

1 Einleitung und Grundlagen

Die Bedeutung des Requirements Engineering (RE) für die erfolgreiche, den Kunden zufriedenstellende Entwicklung von Systemen ist mittlerweile kaum mehr zu übersehen. In der Praxis ist es üblich, einen entsprechenden Aufwand für das Requirements Engineering einzuplanen. Immer häufiger findet man zudem die Erkenntnis, dass der Requirements Engineer eine eigenständige Rolle mit anspruchsvollen Tätigkeiten ist.

1.1 Einleitung

Glaubt man den Zahlen im Chaos Report 2006 der Standish Group, so hat sich in den zwölf Jahren zwischen 1994 und 2006 bei der erfolgreichen Abwicklung von Softwareprojekten einiges zum Besseren gewendet. Sind im Jahre 1994 noch gut 30% der untersuchten Softwareprojekte gescheitert, so waren es 2006 nur noch knapp 20%. Die Anzahl der Projekte, die mit starken Zeit- oder Budgetüberziehungen und/oder nicht zur Zufriedenheit der Kunden abgeschlossen werden konnten, verringerte sich von 53% auf 46% [Chaos 2006]. Jim Johnson, Vorsitzender der Standish Group, nennt als einen von drei Gründen für die positive Entwicklung der Zahlen seit 1994 die Tatsache, dass Anforderungen besser kommuniziert würden als noch vor zehn Jahren. Interessant sind diese Zahlen, da der Umgang mit Anforderungen eines Systems eine signifikante Ursache für Projektfehlschläge bzw. für Zeit- und Budgetüberschreitungen darstellt.

Wozu Requirements Engineering?

1.1.1 Zahlen und Fakten im Projektalltag

Studien belegen, dass etwa 60% der Fehler in Systementwicklungsprojekten bereits im Requirements Engineering entstehen [Boehm 1981]. Irrtümer aus dem Requirements Engineering werden jedoch oft erst in späteren Projektphasen oder im Betrieb des Systems entdeckt, da feh-

Requirements Engineering als Fehlerquelle

lerhafte oder unvollständige Anforderungen für Entwickler so interpretiert werden können, dass sie subjektiv schlüssig sind oder aus der subjektiven Perspektive der Entwickler (unbewusst) vervollständigt werden. Fehlende Anforderungen werden im Entwurf und in der Realisierung häufig nicht entdeckt, da man sich hier auf die qualitativ hochwertige Arbeit des Requirements Engineer verlässt. Es wird umgesetzt, was man aus dem Anforderungsdokument entnehmen kann oder glaubt, entnehmen zu können. Missverständliche, unvollständige oder falsche Anforderungen führen somit unausweichlich zu einem System, das wichtige Eigenschaften nicht besitzt oder Eigenschaften aufweist, die nicht gefordert wurden.

*Kosten von Fehlern im
Requirements Engineering*

Je später ein Fehler in den Anforderungen im Verlauf des Entwicklungsprojekts behoben wird, umso höher sind die damit verbundenen Kosten. So wird beispielsweise für die Beseitigung eines Anforderungsfehlers, der erst beim Programmieren entdeckt wird, ein um ca. den Faktor 20 höherer Aufwand notwendig, als wenn derselbe Fehler während des Requirements Engineering behoben worden wäre – für die Fehlerbeseitigung in der Abnahmephase des Systems geht man von dem Faktor 100 aus [Boehm 1981].

*Symptome und Gründe
für mangelhaftes
Requirements Engineering*

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, hat dies häufig zur Folge, dass das erstellte System nicht den Erwartungen der Auftraggeber bzw. Nutzer entspricht.

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.

*Die Bedeutsamkeit von
gutem Requirements
Engineering*

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 vollständige Anforderungen sind die Basis für eine erfolgreiche Systementwicklung. Bereits im Requirements Engineering müssen potenzielle Risiken aufgedeckt und soweit möglich behoben werden, um einen erfolgreichen Projektablauf zu ermöglichen. Fehler und

Lücken in Anforderungsdokumenten müssen frühzeitig erkannt werden, um langwierige Änderungsprozesse zu vermeiden.

1.1.2 Requirements Engineering – was ist das?

Um ein Entwicklungsprojekt zum Erfolg führen zu können, muss zunächst bekannt sein, was die Anforderungen an das System sind, und diese müssen geeignet dokumentiert sein.

Definition 1-1: Anforderung

Eine Anforderung ist:

- (1) Eine Bedingung oder Fähigkeit, die von einem Benutzer (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.
- (2) Eine Bedingung oder Fähigkeit, die ein System oder Teilsystem erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.
- (3) Eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft gemäß (1) oder (2).

Übersetzt aus [IEEE Std 610.12-1990]

Der Stakeholder (Projektbetroffener) ist einer der zentralen Begriffe im Requirements Engineering. Stakeholder dienen u.a. als wichtigste Quellen für 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. der Nutzer oder der Administrator), natürliche Personen, die Interesse an dem System haben, es aber nicht nutzen werden oder sollen (z.B. das Management oder ein Hacker, vor dem man das System schützen muss), aber auch juristische Personen, Institutionen usw., da diese letztlich durch natürliche Personen vertreten werden, die die Anforderungen des betrachteten Systems beeinflussen bzw. definieren können.

Stakeholder

Definition 1-2: Stakeholder

Ein Stakeholder eines Systems ist eine Person oder Organisation, die (direkt oder indirekt) Einfluss auf die Anforderungen des betrachteten Systems hat.

Ziel des Requirements
Engineering

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 [Pohl 1996].

Definition 1-3: Requirements Engineering

Das Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit den folgenden Zielen:

- (1) Die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen, die Anforderungen konform zu vorgegebenen Standards zu dokumentieren und die Anforderungen systematisch zu managen.
- (2) Die Wünsche und Bedürfnisse der Stakeholder zu verstehen, zu dokumentieren sowie die Anforderungen zu spezifizieren und zu managen, um das Risiko zu minimieren, dass das System nicht den Wünschen und Bedürfnissen der Stakeholder entspricht.

Vier Haupttätigkeiten
im Requirements
Engineering

Die vier damit einhergehenden Haupttätigkeiten sind:

■ *Ermitteln:*

Beim Ermitteln der Anforderungen werden verschiedene Techniken genutzt, um die Anforderungen der Stakeholder und anderer Quellen zu gewinnen, zu detaillieren und zu verfeinern (siehe Kapitel 3).

■ *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 4, 5, 6).

■ *Prüfen und abstimmen:*

Dokumentierte Anforderungen müssen frühzeitig geprüft und abgestimmt werden, um zu gewährleisten, dass sie allen geforderten Qualitätskriterien genügen (siehe Kapitel 7).

■ *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 und umzusetzen (siehe Kapitel 8).

Diese Haupttätigkeiten können durch Prozesse, wie sie z.B. im Standard ISO/IEC/IEEE 29148:2011 empfohlen werden, in eine Reihenfolge gebracht werden. Die Haupttätigkeiten können für Anforderungen auf unterschiedlichen Ebenen durchgeführt werden, wie z.B. System-, Software- oder Stakeholderanforderungen.

Unterschiedliche Projekttrandbedingungen beeinflussen das Requirements Engineering. So haben z.B. die beteiligten Menschen, fachliche Faktoren (z.B. die Branche) oder organisatorische Randbedingungen (z.B. die räumliche Verteilung oder die zeitliche Verfügbarkeit der Projektbeteiligten) eine große Auswirkung auf die Auswahl geeigneter Techniken.

Randbedingungen

1.1.3 Einbettung des Requirements Engineering in Vorgehensmodelle

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.

Requirements Engineering als abgeschlossene Phase

Leichtgewichtige Vorgehensmodelle (z.B. eXtreme Programming [Beck 1999]) ermitteln die benötigten Anforderungen dagegen erst, wenn sie implementiert werden sollen, da »Hellsehen« bzgl. 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.

Requirements Engineering als begleitender Prozess

1.2 Kommunikationstheoretische Grundlagen

*Sprache als Medium
zur Kommunikation
von Anforderungen*

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 kultureller, familiärer und Bildungshintergrund, Fachgebiet und Arbeitsalltag sind, umso besser klappt der Austausch von Informationen. Da solch optimale Bedingungen häufig unter den Stakeholdern nicht gegeben sind, ist es sinnvoll, sich zunächst auf eine gemeinsame Sprache und deren Verwendung zu einigen. Das kann z.B. der Einsatz eines Glossars (siehe Kapitel 4) sein, in dem alle wichtigen Begriffe erläutert werden, oder die Einigung auf eine formale Beschreibungssprache, z.B. die Unified Modeling Language, UML, der OMG¹ (siehe Kapitel 6).

*Art des
Kommunikations-
mediums*

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 technischer schriftlicher Kommunikation wird redundanzarm und ohne (oder mit wenig) Rückkopplung kommuniziert.

*Sprachliche
Bequemlichkeit*

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*.

Impliziertes Vorwissen

Kommunikation, der sprachliche Ausdruck des Wissens, ist notwendigerweise vereinfachend. Ein Autor setzt beim Leser ein gewisses Vorwissen voraus. Diese Vereinfachungen im sprachlichen Ausdruck sind es, die im Zusammenhang mit Anforderungen problematisch werden, da sie Anforderungen unterschiedlich interpretierbar machen. In Kapitel 5 wird näher auf die Darstellung von Anforderungen in natürlicher Sprache eingegangen.

1. OMG (Object Management Group) – www.omg.org.

1.3 Eigenschaften eines Requirements Engineer

Der Requirements Engineer als Projektrolle steht häufig im Mittelpunkt des Geschehens. 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 fachfremde Architekten und Entwickler sie verstehen und umsetzen können. Man kann sich den Requirements Engineer also als eine Art Dolmetscher vorstellen, der sowohl das Fachgebiet und dessen Sprache ausreichend kennt als auch über genug IT-Know-how verfügt, um sich der Probleme der Entwickler bewusst zu sein und mit ihnen gleichberechtigt kommunizieren zu können. Der Requirements Engineer nimmt daher eine zentrale Rolle im Projekt ein.

Zentrale Rolle

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

*Sieben notwendige
Fähigkeiten eines
Requirements Engineer*

■ *Analytisches Denken:*

Der Requirements Engineer muss fähig sein, sich in ihm 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.

■ *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, und 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 auflösen.

■ *Moderationsfähigkeit:*

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

■ *Selbstbewusstsein:*

Da der Requirements Engineer häufig im Mittelpunkt steht und dabei gelegentlich auch der 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.

■ *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 zu vertreten. Zudem muss er die unterschiedlichen Meinungen der Stakeholder konsolidieren und im Falle eines Dissenses eine Entscheidung herbeiführen oder Konsens unter den Stakeholdern herstellen.

1.4 Arten von Anforderungen

Im Allgemeinen unterscheidet man zwischen drei Arten von Anforderungen:

- *Funktionale Anforderungen* legen die Funktionalität fest, die das geplante System zur Verfügung stellen soll. Sie werden typischerweise in Funktions-, Verhaltens- und Strukturanforderungen unterteilt (siehe Kapitel 4).

Definition 1–4: *Funktionale Anforderung*

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

- *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 Performanz, die Verfügbarkeit, die Zuverlässigkeit, die Skalierbarkeit oder die Portabilität des betrachteten Systems. Anforderungen dieses Typs werden häufig auch der Klasse »nicht funktionaler Anforderungen« zugeordnet.

Definition 1-5: *Qualitätsanforderung*

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

- *Randbedingungen* (auch: Rahmenbedingung) 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 2010 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*

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

Neben der Unterscheidung in funktionale Anforderungen, Qualitätsanforderungen und Randbedingungen wird eine Reihe anderer Klassifizierungen von Anforderungen in der Praxis verwendet. 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 (siehe Kapitel 4 und 8).

1.5 Bedeutung und Kategorisierung von Qualitätsanforderungen

In der täglichen Praxis werden die Qualitätsanforderungen eines Systems häufig nur unzureichend dokumentiert und zwischen Stakeholdern abgestimmt. Dies kann in erheblichem Maße den Projekterfolg oder die spätere Akzeptanz des entwickelten Systems gefährden. 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 [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 sind (vgl. [ISO/IEC 25010:2011]):

- Anforderungen, die die *Performanz* des Systems definieren, insbesondere das Antwortzeitverhalten und der Ressourcenverbrauch.
- Anforderungen, die die *Sicherheit* des Systems definieren, insbesondere die Nachweisbarkeit, Authentizität, Vertraulichkeit und Integrität.
- Anforderungen, die die *Zuverlässigkeit* der Funktionalität 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.

Qualitätsanforderungen werden gegenwärtig meist in natürlicher Sprache formuliert. Alternativ zu der bisher üblichen Dokumentation von Qualitätsanforderungen in natürlicher Sprache wurde eine Reihe

von Ansätzen vorgestellt, die darauf abzielen, Qualitätsanforderungen in Form von Modellen bzw. als Erweiterungen gängiger Modellierungsansätze zu dokumentieren.

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 *Performance* des Systems festlegen, dass das System 95% aller Anfragen in weniger als 1,5 Sekunden abarbeitet und die Abarbeitung einer Anfrage auf keinen Fall mehr als 4 Sekunden in Anspruch nehmen darf. Qualitätsanforderungen können hierbei durch zusätzliche funktionale Anforderungen konkretisiert werden. Beispielsweise kann eine Qualitätsanforderung bezüglich der Sicherheit des Systems durch die Forderung nach der Verschlüsselung der Ausgabedaten präzisiert werden. Die geforderte Verschlüsselung der Ausgabedaten stellt dabei eine funktionale Anforderung dar, die die geforderten Sicherheitseigenschaften des Systems konkretisiert.

Qualitätsanforderungen stehen daher häufig mit verschiedenen funktionalen Anforderungen in Beziehung. Aus diesem Grunde sollten Qualitätsanforderungen stets getrennt von funktionalen Anforderungen definiert werden. Zudem sollte der Bezug von Qualitätsanforderungen und funktionalen Anforderungen explizit dokumentiert werden.

1.6 Zusammenfassung

Requirements Engineering ist kaum noch wegzudenken, wenn es darum geht, den Kunden zufriedenstellende Systeme zu entwickeln und dabei Budget- und Zeitpläne einzuhalten. Ziel des Requirements Engineering ist es, möglichst vollständige Kundenanforderungen in guter Qualität zu dokumentieren und dabei Fehler möglichst frühzeitig zu erkennen und zu beheben. Grundlage für erfolgreiches Requirements Engineering ist sowohl die Einbeziehung der richtigen Stakeholder als auch die Einbettung der vier Haupttätigkeiten des Requirements Engineering (*Ermitteln*, *Dokumentieren*, *Prüfen* und *Abstimmen* sowie *Verwalten* von Anforderungen) in den Systementwicklungsprozess. Im Mittelpunkt des Geschehens steht der Requirements Engineer, der primäre Anlaufstelle im Requirements Engineering ist und neben Fachwissen und Methodenwissen auch über eine Vielzahl an Softskills verfügen muss.