen, sondern besser „autorisierten Personen“ die automatische Entriegelung erlauben. Weiterhin kann man den Begriff Tür hinterfragen. Die Entriegelung wird sich normalerweise nur auf eine Haustür und nicht auf jede Tür des Hauses beziehen. Somit ändert sich der obige Use-Case zu:

Das SHS muss autorisierten Personen die Möglichkeit bieten, eine Haustür ohne manuelle Bedienung zu entriegeln. (UC1')

Überflüssige Anteile in einer Anforderung lassen sich häufig zwei Gruppen zuordnen: Zum einen beschreiben sie Begründungen für den eigentlich geforderten Teil und zum anderen werden Erläuterungen gegeben. Beide Anteile sollten Sie von der eigentlichen Anforderung separieren, indem Sie diese z. B. als Kommentar zu der Anforderung beschreiben.

Wir könnten an dieser Stelle noch viele weitere Beispiele zeigen, verweisen aber stattdessen auf [Kapitel 9 „Das SOPHIST-REgelwerk“](#), das die Verbesserung einzelner Anforderungen im Fokus hat. Zusätzlich hilft Ihnen die Formulierung nach der MAS-TER-Schablone (siehe [Kapitel 19 „Schablonen für Anforderungen und User-Stories“](#)), die Qualität einer Anforderung zu verbessern.

12.3 Angemessener Einsatz der Tätigkeiten

In dem vorigen Abschnitt haben wir Ihnen die einzelnen Aufgaben einer Analyse von Anforderungen erläutert, um diese bezüglich einiger der in [Kapitel 3 „Requirements-Engineering im Überblick“](#) eingeführten Qualitätskriterien zu verbessern. Dabei haben wir keine Rücksicht auf den dazu benötigten Aufwand genommen. Stattdessen sind wir davon ausgegangen, dass Sie eine Spezifikation erstellen wollen, die zu 100 % die hier betrachteten Qualitätskriterien erfüllt.

In der Realität eines Projekts ist es allerdings selten der Fall, dass Sie die Zeit zur Verfügung haben, alle Qualitätskriterien zu 100 % zu erfüllen. Deshalb geben wir Ihnen in diesem Abschnitt einige Tipps, wie Sie die Aufgaben gemäß Ihrer Aufgabenstellung priorisieren können, um eine für Sie optimale Anforderungsspezifikation zu erhalten.



Die Anforderungsanalyse erzeugt aus den Ursprungsanforderungen die Systemanforderungen in der benötigten Qualität.

Dazu werden wir zunächst eine Zuordnung von den Qualitätskriterien zu den einzelnen Aufgaben angeben. Damit können Sie selbst entscheiden, welche Aufgaben Sie bis zu welchem Grad durchführen möchten. Danach schlagen wir ein Vorgehen für die Analyse von Anforderungen vor, mit dem Sie zielgerichtet Ihre Ressourcen einsetzen können.

12.3.1 Die richtige Qualität erzeugen

Das Ziel bei der Analyse der Ursprungsanforderungen muss es sein, aus diesen Eingaben belastbare Systemanforderungen herzuleiten.

Oder anders formuliert: Mit jeder der zuvor vorgestellten Aufgaben werden die Anforderungen bezüglich einiger Qualitätskriterien verbessert. Wir können also eine Tabelle angeben, die die Aufgaben mit den Kriterien in Verbindung setzt. Nach der Festlegung, welche Qualitätskriterien für Sie besonders wichtig sind, können Sie damit dann auswählen, welche Aufgabe Sie fokussiert durchführen sollten und in welche Aufgabe Sie weniger Aufwand investieren können.

In der Tabelle aus [Abbildung 12.8](#) bedeutet ein Kreuz in einer Zelle, dass die entsprechende Aufgabe das jeweilige Qualitätskriterium unterstützt. Zum besseren Verständnis wird dieser Zusammenhang im Einzelnen vorgestellt.

	Anforderungen separieren	Notwendige Anforderungen extrahieren	Anforderungen abstrahieren	Fehlende Anforderungen ergänzen	Anforderungen verfeinern	Anforderungen verbessern
Notwendig	X	X				X
Eindeutig	X				X	X
Vollständig	X					X
Vollständige Spezifikation			X	X	X	
Klare Struktur	X		X		X	
Konsistente Spezifikation	X		X		X	
Angemessenheit	X		X	X	X	

Abbildung 12.8: Zuordnung der Aufgaben zu Qualitätskriterien

Notwendige Anforderungen erhält man in erste Linie dadurch, dass man die Anforderung für den richtigen Betrachtungsgegenstand formuliert. Aber auch die überflüssigen Anteile zu eliminieren hilft Ihnen, sich auf das Notwendige in einer Anforderung zu fokussieren. Indirekt hilft dabei auch das Separieren von Anforderungen, da Sie dann die einzelnen Teile getrennt voneinander auf ihren Betrachtungsgegenstand untersuchen können.

Die **Eindeutigkeit** erreichen Sie durch die Verwendung einheitlicher Begriffe sowie durch das Separieren von Anforderungen, da hierdurch etwa die Bedingungen eindeutig einer Funktionalität zugeordnet werden müssen. Bei den Anforderungen, die nur sehr schwer eindeutig zu beschreiben sind (z. B. viele nicht-funktionale Anforderungen), hilft eine Ersetzung durch die verfeinerten Anforderungen, um sie testbar zu beschreiben.

Die **Vollständigkeit einer Anforderung** wird durch eine für einzelne Funktionen getrennte Betrachtung der Bedingungen unterstützt. Die abschließende Verbesserung einer Anforderung hilft, fehlende Bedingungen zu identifizieren.

Die **Vollständigkeit der gesamten Spezifikation** wird in erster Linie durch die Verfeinerung und Definition von fehlenden Anforderungen gegeben. Die Verfeinerung wird durch Bilden der Abstraktionen unterstützt, da hierdurch unter anderem die Basis für die Vervollständigung auf der obersten Ebene der Anforderungen gelegt wird.

Durch die Abstraktion und Verfeinerung wird auch eine **klare Strukturierung** der Anforderungen ermöglicht, da hierdurch der inhaltliche Zusammenhang der Anforderungen als Struktur der Spezifikation dienen kann (siehe auch den STABLE-Ansatz in Kapitel 21 „Strukturen und Zustände“). In zweiter Linie dient auch die Separierung von Anforderungen diesem Ziel, da die unterschiedlichen Anteile einer Ursprungsanforderung in unterschiedliche Teile der Struktur einsortiert werden können.

Die Herstellung der **Konsistenz der Anforderungen** untereinander wird unterstützt durch das Bilden des Zusammenhangs zwischen den Anforderungen im Rahmen der Abstraktion und Verfeinerung. Darüber hinaus hilft die Verwendung einheitlicher Begriffe Inkonsistenzen zu entdecken, die dann aufgelöst werden können. Aber auch die Separierung führt dazu, die Teile einer gegebenen Anforderung auf Konsistenz zu anderen Anforderungen zu prüfen.

Die **Angemessenheit der Spezifikation** wird durch das Zusammenspiel von Abstraktion und Verfeinerung unterstützt, da Sie dadurch die Möglichkeit haben, für jede Anforderung zu entscheiden, ob Sie an dieser Stelle noch mehr Aussagen treffen müssen oder nicht. Da der Aufwand für die Anforderungsanalyse gerade hierdurch primär bestimmt wird, wollen wir diese Entscheidung in dem nachfolgenden Abschnitt gesondert betrachten.

12.3.2 Was wirklich benötigt wird

Nachdem wir Ihnen die einzelnen Aufgaben anhand von Beispielen erläutert haben und deren Bezug zur Qualitätsverbesserung Ihrer Systemanforderungen beschrieben haben, geben wir Ihnen nun Vorgehen für die Anwendung in der Praxis an die Hand. Dazu möchten wir **Szenario 1 „Kundenanfrage bearbeiten“** und **Szenario 3 „Subunternehmen beauftragen“** aus Kapitel 4 „RE ist nicht gleich RE“ verwenden.

Im zuerst genannten Szenario antwortet die Firma Schlauhause mit einem Angebot in Form eines Pflichtenheftes auf die Anfrage von Familie Meyer. In diesem Fall sind die Ursprungsanforderungen in einem Lastenheft enthalten. Im **Szenario 3 „Subunternehmen beauftragen“** dagegen erstellt Schlauhause ein eigenes Lastenheft für eine Ausschreibung. Dementsprechend handelt es sich bei den Ursprungsanforderungen um Anforderungen von den Stakeholdern aus der Firma Schlauhause.

Für die Analyse der Anforderungen gehen wir in beiden Fällen davon aus, dass eine Spezifikation erstellt und eine Nachvollziehbarkeit zu den entsprechenden Eingaben (also Stakeholder- oder Lastenheftanforderungen) hergestellt werden soll, wie es in einem

regulierten Umfeld oder von vielen Entwicklungsprozessen verlangt wird. Daraus lassen sich die folgenden zwei Qualitätsstufen für eine Spezifikation ableiten:

- **Minimale** Spezifikation: Hier werden genau die Anforderungen definiert, deren Begründungen sich von den Ursprungsanforderungen ableiten lassen. Die hier beschriebenen Anforderungen beschreiben nur das Nötigste und lassen der weiteren Entwicklung, zum Beispiel dem Auftragnehmer, sehr viele Freiheiten.
- **Vollständige** Spezifikation: Aufbauend auf einer Spezifikation mit minimaler Qualität (minimale Spezifikation) werden weitere Anforderungen definiert, die den Gesamtumfang des zu entwickelnden Systems beschreiben (vollständige Spezifikation).

Sie können auch die Erstellung der minimalen Spezifikation als Pflichtprogramm ansehen, da sie das Mindeste darstellt, das in einer Anforderungsanalyse entstehen sollte. Auf dem Weg zur vollständigen Spezifikation haben Sie dann die Wahl, wie viel Aufwand Sie investieren bzw. welchen Grad von Vollständigkeit Sie erreichen möchten. Demnach wird sich Ihre Spezifikation in Ihrem Projekt irgendwo zwischen diesen beiden Extrema bewegen.

Erzeugung einer minimalen Spezifikation

Wie eingangs erwähnt, möchten wir zunächst die Anforderungen finden, die ihre Basis in den Anforderungen haben, die Ihnen als Requirements-Engineer als Eingabe, z. B. von ihren Stakeholdern oder in einem Lastenheft, vorliegen. Die einfachste Art zur Erzeugung einer minimalen Spezifikation wäre es, die Ursprungsanforderungen einfach zu übernehmen. Um die weitere Bearbeitung der Anforderungen zu unterstützen, wollen wir allerdings die Qualität der Anforderungen in den Bereichen *Notwendigkeit* und *Eindeutigkeit* steigern. Des Weiteren soll die Spezifikation *klar strukturiert* sein, auch um die Erweiterung der minimalen Spezifikation hin zu einer vollständigen Spezifikation zu ermöglichen.

Die hierfür benötigten Tätigkeiten bei der Analyse der Anforderungen sind in [Abbildung 12.9](#) dargestellt. Bitte beachten Sie, dass wir hier weder die Schritte zur Dokumentation noch zur Prüfung der gefundenen Anforderungen adressiert haben. Wir beziehen uns nur auf die zuvor definierten Aufgaben.

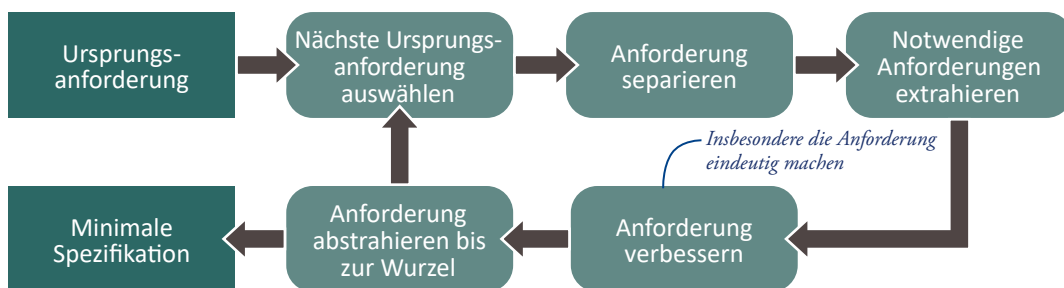


Abbildung 12.9: Erzeugung einer minimalen Spezifikation

Bitte beachten Sie auch, dass Sie auf jeden Fall die Aufgabe *Notwendige Anforderungen extrahieren* durchführen sollten. Bei den nachfolgenden Aufgaben können Sie bei Bedarf Abstriche machen.

Wenn Sie die dargestellten Schritte für jede der Ursprungsanforderungen durchgeführt haben, so werden Sie schon ein gutes Gefühl haben, was Ihre Stakeholder von dem zu entwickelnden System erwarten.

Erweiterung zu einer vollständigen Spezifikation

In dieser zweiten Phase wollen wir nun die zuvor erzeugte minimale Spezifikation zu einer vollständigen Spezifikation erweitern. Die Ursprungsanforderungen müssen nun nicht mehr betrachtet werden, da sie in der Phase zuvor vollständig in neue Anforderungen überführt wurden.

Zunächst sollten Sie sowohl die Use-Cases als auch die Kategorien nicht-funktionaler Anforderungen auf Vollständigkeit überprüfen. Danach können Sie mit einem iterativen Prozess starten: Wählen Sie die risikoreichste Anforderung aus und führen Sie die angegebenen Schritte durch. Danach haben Sie weitere Anforderungen gefunden, die nun für die weitere Bearbeitung zur Verfügung stehen.



Abbildung 12.10: Erzeugung einer vollständigen Spezifikation

Mit dem hier angegebenen Vorgehen sind Sie in der Lage, Ihre Ressourcen zur Anforderungsanalyse zielgerichtet zu investieren und immer wieder bewusst zu entscheiden, an welcher Stelle Ihrer Anforderungsspezifikation Sie Aufwand zur Qualitätssteigerung investieren möchten.