

3 Einführung Requirements Engineering

Dieses Kapitel gibt eine kompakte Einführung in die Disziplin des Requirements Engineering (kurz RE) wieder. Zunächst erhalten Sie einen Überblick darüber, wie sich RE in das Software Engineering als Ganzes einsortiert. Anschließend wird eine geschichtliche Perspektive eingenommen und es wird die historische Entwicklung von RE betrachtet. Dann folgt die Bedeutung von RE für den Erfolg von Softwareprojekten, dabei wird auf essenzielle Aktivitäten und Arbeitsergebnisse des RE eingegangen.

3.1 Was ist Requirements Engineering?

Die Definition von Anforderungen an eine Software ist eine zentrale Aufgabe des Software Engineering und gehört neben dem Design der Software-Architektur, der Implementierung und der Qualitätssicherung zu den Kerndisziplinen der Entwicklung von Software (vgl. [SWE14]). Einen vollständigen Überblick über alle weitere Disziplinen und Wissensgebiete des Software Engineering findet sich ebenfalls in [SWE14].

3.1.1 Stark vereinfachte Definition von RE

Die Verantwortung für die Definition und Dokumentation von Anforderungen an ein zu entwickelndes System übernimmt im Software Engineering das RE.

Abbildung 1 stellt eine stark vereinfachte visuelle Definition des RE dar: Das RE erstellt im Rahmen eines Interaktionsprozesses mit Teilen der Welt (bspw. Unternehmen, Nutzer, Kunden, Gesetzgeber) eine Spezifikation oder allgemeiner ein Konzept einer Software.

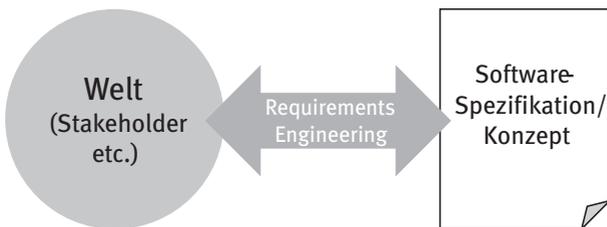


Abbildung 1: Stark vereinfachte Definition von RE

Als Ergebnis des RE entsteht die Spezifikation bzw. das Konzept einer Software. Diese Spezifikation muss nicht zwangsläufig in Form eines Dokumentes aus Papier erstellt werden. Spezifikationen werden auch häufig in Software ab-

gebildet und bspw. in Datenbanken gespeichert. Die Form der Dokumentation sollte maßgeblich durch die Projektgegebenheiten definiert werden. Hinweise zur Wahl der Form werden in Kapitel 9 gegeben.

Die Frage nach dem Umfang und dem Detailgrad einer Spezifikation ist ein häufiger Streitpunkt im RE. Eine einheitliche Antwort auf diese Frage gibt es nicht, aber eine wichtige Richtlinie. Die Spezifikation an sich ist kein Selbstzweck, sie dient als Grundlage für die technische Umsetzung bzw. Realisierung der beschriebenen Software. Damit reduziert eine Spezifikation (und ihre Diskussion mit den Stakeholdern) das Risiko, dass eine unpassende Software erstellt wird, die nicht den Vorstellungen der Stakeholder entspricht.

Anders formuliert, die Spezifikation ist im Wesentlichen so etwas wie die Bau- und Konstruktionszeichnungen aus dem Bauwesen und dem Maschinenbau. Auch hier sollen die essenziellen Entscheidungen zur Gestaltung des Gebäudes oder der Maschine getroffen werden, bevor mit der eigentlichen Produktion des Gebäudes oder der Maschine begonnen wird. Wie viel Aufwand in eine Spezifikation gesteckt werden sollte, hängt damit maßgeblich von der Risikoeinschätzung der Beteiligten darüber ab, wie viel des geplanten Systems vor der eigentlichen Konstruktion abgestimmt und definiert werden muss (vgl. [Gli08]).

Wann und wie die technische Umsetzung der Spezifikation erfolgt, ist nicht mehr Entscheidung oder Bestandteil des RE, sondern Bestandteil des Vorgehensmodells, das zur Entwicklung der Software gewählt wurde. Prinzipiell können zwei extreme Arten von Modellen zur Gestaltung des Entwicklungsprozesses unterschieden werden:

- *Phasenorientierte Modelle*: Die Entwicklung des Konzeptes ist eine eigenständige Phase des Vorgehensmodells. Die Umsetzung kann erst nach vollständiger Fertigstellung des Konzeptes starten. Ein Beispiel für diese Art Vorgehensmodelle ist das V-Modell [VMo00].
- *Iterative Modelle*: Die Entwicklung des Konzeptes geschieht parallel zur Umsetzung. Jeweils fertige Teile des Konzeptes werden in die Umsetzung gegeben. Ein Beispiel für ein iteratives Modell ist SCRUM [Dee12].

Das Konzept oder die Spezifikation für eine Software wird nicht aus der Tiefe des Raums geschöpft, sondern wird gemeinsam mit Menschen entwickelt, die für die Entwicklung der Software verantwortlich sind bzw. die Entwicklung der Software wünschen. Diese Menschen werden im Verlauf dieses Buches weiter konkretisiert, für den Moment soll es reichen, dass RE das Konzept in Zusammenarbeit mit der Welt als Platzhalter für alle Beteiligten erstellt.

Schlussendlich ist das RE ein interaktiver Prozess (Doppelpfeil in Abbildung 1) zur Erstellung des Konzeptes. Die Interaktivität ist ein wesentliches Merkmal

des Prozesses und soll an dieser Stelle besonders betont werden, da RE häufig dahingehend missverstanden wird, dass die Information über das Konzept nur von der Welt in das Konzept fließt.

In der Praxis kann man häufig das folgende, stark vereinfachte Fehlverhalten beobachten: Im Rahmen des RE spricht eine Person mit Nutzern und anderen relevanten Personen (bspw. in Interviews oder Workshops). Die Erkenntnisse aus diesen Gesprächen werden ohne weitere Einbeziehung der betroffenen Personen zu einer Spezifikation verarbeitet und in die Umsetzung gegeben. Das Ergebnis ist in den meisten Fällen ein System, das von Nutzern nicht akzeptiert wird. Durch eine aktive Einbeziehung aller relevanten Personen in einen systematischen Prozess wird das Risiko der mangelhaften Akzeptanz einer Software signifikant reduziert.

Diese stark vereinfachte Definition wird nun im folgenden Abschnitt durch eine umfassendere Definition konkretisiert.

3.1.2 Umfassende Definition von RE

Das International Requirements Engineering Board (IREB) definiert RE wie folgt [Gli14]:

DEFINITION

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 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.

Diese Definition macht deutlich, dass RE ein breites Spektrum an Aufgaben und Verantwortlichkeiten umfasst. Da Definitionen einen Sachverhalt stets sehr komprimiert darstellen, werden die einzelnen Facetten der Definition im Folgenden ausführlicher erläutert. Weiterhin wird aufgezeigt, in welchen Buchkapiteln die jeweiligen Facetten der Definition genauer betrachtet werden. Die jeweils erläuterten Definitionsteile werden so *hervorgehoben*.

Das Selbstverständnis und die zentrale Aufgabe des RE wird am Ende der Definition deutlich gemacht. RE setzt als Disziplin alles daran, *das Risiko zu minimieren, dass das System nicht den Wünschen und Bedürfnissen der Stakeholder entspricht*.

Daraus kann man im Umkehrschluss ableiten, dass dem RE eine zentrale Verantwortung für die spätere Gestalt der zu entwickelnden Software zugesprochen wird. Denn im RE werden die zentralen Weichen dafür gestellt, wie Software auszusehen hat, nicht nur im Sinne der Optik (bspw. der Nutzeroberfläche), sondern insbesondere auch im Sinne der Funktionalität und des Verhaltens. Diese Verantwortung wird in der industriellen Praxis leider oft nicht gelebt, insbesondere dann nicht, wenn ein Projekt nicht das gewünschte Ergebnis liefert. Häufig werden die Projektleiter oder Softwareentwickler für schlechte Software verantwortlich gemacht. Vielmehr sollten hier die für das RE zuständigen Personen zur Verantwortung gezogen werden, natürlich nur dann, wenn RE explizit betrieben wurde.

Die Definition beginnt damit, dass RE ein *systematischer und disziplinierter Ansatz* ist. Hierdurch soll deutlich gemacht werden, dass Tätigkeiten in einem Projekt nur dann als RE bezeichnet werden sollten, wenn sie einer gewissen Systematik folgen und mit angemessener Disziplin durchgeführt werden. Die Systematik und Disziplin spiegeln sich insbesondere in der Denkweise (siehe Kapitel 4), in den erstellten Ergebnissen (siehe Kapitel 6) und der Durchführung der Aktivitäten (siehe Teil IV) wider.

Das Bestreben nach Systematik und Disziplin im RE darf allerdings nicht dahingehend missverstanden werden, dass Regeln und Vorgaben über den eigentlichen Projektzweck gestellt werden: Software, die den Wünschen und Bedürfnissen der Stakeholder entspricht. Ein solches Missverständnis über das notwendige Maß an Systematik und Disziplin kann sehr schnell dazu führen, dass der Mehrwert von gutem RE in Projekten verdeckt wird oder sogar dazu führen, dass RE als Disziplin für unnötig oder sogar schädlich erachtet wird.

Die *Spezifikation und das Management* von Anforderungen sind die wesentlichen Tätigkeiten im RE bzw. die Tätigkeiten, die ein Außenstehender beobachten kann. Eine Person schreibt etwas auf (spezifiziert) bzw. kümmert sich darum, dass das Aufgeschriebene verwaltet und weiterentwickelt wird. Der Fokus auf diese beiden Tätigkeiten führt allerdings häufig zu dem Missverständnis, dass das RE lediglich eine passive Buchhaltertätigkeit ist, die die Ergebnisse anderer Personen aufnimmt und verwaltet. Diese Sichtweise war für eine gewisse Periode der Softwareentwicklung durchaus passend. Mit der zunehmenden Bedeutung von Software und ihrem zunehmenden Anteil an innovativen Entwicklungen wird auch das RE zu einer kreativen Arbeit. Diesen

Punkt werden wir im weiteren Verlauf des Buchs erneut aufnehmen (siehe Abschnitt 3.2).

Die relevanten Anforderungen zu kennen wird als Ziel des RE genannt. Dieses Ziel erscheint auf den ersten Blick als trivial, ist allerdings ein Problem in Softwareprojekten, da nicht immer alle relevanten Anforderungen als bekannt vorausgesetzt werden dürfen. Weiterhin ist das Adjektiv *relevant* von großer Bedeutung für die Arbeit im RE, denn häufig enthalten Anforderungsdokumente sehr viele Informationen, die eben nicht relevant für das geplante System sind. Für die Erfüllung dieses Ziels wird im RE eine eigene Aktivität vorgesehen: Das Ermitteln von Anforderungen (siehe Kapitel 8).

Als nächstes Ziel wird *Konsens unter den Stakeholdern über die Anforderungen herzustellen* benannt. Mit diesem Ziel wird die aktive Rolle des RE im Gestaltungsprozess einer Software unterstrichen. Stakeholder haben nicht selten unterschiedliche Ansichten über die Ausgestaltung einer Software. Dadurch können Konflikte über die Anforderungen entstehen, die im RE bearbeitet und gelöst werden müssen. Diese Aufgabe bedingt, dass im RE handelnde Personen eine Mediatorenrolle einnehmen und in der Behandlung und Lösung von Konflikten geschult sein müssen. Für die Erfüllung dieses Ziels ist im RE ebenfalls eine eigene Aktivität vorgesehen: Die Abstimmung von Anforderungen (siehe Kapitel 11).

Anforderungen konform zu vorgegebenen Standards zu dokumentieren. Dieses Ziel soll deutlich machen, dass die Dokumentation von Anforderungen gewissen Standards folgen sollte und in vielen Projekten sogar folgen muss. Der Zwang zu einer standardisierten Dokumentation entsteht insbesondere bei Softwareprojekten, die in gesetzlich regulierten Branchen durchgeführt werden. Typische Beispiele sind die Medizintechnik, Teilbereiche der Automobilindustrie oder die Luft- und Raumfahrt. Das Einhalten gewisser Standards bei der Dokumentation von Anforderungen ist kein Selbstzweck und dient keinesfalls nur zur Erfüllung von gesetzlichen Vorgaben. Standards in diesen Branchen sollen die hohen Qualitätsansprüche an die zu erstellenden Systeme garantieren und haben damit unmittelbaren Einfluss auf das RE. Es ist daher ratsam und vor allen Dingen auch sehr lehrreich, sich auch dann mit solchen Standards zu befassen, wenn man nicht in einer regulierten Branche unterwegs ist. In Kapitel 6 wird der Aufbau von Anforderungsdokumenten im Detail besprochen.

Anforderungen systematisch zu managen, dieses Ziel beschreibt definitiv eine Buchhaltertätigkeit und wird durch Außenstehende nicht immer als attraktiv angesehen. Nichtsdestotrotz ist das Management von Anforderungen das Rückgrat jedes langfristig erfolgreichen Softwareprojektes, denn durch das systematische Management von Anforderungen wird das Wissen um die Anfor-

derungen und damit das Wissen um die gewünschte Gestalt des geplanten Systems verwaltet und auf dem aktuellen Stand gehalten. Anforderungsdokumente werden häufig mit schnell veraltendem Wissen gleichgesetzt, da sich das Wissen um die Software im Rahmen der Entwicklung stets weiterentwickelt. Dies ist auch vollkommen richtig, allerdings ist der Schluss aus dieser Erkenntnis meist nicht korrekt. Häufig wird daraus geschlossen: Unser Wissen über eine Software ändert sich so schnell, dass wir das Managen der Anforderungen aufgeben. Außerdem ist das Wissen sowieso in den Köpfen und damit brauchen wir es nicht aufzuschreiben oder gar zu managen. Für die initiale Umsetzung einer Software kann diese Einstellung zum Erfolg führen, denn das Wissen um die Anforderungen ist frisch im Team und in den Köpfen.

Systematisches Anforderungsmanagement ist eine Investition, die sich erst in der zweiten Lebenshälfte einer Software auszahlt: in der Wartung und Weiterentwicklung. Hier ist das Wissen um die Anforderungen nicht mehr so frisch oder faktisch nicht mehr vorhanden und jede Form der Dokumentation ist hilfreich und wertvoll. Die Kernaufgaben eines guten Managements (oder einer guten Verwaltung) von Anforderungen werden in Kapitel 12 beschrieben.

Die wesentliche Verantwortung des RE, die Entwicklung eines Systems, das den Wünschen und Bedürfnissen der Stakeholder entspricht, wurde zu Beginn dieses Kapitels bereits erläutert. Dieser Verantwortung gegenüber steht ein äquivalentes Ziel des RE: *Die Wünsche und Bedürfnisse der Stakeholder zu verstehen, zu dokumentieren*. Dieses Ziel macht deutlich, dass die Aufgabe des RE weit über das reine Aufschreiben von Informationen hinausgeht. RE setzt voraus, dass sich die handelnden Personen mit den Wünschen und Bedürfnissen der Stakeholder auseinandersetzen, diese tatsächlich begreifen und verstehen. Diese Empathie für die Stakeholder ist eine zwingende Voraussetzung für die Konzeption hochwertiger Software. Ohne Empathie, d. h. ohne ein tiefes Verständnis der Wünsche und Bedürfnisse der Stakeholder ist die Entwicklung eines passgenauen und hochwertigen Softwaresystems reiner Zufall. Diese Erkenntnis ist in der Design-Literatur hinreichend bekannt [Kel02] und setzt sich zum Glück immer mehr in der stark technisch geprägten Softwareindustrie durch.

3.2 40 Jahre RE in der Softwareentwicklung – Eine kurze subjektive Zusammenfassung der Geschichte des RE

Eine systematische Untersuchung zur Geschichte des RE lag während der Entstehung dieses Buches in der Literatur nicht vor. Dies mag daran liegen, dass die Informatik als Oberdisziplin selbst noch keine 100 Jahre alt ist. Nichtsdestotrotz kann auch einer vergleichsweise jungen Disziplin eine historische Perspektive nicht schaden. Insbesondere für Neulinge im Thema kann ein solcher Überblick hilfreich sein, um den Status quo besser verstehen zu können. Daher wird im Folgenden ein durch die Autoren vermutlich sehr subjektiv geprägter Überblick über die Geschichte des RE gegeben.

Die Vorzeit des RE lässt sich dadurch charakterisieren, dass Softwareentwicklung als eine Hilfstätigkeit angesehen wurde. Ursache hierfür war, dass Computer als Hilfsmittel in etablierten Disziplinen verwendet wurden, um komplexe Berechnungen durchzuführen (bspw. in der Physik oder der Mathematik). Die in den jeweiligen Disziplinen handelnden Personen haben die Software im Wesentlichen selbst konzipiert und entwickelt. Damit entfiel die Frage nach den Anforderungen und der Konzeption der Software, denn die handelnden Personen wussten ziemlich genau, was sie vom Computer erwartet haben und dies wurde mit den verfügbaren Programmieretechniken umgesetzt.

Diese Situation änderte sich, als die Programmiersprachen und Computertechnologien so mächtig und komplex wurden, dass das Beherrschen der Technologie nicht mehr wirklich sinnvoll als Hobby oder Nebentätigkeit ausgeführt werden konnte. Von diesem (vermutlich nicht exakt bestimmbar) Moment an, war der Beruf des Softwareentwicklers oder Programmierers geboren. Mit einem Mal bestand nun das Problem darin, dem Programmierer das notwendige Wissen zu vermitteln, was von der zu erstellenden Software erwartet wird. Dieses notwendige Wissen wurde nun als Anforderung bezeichnet und die Erfassung, Kommunikation und Verwaltung dieses Wissens waren die Anfänge des RE. Die frühesten nachvollziehbaren Publikationen zu diesem Gebiet gehen in die Mitte der 1970er Jahre zurück und wurden auf der International Conference on Software Engineering (siehe [Bel76]) bzw. in der Zeitschrift IEEE Transactions on Software Engineering (siehe [Ros77]) veröffentlicht.

Kennzeichnend für die Anfänge des RE war der Umstand, dass Computer und Software dazu genutzt wurden, bestehende und gut verstandene Prozesse und Aktivitäten zu automatisieren bzw. in Software abzubilden. Vorreiter waren hier sicherlich Banken und Versicherungen, die in Computern und Software ein großes Potenzial zur Steigerung der Effizienz ihrer Prozesse gesehen haben.

Für das RE bedeutete dieser Umstand, dass es immer ein Gegenüber gab, der die Aufgaben, Tätigkeiten oder notwendigen Funktionen bereits intensiv durchdacht und verstanden hat (bspw. das Führen von Bankkonten oder die Verwaltung von Versicherungen). Die Herausforderung für die Softwareentwicklung bestand eher darin, die gewünschten Aufgaben mit der bestehenden Technologie zu realisieren. Dem RE kam in dieser Situation die (zugegeben stark vereinfachte) Aufgabe zu, das Wissen der Experten zu erfassen und für die Entwickler verfügbar zu machen. In solchen Situationen konnte und wurde das RE als eine passive Tätigkeit aufgefasst, die Information aufgenommen und weiterverarbeitet hat. Entsprechend wurden in Forschung und Industrie Methoden und Techniken entwickelt, um diesen Aufgaben Rechnung zu tragen (siehe bspw. [vLa00] oder [Che07] für einen Überblick zu wissenschaftlichen Arbeiten im RE).

Diese Situation begann sich zu ändern, als Software nicht mehr nur als reine Umsetzungstechnologie verwendet wurde, sondern selbst zum Gegenstand und Treiber von neuen Ideen und Innovationen geworden ist. Auch hier ist eine genaue Datierung nicht wirklich möglich. Sicherlich kann der Durchbruch des Internets als Meilenstein in dieser Entwicklung gesehen werden.

Was genau hat sich geändert? Die wesentliche Voraussetzung des RE hat sich geändert. Auf einmal waren nicht mehr unmittelbar ein oder mehr Personen verfügbar, die den gegebenen Sachverhalt so gut durchdrungen hatten, dass man ihr Wissen einfach nur noch für die Entwicklung verfügbar machen musste. Nun standen vielmehr das gemeinsame Arbeiten und das gemeinsame Gestalten der geplanten Software im Vordergrund. Augenscheinlich musste sich das RE als Disziplin neu erfinden, zumindest in Projektsituationen, in denen ein signifikanter Anteil an Innovation vorhanden war. Konkret bedeutet dies, dass Kreativitätstechniken einen stärkeren Anteil bei der Ermittlung von Anforderungen bekommen. Weiterhin ist die Planung des RE-Prozesses mit mehr Unsicherheiten verbunden, da die Konzeption eines innovativen Systems viel schwieriger zu planen ist als die Konzeption eines verstandenen Systems.

3.3 Einbettung von RE und seine Bedeutung für erfolgreiche Softwareentwicklung

Bevor die Details und Spezialitäten des RE betrachtet werden, soll die Bedeutung von RE für erfolgreiche Softwareentwicklung dargestellt werden.

Zunächst betrachten wir die Bedeutung des RE aus einer strukturellen Perspektive für den Erfolg von Softwareprojekten. Abbildung 2 zeigt einen stark vereinfachten Blick auf die Aktivitäten der Softwareentwicklung und die Abhängigkeiten vom RE:

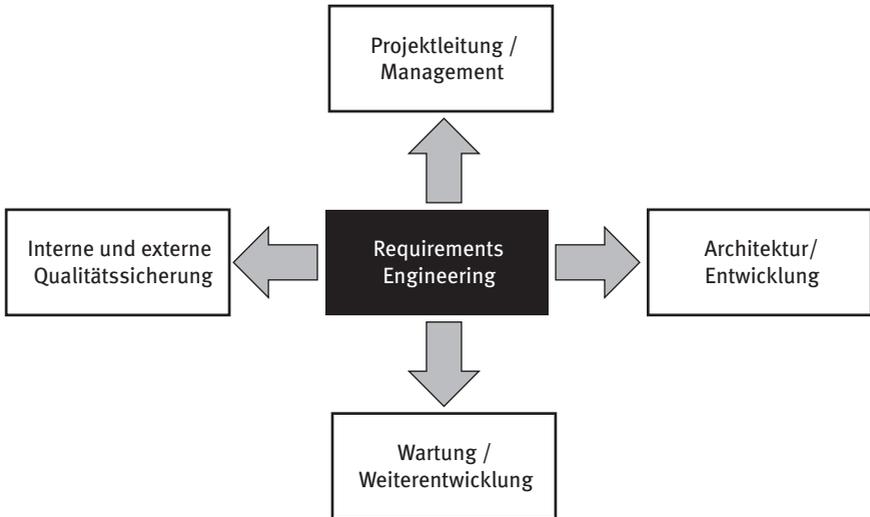


Abbildung 2: Abnehmer von Anforderungen in der Softwareentwicklung

Anforderungen werden im Rahmen der Projektleitung/im Management für die Planung des Projektes, für die Aufwandsabschätzung und für weitere Managementaktivitäten (bspw. Entscheidungen zur Unternehmensstrategie in Bezug auf Software) benötigt. Die Architektur und Entwicklung benötigt die Anforderungen, um zu wissen, was zu implementieren ist. Die interne Qualitätssicherung (damit ist insbesondere der Softwaretest im Projekt gemeint) benötigt die Anforderungen als Referenz für die Entwicklung der Tests. Ohne Anforderungen wäre ein systematisches Testen nicht möglich. Die externe Qualitätssicherung (damit sind insbesondere Zertifizierungsorganisationen in regulierten Branchen, bspw. der Medizintechnik gemeint) setzt ebenfalls auf den Anforderungen auf, um die Qualität der entwickelten Software im Vergleich zu den gesetzlich festgelegten Standards zu beurteilen. Schlussendlich sind Anforderungen der Ausgangspunkt für die Wartung und Weiterentwicklung von Software. Unterm Strich kann man festhalten, dass das RE Informationsgeber für alle Aktivitäten der Softwareentwicklung ist.

Zu den Klassikern in der Literatur zur Ökonomie der Softwareentwicklung zählen die Arbeiten von Barry Boehm [Boe81]. Boehm konnte zeigen, dass über die Hälfte aller Fehler in Softwareprojekten in den Anforderungen gemacht werden. Darüber hinaus konnte er zeigen, dass die Kosten für die Behebung von Anforderungsfehlern mit zunehmendem Projektfortschritt nahezu exponentiell steigen. Die Behebung eines Anforderungsfehlers in der Umsetzung

kostet bspw. das 100-Fache im Vergleich zur Behebung im RE. Auch wenn diese Zahlen schon einige Jahre alt sind, ist das Prinzip hinter dieser Kostenexplosion auch heute noch gültig und einsichtig. Je weiter fortgeschritten die Entwicklung einer Software ist, desto teurer ist die Behebung von Fehlern. Umgekehrt macht diese Kostenexplosion deutlich, wie wichtig die in Abschnitt 3.1 beschriebene Systematik und Disziplin ist, um Fehler in den Anforderungen möglichst zu vermeiden.

Eine weitere, oft zitierte Quelle für die Bedeutung des RE in der Softwareentwicklung sind die Chaos Reports der Standish Group². Die Chaos Reports werden seit 1994 regelmäßig veröffentlicht und berichten über den Stand der Praxis in der Softwareentwicklung, unter anderem über die Erfolgsraten bzw. Abbruchraten von Softwareprojekten und über die Gründe und Ursachen für Erfolge bzw. Fehlschläge. Gutes RE bzw. schlechtes RE findet sich in den Statistiken des Chaos Reports mit hoher Regelmäßigkeit als Ursache für Projekterfolg und -fehlschläge.

Eine dritte Perspektive auf die Bedeutung von RE liefert die Literatur zur Aufwandsschätzung für Softwareprojekte. Die Literatur besagt (siehe bspw. [McC06]), dass die RE-Kompetenzen der für das RE eingesetzten Personen den Gesamtaufwand eines Projektes um über 40% vergrößern (bei Personen mit geringer RE-Kompetenz) bzw. um bis zu 30% reduzieren können (bei hoher RE-Kompetenz). Hierbei ist wichtig zu betonen, dass sich die Prozentzahlen auf den Gesamtaufwand des Projektes beziehen und nicht nur auf den RE-Aufwand.

Diese Auflistung macht sehr deutlich, warum Budget für gutes RE und insbesondere für gut qualifiziertes Personal sinnvoll investiertes Geld ist.

3.4 Haupttätigkeiten und Arbeitsergebnisse im Überblick

Die bisherigen Abschnitte haben den groben Rahmen des RE abgesteckt.

Für die Aufteilung des RE in Aktivitäten werden die Standards des IREB e.V. herangezogen. Das IREB teilt das RE in die folgenden Haupttätigkeiten auf [IRE15a]:

- Ermitteln,
- Dokumentieren,
- Prüfen,
- Abstimmen und
- Verwalten.

² Siehe <https://www.standishgroup.com>

Bei der *Ermittlung von Anforderungen* geht es darum, die Anforderungen der Stakeholder zu erfassen, gemeinsam mit den Stakeholdern Anforderungen zu entwickeln und Anforderungen zu verfeinern. Neben den Stakeholdern können weitere Anforderungsquellen, d. h. Dokumente und bestehende Systeme herangezogen werden. Weitere Details zu dieser Aktivität werden in Abschnitt 8 beschrieben.

Das *Dokumentieren von Anforderungen* umfasst sämtliche Tätigkeiten, die sich mit der Kunst des Aufschreibens bzw. Spezifizierens von Anforderungen befassen. Hierbei ist die Tätigkeit des Aufschreibens von der Form zu trennen, d. h. wie die Inhalte eines Anforderungsdokuments zu gestalten sind. Die Tätigkeit der Dokumentation ist Gegenstand von Abschnitt 9, das Handwerkzeug zum Aufschreiben wird in Kapitel 6 diskutiert.

Das *Prüfen von Anforderungen* ist die Aktivität zur Qualitätssicherung von Anforderungen. Die Qualitätssicherung von Anforderungen kennt verschiedene Techniken und bezieht sich auf drei Qualitätsaspekte, insbesondere auf inhaltliche Richtigkeit und auf die Verwendung der richtigen Form. Weitere Details zur Prüfung von Anforderungen werden in Kapitel 10 beschrieben.

Das *Abstimmen von Anforderungen* ist gemäß dem IREB-Standard ein Teil der Prüfung von Anforderungen, da die Abgestimmtheit (d. h. das Fehlen von Konflikten in den Anforderungen) ein Qualitätskriterium im Rahmen der Prüfung von Anforderungen ist. Zur Abstimmung von Anforderungen gehört jedoch auch das Bearbeiten von Konflikten. Die Bearbeitung von Konflikten wird in Abschnitt 11 als eigenständiges Thema betrachtet.

Die *Verwaltung von Anforderungen* ist die Aktivität der Buchhaltung und Buchführung von Anforderungen. Diese Analogie führt häufig zu einem fälschlicherweise negativen Beigeschmack. In Abschnitt 3.1 wurde bereits ausgeführt, dass diese negative Sicht auf die Verwaltung von Anforderungen unbegründet ist und die systematische Verwaltung von Anforderungen ein wertvoller Bestandteil für nachhaltig erfolgreiche Softwareprojekte ist. Details zur Verwaltung von Anforderungen werden in Abschnitt 12 dargelegt

Die soeben beschriebene Aufteilung in Haupttätigkeiten wirkt auf den ersten Blick recht künstlich. Dieser spontane Eindruck ist vollkommen richtig, denn die Aufteilung ist in erster Linie ein didaktisches Mittel, um die notwendigen Kompetenzen zur Durchführung der Tätigkeiten leichter zu strukturieren und zu vermitteln.

In der praktischen Tätigkeit werden die Hauptaktivitäten nur selten in der Reinform durchgeführt, vielmehr werden Kombinationen der Tätigkeiten ausgeführt. Betrachtet man bspw. ein Interview mit zwei Kundenvertretern, dann

wird dieses Interview primär der Erhebung von Anforderungen dienen. Wenn im Rahmen des Interviews jedoch konfliktäre Anforderungen von Interviewpartnern geäußert werden, kann der Interviewer in die Abstimmung von Anforderungen wechseln und den Konflikt zwischen den Beteiligten bearbeiten.

Häufig wird die Reihenfolge der Darstellung der Aktivitäten (auch in diesem Buch ist sie wie folgt: Ermittlung, Dokumentation, Prüfung, Abstimmung und Verwaltung) als Prozessvorgabe missverstanden, d. h. erst kommt die Ermittlung, dann die Dokumentation, dann die Prüfung usw. In aller Deutlichkeit:

Diese Sichtweise ist nicht beabsichtigt und vor allen Dingen *falsch*.

Details zur Planung und Durchführung von RE-Aktivitäten werden in Kapitel 13 beschrieben.

Neben der Betrachtung der Hauptaktivitäten des RE legt der IREB-Standard einen großen Wert auf die Form der Dokumentation von Anforderungen. Dieser Schwerpunkt ist von Bedeutung, da die Literatur verschiedenste Arten der Dokumentation kennt – man könnte hier sogar fast von Stilrichtungen sprechen. Die Formen der Dokumentation können ganz grob in natürlichsprachige und modellbasierte Formen unterschieden werden. Die Auswahl der Dokumentationsform wird als ein wesentlicher Erfolgsfaktor für das RE angesehen und sollte daher immer mit Blick auf die vorliegenden Projektgegebenheiten festgelegt werden. Details zur Dokumentationsform folgen in Kapitel 6.