

## Unsere Familie Meyer



Um die Inhalte des vorliegenden Buches für Sie nachvollziehbarer zu machen, haben wir die teils theoretischen Inhalte mit Beispielen hinterlegt. Diese beziehen sich auf die Entwicklung eines Smart-Home-Systems (SHS), das hier kurz eingeführt werden soll.

Familie Meyer möchte mehr Zeit für sich haben und plant deswegen, ihr Haus zu einem Smart Home aufzurüsten. Robert Meyer, seine Frau Lina und ihre Tochter Johanna haben viel diskutiert, um herauszufinden, was denn in ihrem neuen Zuhause für sie wichtig ist. Schnell hat sich herausgestellt, dass die Sicherheit ein großes Thema ist. Aber sie möchten nicht nur eine konventionelle Alarmanlage installieren. Ihr Haus soll mehr können. Sie stellen sich z.B. die Frage, wie das Haus zuverlässig erkennen kann, ob ein Einbrecher im Haus ist oder doch nur Robert nachts den Kühlschrank plündern will. Und kann das Haus im Ernstfall nicht direkt die Polizei benachrichtigen?

Ein weiteres wichtiges Thema für sie ist der Zugang zu ihrem Haus. So wollen Lina und Robert vermeiden, mit vollen Einkaufstaschen bepackt nach ihrem Haustürschlüssel suchen zu müssen. Auch ihre Kinderbetreuung und der Vater von Robert sollen einen personalisierten Zugang zu dem Haus erhalten.

Aber die Liste geht noch weiter. Klima- und Lichtsteuerung sollten doch machbar sein. Und können sie nicht noch mehr Energiekosten einsparen? Da die Anschaffung eines Elektroautos ansteht, könnte man doch eine Photovoltaik-Anlage installieren?

Mit diesen und vielen weiteren Ideen und Wünschen gehen Lina und Robert zur Firma Schlauhause, die ihnen ihr Haus aufrüsten soll. Doch zunächst wollen Lina und Robert natürlich wissen, was denn alles im Rahmen ihres Budgets realisierbar ist, was sie eventuell vergessen haben und welche tollen Überraschungen die Firma Schlauhause für sie bereithält.

Man kann der Familie Meyer nur wünschen, dass die Firma Schlauhause das vorliegende Buch gelesen hat, weil deren erste Aufgabe darin besteht, das zu tun, was wir als Requirements Engineering bezeichnen.

# 1 Einleitung und Grundlagen

Die Bedeutung des Requirements Engineering für die erfolgreiche, den Kunden<sup>1</sup> zufriedenstellende Entwicklung von Systemen ist mittlerweile nicht mehr zu übersehen. In der Praxis ist es üblich, einen entsprechenden Aufwand für das Requirements Engineering einzuplanen. Der Requirements Engineer ist dementsprechend meist eine eigenständige Rolle im Projekt mit anspruchsvollen Tätigkeiten.

## 1.1 Requirements Engineering – Was?

Um ein Entwicklungsprojekt zum Erfolg zu führen, muss zunächst bekannt sein, was die Anforderungen an das System sind.

### **Definition 1-1:** *Anforderung*

Eine Anforderung ist:

- (1) Ein notwendiges Bedürfnis eines Stakeholders.
- (2) Eine Fähigkeit oder Eigenschaft, die ein System erfüllen muss.
- (3) Eine dokumentierte Repräsentation eines Bedürfnisses, einer Fähigkeit oder Eigenschaft.

Übersetzt<sup>a</sup> aus [IREB-Glossar 2020]

- a. Die Begriffe sind im IREB-Glossar sowohl in Deutsch als auch in Englisch angegeben, die Definition aber nur in Englisch.

Stakeholder (Interesseneigner) ist einer der zentralen Begriffe im Requirements Engineering. Stakeholder dienen u. a. als wichtigste Quellen für

*Die Bedeutung der Stakeholder*

- 
1. Wir verwenden im Buch überwiegend die männliche Form, wir wollen jedoch damit Frauen und alle anderen Geschlechter selbstverständlich nicht ausschließen bzw. ausgrenzen. Da dieses Buch zur Prüfungsvorbereitung dienen soll und der Lehrplan sowie das Handbuch nicht gegendert sind, haben wir uns dazu entschlossen, auf die Verwendung anderer Begriffe zu verzichten. Wir haben uns jedoch bemüht, in den Beispielen der Vielfalt an Menschen Rechnung zu tragen.

Anforderungen, und das Übersehen eines Stakeholders hat häufig zur Konsequenz, dass die ermittelten Anforderungen an das System lückenhaft sind [Macaulay 1993]. Stakeholder sind also alle diejenigen Personen oder Organisationen, die Anforderungen in irgendeiner Weise beeinflussen. Das können natürliche Personen sein, die das System später nutzen werden (z.B. Nutzer oder Administrator), und natürliche Personen, die Interesse an dem System haben, es aber nicht nutzen werden oder sollen (z.B. das Management oder Hacker, vor denen man das System schützen muss). Stakeholder können aber auch juristische Personen oder Institutionen sein, die allerdings durch natürliche Personen vertreten werden müssen, die die Anforderungen des betrachteten Systems beeinflussen bzw. definieren können.

**Definition 1-2:** *Stakeholder (Interesseneigner)*

Ein Stakeholder ist eine Person oder Organisation, die Einfluss auf die Anforderungen des Systems hat oder auf die das System Auswirkungen hat.

Übersetzt aus [IREB-Glossar 2020]

Anforderungen werden für eine Vielzahl von Dingen erhoben, die im Requirements Engineering häufig alle vereinfacht als Systeme bezeichnet werden. Hierzu zählen:

■ *Dem Kunden zur Verfügung gestellte Produkte*

Hierunter werden häufig klassische Softwaresysteme verstanden, die an den Kunden ausgeliefert werden. Daneben finden sich vermehrt Systeme mit einem Softwareanteil, die aber auch Hardware oder mechatronische Bestandteile umfassen.

■ *Dem Kunden zur Verfügung gestellte Dienstleistungen*

Neben den zuvor angesprochenen Systemen, die als Produkt an den Kunden ausgeliefert werden, zählen hierzu beispielsweise Services, die zur Verfügung gestellt werden.

■ *Andere Arbeitsergebnisse*

Neben Softwareprodukten und Dienstleistungen können Stakeholder aber auch Anforderungen an andere Systeme definieren. Dies können u.a. Arbeitsprodukte, Geräte, Hardwarebauteile oder beliebige Subsysteme oder Komponenten eines Systems umfassen, die benötigt werden, um ein bestimmtes Ziel der Stakeholder zu erreichen.

■ *Zusammensetzungen oder Bestandteile der oben genannten Dinge.*  
Also beispielsweise Komponenten oder sonstige Zulieferungen für Systeme des Kunden.

Dem Requirements Engineering im Entwicklungsprozess kommt die Aufgabe zu, die Anforderungen der Stakeholder zu ermitteln, zweckmäßig zu dokumentieren, zu überprüfen und abzustimmen sowie die dokumentierten Anforderungen über den gesamten Lebenszyklus des Systems hinweg zu verwalten.

*Ziel des Requirements Engineering*

**Definition 1-3:** *Requirements Engineering*

Das Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit dem Ziel, die Wünsche und Bedürfnisse der Stakeholder zu verstehen und die Gefahr zu minimieren, ein System auszuliefern, das diese Wünsche und Bedürfnisse nicht erfüllt.

Übersetzt aus [IREB-Glossar 2020]

Im Allgemeinen unterscheidet man zwischen drei Arten von Anforderungen:

- *Funktionale Anforderungen*
- *Qualitätsanforderungen*
- *Randbedingungen (Constraints)*

*Funktionale Anforderungen* legen die Funktionalität fest, die das geplante System zur Verfügung stellen soll. Sie werden typischerweise in Funktions-, Verhaltens- und Strukturanforderungen unterteilt.

**Definition 1-4:** *Funktionale Anforderung*

Eine funktionale Anforderung ist eine Anforderung bezüglich des Ergebnisses oder des Verhaltens, das von einer Funktion des Systems bereitgestellt werden soll.

Übersetzt aus [IREB-Glossar 2020]

*Qualitätsanforderungen* legen gewünschte Qualitäten des zu entwickelnden Systems fest und beeinflussen häufig – in größerem Umfang als die funktionalen Anforderungen – die Gestalt der Systemarchitektur. Typischerweise beziehen sich Qualitätsanforderungen auf die Performance, die Verfügbarkeit, die Zuverlässigkeit, die Skalierbarkeit oder die Portabilität des betrachteten Systems.

**Definition 1-5: Qualitätsanforderung**

Eine Qualitätsanforderung ist eine Anforderung, die sich auf ein Qualitätsmerkmal bezieht, das nicht durch funktionale Anforderungen abgedeckt wird.

Übersetzt aus [IREB-Glossar 2020]

Zusammenhang zwischen  
funktionalen  
Anforderungen und  
Qualitätsanforderungen

Qualitätsanforderungen stehen häufig mit verschiedenen funktionalen Anforderungen in Beziehung. Qualitätsanforderungen können etwa funktionale Anforderungen weiter konkretisieren oder ihre Umsetzung wird durch funktionale Anforderungen beschrieben. Dennoch empfiehlt es sich, Qualitätsanforderungen und funktionale Anforderungen getrennt voneinander zu spezifizieren. Zudem sollte der Bezug von Qualitätsanforderungen und funktionalen Anforderungen nachvollziehbar dokumentiert werden, um eine größtmögliche Nachvollziehbarkeit zu erreichen.

**Exkurs: Qualitätsanforderungen**

In der Praxis werden die Qualitätsanforderungen eines Systems häufig nur unzureichend dokumentiert und zwischen Stakeholdern abgestimmt. Dies gefährdet den Projekterfolg oder die spätere Akzeptanz des entwickelten Systems erheblich. Der Requirements Engineer sollte daher im Entwicklungsprozess eines Systems möglichst frühzeitig besonderes Augenmerk auf die Ermittlung, Dokumentation und Abstimmung der Qualitätsanforderungen legen.

Typischerweise werden sehr unterschiedliche Qualitäten eines Systems der Anforderungsart »Qualitätsanforderung« zugeordnet. Um auf strukturierte Art und Weise mit den Qualitätsanforderungen eines Systems umgehen zu können, wurden verschiedenste Kategorisierungen von Qualitätsanforderungen vorgeschlagen. Beispielsweise schlägt der Standard [ISO/IEC 25010:2011] eine Kategorisierung für Qualitätsanforderungen vor, die als Standardstruktur für die Dokumentation von Qualitätsanforderungen und als Checkliste zur Ermittlung und Überprüfung von Qualitätsanforderungen dienen kann. Typische Kategorien dieses Standards sind:

- Anforderungen, die die *Leistung* des Systems definieren, insbesondere das Antwortzeitverhalten und der Ressourcenverbrauch.
- Anforderungen, die die *Sicherheit* des Systems definieren, beispielsweise die Nachweisbarkeit, Authentizität, Vertraulichkeit und Integrität, aber auch der Schutz des Nutzers vor körperlichen Schäden.

- Anforderungen, die die *Zuverlässigkeit* der Funktionalität des Systems definieren, insbesondere in Bezug auf Verfügbarkeit, Fehlertoleranz und Wiederherstellbarkeit.
- Anforderungen, die die *Benutzbarkeit* des Systems definieren, insbesondere in Bezug auf Barrierefreiheit, Erlernbarkeit und Bedienbarkeit.
- Anforderungen, die die *Änderbarkeit (Wartbarkeit)* des Systems definieren, insbesondere in Bezug auf die Wiederverwendbarkeit, Analysierbarkeit, Modifizierbarkeit und Prüfbarkeit.
- Anforderungen, die die *Übertragbarkeit* des Systems definieren, insbesondere in Bezug auf Anpassbarkeit, Installierbarkeit und Austauschbarkeit.

Anforderungen einiger der o.g. Kategorien werden vorherrschend für einzelne Funktionen definiert (z.B. Leistung, Zuverlässigkeit), andere meist für das gesamte System gefordert (z.B. Änderbarkeit, Übertragbarkeit). Qualitätsanforderungen werden gegenwärtig meist in natürlicher Sprache formuliert, da eine formale Beschreibung nur mit viel Aufwand und Expertise durchgeführt werden kann. In einigen Fällen, beispielsweise bei der Entwicklung sicherheitskritischer Systeme, werden aber auch formale Beschreibungen eingesetzt, um eine höhere Nachweisbarkeit zu erreichen. So können Qualitätsanforderungen in Form von Modellen bzw. als Erweiterungen gängiger Modellierungsansätze dokumentiert werden.

Der Requirements Engineer sollte sicherstellen, dass die Qualitätsanforderungen möglichst objektiv an dem entwickelten System überprüfbar sind. Dies erfordert in der Regel, dass die geforderten Qualitäten durch quantitative Angaben konkretisiert werden. Beispielsweise könnte eine Qualitätsanforderung in Bezug auf die geforderte *Leistung* des Systems festlegen, dass die Abarbeitung einer Anfrage auf keinen Fall mehr als 4 Sekunden in Anspruch nehmen darf. Qualitätsanforderungen können hierbei durch zusätzliche funktionale Anforderungen konkretisiert werden. Oder eine Qualitätsanforderung bezüglich der Sicherheit des Systems kann durch die Forderung nach der Verschlüsselung der Ausgabedaten präzisiert und verfeinert werden. Die geforderte Verschlüsselung der Ausgabedaten stellt dabei eine funktionale Anforderung dar, die die geforderten Sicherheitseigenschaften des Systems umsetzt.

*Constraints oder Randbedingungen* (auch: Rahmenbedingungen) können von den Projektbeteiligten nicht beeinflusst werden. Randbedingungen können sich sowohl auf das betrachtete System beziehen (z. B. »Das System soll durch Webservices realisiert werden«) als auch auf den Entwicklungsprozess des Systems (z. B. »Das System soll bis spätestens Mitte 2024 am Markt verfügbar sein«). Randbedingungen werden, im Gegensatz zu funktionalen Anforderungen und Qualitätsanforderungen, nicht umgesetzt, sondern schränken die Umsetzungsmöglichkeiten, d. h. den Lösungsraum im Entwicklungsprozess, ein.

**Definition 1–6:** *Randbedingung (Constraint)*

Eine Randbedingung ist eine Anforderung, die den Lösungsraum jenseits dessen einschränkt, was notwendig ist, um die funktionalen Anforderungen und die Qualitätsanforderungen zu erfüllen.

Übersetzt aus [IREB-Glossar 2020]

Eine Randbedingung ist also in der Regel eine organisatorische oder technologische Vorgabe, die die Art und Weise einschränkt, wie das betrachtete System realisiert werden kann.

**Exkurs: Randbedingungen (Constraints)**

Um Randbedingungen näher zu betrachten und zu unterscheiden, existieren verschiedene Klassifikationen. In [Robertson und Robertson 2012] werden Randbedingungen in technische Einschränkungen und Einschränkungen des Entwicklungsprozesses unterschieden.

Des Weiteren können Randbedingungen auch im Hinblick auf ihren Ursprung unterschieden werden. So existieren beispielsweise:

- Einschränkungen aufgrund des kulturellen Umfelds
- Rechtliche Rahmenbedingungen
- Organisatorische Einschränkungen
- Einschränkungen aufgrund der physikalisch technischen Umgebung des zu entwickelnden Systems
- Rahmenbedingungen, die sich aus dem definierten Vorgehen im Projekt ergeben

Eine derartige Strukturierung von Randbedingungen hilft beim Erheben von Anforderungen, aber auch bei ihrer Verwaltung.

### Kernfakten 1-1: Requirements Engineering – Was



[www.cpre-buch.de/pk1w1](http://www.cpre-buch.de/pk1w1)



## 1.2 Requirements Engineering – Warum?

Requirements Engineering trägt dazu bei, dass eine Problemstellung früher und besser verstanden wird. Dies ermöglicht es, die Kosten bei einem späteren Aufdecken von Fehlern zu minimieren. Außerdem legen gute Anforderungen die Grundlage, um nachzuweisen, dass das System die Anforderungen wie gewünscht realisiert. Hierzu werden direkt aus den Anforderungen Testfälle gewonnen, die es ermöglichen, Defekte im System aufzudecken.

*Problemverständnis  
schärfen und Testfälle  
ableiten*

Anforderungen leisten daneben einen nicht zu unterschätzenden Beitrag zur Projektplanung. Wie wir später sehen werden, erlauben explizit dokumentierte Anforderungen eine Priorisierung und damit eine Projektplanung vorzunehmen. Anhand dieser Priorisierungen und Planungsdokumente lassen sich dann, basierend auf dem zu erwartenden Aufwand für die Umsetzung der Anforderungen, die Kosten der einzelnen Arbeitspakete und damit des gesamten Entwicklungsprojekts besser schätzen.

*Anforderungen zur  
Projektplanung und  
Kostenschätzung*

Zusammengefasst bietet Requirements Engineering laut [IREB-Lehrplan 2020] einen Mehrwert bei der Entwicklung und Weiterentwicklung eines Systems, indem:

- das Risiko, ein falsches System zu entwickeln, verringert wird;
- ein besseres Verständnis des Problems erzeugt wird;
- die Grundlage für die Schätzung von Entwicklungsaufwand und Kosten gelegt wird;
- die Voraussetzung für das Testen des Systems geschaffen wird.

Symptome für mangelhaftes Requirements Engineering sind ebenso zahlreich wie ihre Ursachen. Häufig fehlen Anforderungen oder sie sind unklar formuliert. Wenn beispielsweise die Anforderungen nicht genau den Kundenwunsch widerspiegeln oder die Anforderungen zu ungenau beschrieben und damit verschiedenartig interpretierbar sind, kann dies zur Folge haben, dass das erstellte System nicht den Erwartungen der Auftraggeber bzw. Nutzer entspricht.

*Symptome und Gründe  
für fehlerhafte  
Anforderungen*

Der häufigste Grund für fehlerhafte Anforderungen ist die falsche Annahme der Stakeholder, dass vieles selbstverständlich ist und nicht explizit genannt werden muss. Es entstehen Kommunikationsprobleme

zwischen den Beteiligten, die oft aus unterschiedlichem Erfahrungs- bzw. Wissenstand resultieren. Erschwerend kommt hinzu, dass besonders der Auftraggeber in vielen Fällen kurzfristige Ergebnisse in Form eines produktiven Systems erhalten möchte.

Zusammengefasst sind die typischsten Ursachen für mangelhaftes Requirements Engineering laut [IREB-Lehrplan 2020]:

- Direkt mit der Entwicklung des Systems zu beginnen, ohne eine ausreichende Verständnisgrundlage geschaffen zu haben.
- Kommunikationsprobleme zwischen den beteiligten Parteien (siehe Exkurs)
- Die Annahme, dass die Anforderungen selbstverständlich sind und keiner weiteren Erläuterung bedürfen bzw. gar nicht erst erfasst werden müssen.
- Unzureichende Ausbildung und Fähigkeiten des Requirements Engineer

#### **Exkurs: Kommunikationsprobleme**

Anforderungen müssen kommuniziert werden. In den meisten Fällen bedient man sich hierbei eines allen Kommunikationspartnern zugänglichen, regelgeleiteten Mediums – der Sprache.

Damit die Übertragung von Informationen von einem Individuum zu einem anderen funktioniert, wird ein gemeinsamer Code benötigt. Der Sender verschlüsselt seine Botschaft, die der Empfänger dann wieder entschlüsseln muss. Ein solcher gemeinsamer Code ist Menschen gegeben, die die gleiche natürliche Sprache (z.B. Deutsch) sprechen, den gleichen kulturellen Hintergrund haben und auf ähnliche Erfahrungen zurückgreifen können. Je ähnlicher der kulturelle und Bildungshintergrund, das Fachgebiet und der Arbeitsalltag sind, umso besser klappt der Austausch von Informationen. Da dies häufig unter den Stakeholdern nicht gegeben ist, ist es sinnvoll, sich zunächst auf eine gemeinsame Sprache und deren Verwendung zu einigen. Das kann z.B. der Einsatz eines Glossars (siehe Abschnitt 3.5) sein, in dem alle wichtigen Begriffe erläutert werden, oder die Eini-gung auf eine formale Beschreibungssprache, z.B. die Unified Modeling Language (UML) der OMG (siehe Abschnitt 3.4).



Ein weiterer Faktor ist die Art des Kommunikationsmediums. Bei mündlicher Kommunikation beruht der Kommunikationserfolg stark auf Redundanz (z.B. Sprache und Gestik oder Sprache und Tonfall) sowie auf Rückkopplung. Bei schriftlicher Kommunikation wird redundanzarm und ohne (oder mit wenig direkter) Rückkopplung kommuniziert.

Zusätzlich zu den Problemen der unterschiedlichen Begriffswelten und Kommunikationsmedien ist meist zu beobachten, dass Informationen gar nicht oder nicht adäquat weitergegeben werden. Dies lässt sich oftmals auf natürliche Vorgänge zurückführen, die bei der Wahrnehmung des Menschen und der Kommunikation des Wahrgenommenen immer mehr oder weniger ausgeprägt auftreten: die Fokussierung und die Vereinfachung.

Die Fokussierung geschieht primär unter zwei Gesichtspunkten. Zum einen wird man immer auf das fokussieren, was einem wichtig ist. Zum anderen kann man nur auf das fokussieren, was einem bewusst ist. Das heißt, unbewusste Informationen gehen verloren.

Kommunikation, die auf dem Ausdruck von Wissen basiert, ist notwendigerweise vereinfachend. Ein Autor setzt beim Leser ein gewisses Vorwissen voraus. Diese Vereinfachungen im Ausdruck sind es, die im Zusammenhang mit Anforderungen problematisch werden, da sie Anforderungen unterschiedlich interpretierbar machen. In Kapitel 3 wird näher auf die Darstellung von Anforderungen in natürlicher Sprache und mit Modellen eingegangen.

Die steigende Bedeutung von Systemen mit einem signifikanten Softwareanteil in industriellen Projekten sowie die Notwendigkeit, innovativere, individuellere und umfangreichere Systeme schneller, besser und mit höchster Qualität auf den Markt zu bringen, setzen ein leistungsfähiges Requirements Engineering voraus. Fehlerfreie und ausreichend vollständige Anforderungen sind die Basis für eine erfolgreiche Systementwicklung. Bereits im Requirements Engineering müssen potenzielle Risiken aufgedeckt, und, wenn es geht, behoben werden, um einen erfolgreichen Projektablauf zu ermöglichen. Fehler und Lücken in Anforderungsdokumenten müssen frühzeitig erkannt werden, um langwierige Änderungsprozesse zu vermeiden.

*Die Bedeutsamkeit von gutem Requirements Engineering*

### Kernfakten 1–2: Requirements Engineering – Warum



[www.cpre-buch.de/pk1w2](http://www.cpre-buch.de/pk1w2)



### 1.3 Requirements Engineering – Wo?

Neben der Unterscheidung in funktionale Anforderungen, Qualitätsanforderungen und Randbedingungen wird eine Reihe anderer Klassifizierungen von Anforderungen in der Praxis verwendet. Diese können unternehmensspezifisch, projektspezifisch oder aufgrund anderer Vorgaben entstehen. Dies gilt beispielsweise für die in diversen Standards definierten Anforderungsklassen (z.B. CMMI [SEI 2006], SPICE [ISO/IEC 15504-5]) oder in Bezug auf die Klassifikation über Attributwerte von Anforderungen, etwa für den Detaillierungsgrad, die Priorität oder die rechtliche Verbindlichkeit von Anforderungen.

*Klassifikation von Anforderungen*

Häufig findet sich die Unterscheidung zwischen Systemanforderungen und Nutzeranforderungen. Daneben sind aber auch weitere Unterscheidungen gängig. Eine häufig verwendete Klassifikation von Anforderungen ist die Unterscheidung zwischen Systemanforderungen, Stakeholder-Anforderungen, Nutzeranforderungen, Domänenanforderungen und Geschäftsanforderungen [IREB-Lehrplan 2020].

- *Systemanforderungen* beschreiben, was das System leisten soll. Somit können Systemanforderungen als die »klassischen« Anforderungen aufgefasst werden, die dann in funktionale Anforderungen an das System, Anforderungen an die Qualität des Systems und Randbedingungen für Systemausführung und -entwicklung unterschieden werden.
- *Stakeholder-Anforderungen* beschreiben aus der Sicht eines Stakeholders, was dieser mit dem System erreichen will. Stakeholder-Anforderungen eignen sich, um Anforderungen aus unterschiedlichen Perspektiven zu erfassen und zu dokumentieren. Im Zuge der Definition von Systemanforderungen müssen basierend auf diesen Stakeholder-Anforderungen u.a. die Anforderungen der verschiedenen Stakeholder konsolidiert werden, Konflikte identifiziert und aufgelöst werden (siehe Abschnitt 4.3).
- *Benutzeranforderungen* beschreiben aus der Nutzerperspektive, was die Benutzer mit dem System erreichen wollen bzw. wie dieses genutzt werden soll. Nutzeranforderungen können als eine Unterkategorie der Stakeholder-Anforderungen aufgefasst werden, da die Nutzer eine spezifische Gruppe von Stakeholdern sind.
- *Domänenanforderungen* beschreiben Anforderungen (häufig Randbedingungen), die von dem jeweiligen Umfeld, in dem das System eingesetzt werden soll, vorgegeben werden.

- *Geschäftsanforderungen* beschreiben Anforderungen, die eng mit dem gewünschten Business Value verzahnt sind. Geschäftsanforderungen werden im Allgemeinen von der Organisation vorgegeben. Außerdem umfassen Geschäftsanforderungen wirtschaftliche Zielgrößen und Budgetbeschränkungen.

### Exkurs: Ziele und Szenarien

Neben der Unterscheidung von Anforderungen entsprechend verschiedener Bereiche wird häufig zwischen weiteren Anforderungsarten unterschieden. Eine hilfreiche Unterscheidung ist hierbei die Unterteilung in:

- Ziele
- Szenarien
- Lösungsorientierte Anforderungen

Dies bringt die unterschiedliche Granularität von Anforderungen zum Ausdruck. Mit Zielen und Szenarien werden zwei Anforderungsarten in den Fokus gerückt, die im Gegensatz zu sehr detaillierten technischen Anforderungen auf das Verständnis des Problemraums hinwirken.

Unter einem Ziel versteht man die intentionale Beschreibung eines von Stakeholdern (z.B. Personen oder Organisationen) gewünschten charakteristischen Merkmals des zu entwickelnden Systems bzw. des zugehörigen Entwicklungsprojekts. Ziele werden somit sehr nah an der ursprünglichen Intention der Stakeholder definiert. Damit eignen sich Ziele sehr gut, um in frühen Phasen erste Anforderungen zu erheben und diese bereits zu validieren und zwischen den verschiedenen Stakeholdern abzustimmen.

Szenarien werden genutzt, um beispielhafte Beschreibungen für Ziele oder Anforderungen zu definieren. Diese tauchen direkt in die Lebensrealität der Stakeholder ein und sind damit sehr gut geeignet, Detailwissen der Stakeholder zu erfragen und mit dem Verständnis des Requirements Engineer und den initialen Anforderungen abzugleichen. Eine bekannte Form von Szenarien sind User Stories. User Stories beschreiben ein aus Nutzerperspektive dargestelltes Bedürfnis und sind Ausdruck des Wertes/Nutzens, wenn dieses erfüllt ist.

### Kernfakten 1–3: Requirements Engineering – Wo



[www.cpre-buch.de/pk1w3](http://www.cpre-buch.de/pk1w3)



### Hinweis: Requirements-Engineering-Prozess vs. Systementwicklungsprozess

Im weiteren Verlauf werden wir uns mit dem Requirements-Engineering-Prozess und den im Rahmen des Requirements Engineering anfallenden Aufgaben beschäftigen. Es ist jedoch zu berücksichtigen, dass Requirements Engineering kein Selbstzweck ist, sondern dass jeder Requirements-Engineering-Prozess immer auch in einen System- oder Softwareentwicklungsprozess eingebettet ist. Wie wir noch sehen werden, verfügen wir im Requirements Engineering über unterschiedliche Bausteine, die wir kombinieren können, um den jeweiligen System- oder Softwareentwicklungsprozess optimal unterstützen zu können. Eine immer wiederkehrende Unterscheidung, die Einfluss auf die Wahl des richtigen Requirements-Engineering-Vorgehens hat, ist die Unterscheidung zwischen traditionellen (eher wasserfallartigen) und agilen Vorgehensmodellen.

#### ■ *Traditionelle Entwicklung*

In schwergewichtigen Vorgehensmodellen (z.B. Wasserfallmodell [Royce 1987], V-Modell [V-Modell 2004]) wird versucht, alle Anforderungen in einer Projektphase vollständig zu erheben, bevor die ersten Entwurfs- oder Realisierungsentscheidungen getroffen werden. Das Ziel dieser Modelle ist es, bereits im Vorfeld der Umsetzung alle Anforderungen an das System zu ermitteln. Dies führt dazu, dass Requirements Engineering bei diesen Vorgehensmodellen als abgeschlossene, zeitlich befristete erste Phase der Systementwicklung durchgeführt wird.

#### ■ *Agile Entwicklung*

Leichtgewichtige Vorgehensmodelle (z.B. eXtreme Programming [Beck 1999]) ermitteln die benötigten Anforderungen dagegen erst, wenn sie implementiert werden sollen, da »Hellsehen« bezüglich zukünftig gewünschter Funktionalität schwierig ist und Anforderungen sich auch im Laufe eines Projekts ändern. Hier wird Requirements Engineering als kontinuierlicher, phasenübergreifender Prozess in die Systementwicklung integriert.

## 1.4 Requirements Engineering – Wie?

Der Lehrplan [IREB-Lehrplan 2020] unterscheidet zwischen vier Hauptaufgaben eines Requirements Engineer:

*Vier Haupttätigkeiten  
im Requirements  
Engineering*

### ■ *Ermitteln*

Beim Ermitteln der Anforderungen werden verschiedene Techniken genutzt, um die Anforderungen der Stakeholder und anderer Quellen zu gewinnen (siehe Abschnitt 4.2). Insbesondere werden bei der Ermittlung von Anforderungen auch Techniken angewandt, um Konflikte zwischen verschiedenen Stakeholdern aufzudecken und diese aufzulösen.

### ■ *Dokumentieren*

Durch die Dokumentation werden erarbeitete Anforderungen adäquat beschrieben. Hierfür können unterschiedliche Techniken eingesetzt werden, um Anforderungen in natürlicher Sprache oder in Modellen zu dokumentieren (siehe Kapitel 3).

### ■ *Validieren*

Dokumentierte Anforderungen müssen frühzeitig geprüft werden, um zu gewährleisten, dass sie der geforderten Qualität genügen (siehe Abschnitt 4.4).

### ■ *Verwalten*

Die Anforderungsverwaltung (Requirements Management) geschieht flankierend zu allen anderen Aktivitäten und umfasst alle Maßnahmen, die notwendig sind, um Anforderungen zu strukturieren, für unterschiedliche Rollen aufzubereiten sowie konsistent zu ändern (siehe Kapitel 6).



Diese Haupttätigkeiten können durch Prozesse, wie sie z.B. im Standard [ISO/IEC/IEEE 29148:2018] empfohlen werden, in eine Reihenfolge gebracht werden. Grundsätzlich gibt es allerdings nicht den einen Requirements-Engineering-Prozess. Vielmehr muss der Requirements Engineer abhängig vom jeweiligen Entwicklungsprojekt und vor allem den beteiligten Stakeholdern den geeigneten Requirements-Engineering-Prozess zusammenstellen. Dabei kann er sich der verschiedenen Tätigkeiten und Techniken des Requirements Engineering bedienen, die im weiteren Verlauf des Buches näher vorgestellt werden. Die Haupttätigkeiten können für Anforderungen auf unterschiedlichen Ebenen durchgeführt werden, wie z.B. System-, Software- oder Stakeholder-Anforderungen.

## 1.5 Die Rolle und Aufgaben eines Requirements Engineer

Der Requirements Engineer als Projekttrolle steht oftmals im Mittelpunkt des Geschehens. Auch wenn in der Praxis die Rolle des Requirements Engineer häufig unterschiedlichen anderen Tätigkeitsprofilen zugeordnet wird: Business-Analysten, Anwendungsspezialisten, Product Ownern, Systemingenieuren, Entwicklern oder eben klassischen Requirements Engineers. Er pflegt in der Regel als Einziger direkten Kontakt zu allen Stakeholdern und hat die Chance und Verantwortung, sich ausreichend in das Fachgebiet der Stakeholder einzuarbeiten sowie die Sprache in den verschiedenen Fachgebieten zu erlernen und zu verstehen. Er ist derjenige, der die Bedürfnisse hinter den Aussagen der Stakeholder erkennen und so aufbereiten muss, dass Architekten und Entwickler sie verstehen und umsetzen können. Als solches fungiert der Requirements Engineer auch als ein Vermittler zwischen den verschiedenen Stakeholdern. Hierzu sollte der Requirements Engineer sowohl das Fachgebiet und dessen Sprache ausreichend kennen als auch über genug IT-Know-how verfügen. Der Requirements Engineer nimmt somit auf vielfältige Weise eine zentrale Rolle im Projekt ein.

*Fähigkeiten des  
Requirements Engineer*

Typische Fähigkeiten und Aufgaben des Requirements Engineer umfassen laut [IREB-Lehrplan 2020]:

- Anforderungen ermitteln, dokumentieren, validieren und verwalten.
- Über fundierte Kenntnisse im Requirements Engineering verfügen und diese aktiv einbringen.
- Die Lücke zwischen dem Problem und möglichen Lösungen überbrücken.

Um allen Aufgaben gerecht werden zu können, benötigt der Requirements Engineer weit mehr als theoretisches Methodenwissen. Viele der benötigten Fähigkeiten setzen entsprechende praktische Erfahrungen voraus.

### Exkurs: Persönlichkeitsprofil eines Requirements Engineer

Neben Methodenwissen und technischen Kenntnissen sind für einen guten Requirements Engineer zahlreiche Softskills wichtig. Hierzu zählen:

#### ■ *Analytisches Denken*

Der Requirements Engineer muss fähig sein, sich in unbekannte oder wenig bekannte Fachgebiete und Sachverhalte schnell einzuarbeiten und dabei komplizierte Probleme und Zusammenhänge verstehen und analysieren können. Da Stakeholder oft in konkreten Beispielen und (suboptimalen) Lösungen über das eigentliche Problem und die zugehörigen Anforderungen sprechen, muss der Requirements Engineer in der Lage sein, konkrete Aussagen der Stakeholder zu abstrahieren.

#### ■ *Empathie*

Der Requirements Engineer hat die schwierige Aufgabe zu erkennen, was ein Stakeholder tatsächlich benötigt. Hierfür ist ein ausgeprägtes Einfühlungsvermögen eine der zentralen Voraussetzungen. Zudem muss er problematische gruppenspezifische Effekte unter den Stakeholdern erkennen und geeignet darauf reagieren können.

#### ■ *Kommunikationsfähigkeit*

Um die Anforderungen der Stakeholder zu erheben, richtig zu interpretieren und zu kommunizieren, muss der Requirements Engineer über hohe kommunikative Fähigkeiten verfügen. Er muss zuhören können, zur rechten Zeit die richtigen Fragen stellen, bemerken, wenn Aussagen nicht den gewünschten Informationsgehalt haben oder Unsicherheiten beinhalten, und dann rechtzeitig erforderliche Rückfragen stellen.

#### ■ *Konfliktlösungsfähigkeit*

Durch unterschiedliche Meinungen der Stakeholder kommt es im Requirements Engineering häufig zu Konflikten. Der Requirements Engineer muss Konflikte erkennen, zwischen den Parteien vermitteln und schließlich durch den Einsatz geeigneter Techniken den Konflikt durch Konsensfindung oder per Entscheidung auflösen.

#### ■ *Moderationsfähigkeit*

Der Requirements Engineer muss zwischen unterschiedlichen Meinungen vermitteln und Diskussionen leiten können. Dies gilt sowohl für Einzelgesprächen als auch in Gruppengesprächen oder in Workshops.

### ■ *Selbstbewusstsein*

Da der Requirements Engineer häufig im Mittelpunkt steht und dabei gelegentlich auch Kritik ausgesetzt ist, benötigt er ein selbstbewusstes Auftreten und die Fähigkeit, sich auch durch hartnäckige Ablehnungen nicht aus dem Konzept bringen zu lassen. Er sollte Kritik niemals persönlich nehmen, denn diese gilt oftmals seiner Rolle und nicht ihm als Mensch.

### ■ *Überzeugungsfähigkeit*

Der Requirements Engineer ist u. a. eine Art Anwalt für die Anforderungen seiner Stakeholder. Er muss fähig sein, diese nach außen und in Besprechungen und Präsentationen überzeugend zu vertreten.

## 1.6 Was über Requirements Engineering zu lernen ist

Requirements Engineering ist ein intensiv erforschtes und weites Feld, das viele Aspekte umfasst. Im Zuge der Zertifizierung zum »*Certified Professional for Requirements Engineering*« des IREB sollen Sie einen Einblick in viele dieser Teilgebiete gewinnen und grundlegende Kenntnisse für das Berufsfeld Requirements Engineer erwerben. Wir haben dieses Buch eng am Lehrplan organisiert und nehmen uns die Freiheit, Ihnen immer wieder auch darüber hinausgehende Aspekte näherzubringen, die wir als wichtig für die Tätigkeit eines Requirements Engineer erachten. Diese Inhalte markieren wir als Exkurs. Im Detail werden wir uns mit folgenden Punkten beschäftigen:

- Neun grundlegende Prinzipien des Requirements Engineering werden wir in Kapitel 2 betrachten.
- Die Dokumentation von Anforderungen vermitteln wir in Kapitel 3.
- Die Erarbeitung (also die Gewinnung, Abstimmung und Validierung) von Anforderungen beschäftigt uns in Kapitel 4.
- Requirements-Engineering-Prozesse und deren Gestaltung ist das Thema, mit dem wir uns in Kapitel 5 befassen.
- Das Management von Anforderungen liegt im Fokus von Kapitel 6.
- Die Werkzeugunterstützung für alle oben genannten Themenbereiche ist der abschließende Lehrinhalt in Kapitel 7.