

7.

Auflage



Christof Ebert

Systematisches Requirements Engineering

Anforderungen ermitteln, dokumentieren,
analysieren und verwalten

dpunkt.verlag



Christof Ebert ist Gründer und Geschäftsführer von Vector Consulting Services. Er unterstützt Unternehmen bei Strategie, Technologie und agiler Transformation. Zuvor war er zwölf Jahre bei einem IT-Konzern in weltweiten Führungsaufgaben tätig. Er arbeitet in Aufsichtsgremien von Unternehmen und stimuliert Start-ups als Business Angel. An der Universität Stuttgart und der Sorbonne in Paris ist er Professor und hält Patente in Softwaretechnik und AI. Er wirkt in den Herausgeber-Komitees von IEEE Software, Software Quality Journal und dem Journal of Systems and Software. In seiner Freizeit spielt er als Musiker Keyboard und Kirchenorgel und engagiert sich im sozialen Bereich.

Folgen Sie ihm auf Twitter und LinkedIn: [@ChristofEbert](#).

Kontakt: christof.ebert@vector.com, www.christofebert.de

Homepage des Buches: www.requirements-excellence.com

Copyright und Urheberrechte:

Die durch die dpunkt.verlag GmbH vertriebenen digitalen Inhalte sind urheberrechtlich geschützt. Der Nutzer verpflichtet sich, die Urheberrechte anzuerkennen und einzuhalten. Es werden keine Urheber-, Nutzungs- und sonstigen Schutzrechte an den Inhalten auf den Nutzer übertragen. Der Nutzer ist nur berechtigt, den abgerufenen Inhalt zu eigenen Zwecken zu nutzen. Er ist nicht berechtigt, den Inhalt im Internet, in Intranets, in Extranets oder sonst wie Dritten zur Verwertung zur Verfügung zu stellen. Eine öffentliche Wiedergabe oder sonstige Weiterveröffentlichung und eine gewerbliche Vervielfältigung der Inhalte wird ausdrücklich ausgeschlossen. Der Nutzer darf Urheberrechtsvermerke, Markenzeichen und andere Rechtsvorbehalte im abgerufenen Inhalt nicht entfernen.

Christof Ebert

Systematisches Requirements Engineering

**Anforderungen ermitteln, dokumentieren,
analysieren und verwalten**

7., überarbeitete und aktualisierte Auflage



dpunkt.verlag

Christof Ebert

christof.ebert@vector.com, www.christofebert.de

Lektorat: Christa Preisendanz

Lektoratsassistentz: Julia Griebel

Copy-Editing: Ursula Zimpfer, Herrenberg

Layout & Satz: Birgit Bäuerlein

Herstellung: Stefanie Weidner

Umschlaggestaltung: Helmut Kraus, *www.exclam.de*

Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-86490-919-1

PDF 978-3-96910-768-3

ePub 978-3-96910-769-0

mobi 978-3-96910-770-6

7., überarbeitete und aktualisierte Auflage 2022

Copyright © 2022 dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Hinweis:

Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.

Schreiben Sie uns:

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: hallo@dpunkt.de.



Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Meinen Eltern Elfriede und Otto
und meinem Lehrer Rudolf Lauber.
Sie haben mir gezeigt,
dass ein *Werk* nur Wert schafft,
wenn die *Anforderungen* verstanden und umgesetzt sind.

Vorwort zur 7. Auflage

*Wer immer tut, was er schon kann,
bleibt immer das, was er schon ist.*

Henry Ford

Wie kaum ein anderer reüssierte Henry Ford als Ingenieur und Unternehmer. Er rüttelt uns mit seinem Weckruf auf – gerade jetzt im neuen Normal. Viele haben die Pandemie als willkommene Entschuldigung benutzt, sich zu Hause einzuigeln und zu stagnieren. Manche hoffen selbst heute noch auf die Rückkehr in frühere Zeiten. Damit sind wir gerade in einem Hochlohnland nicht wettbewerbsfähig. In den Worten des großen Aufklärers Immanuel Kant müssen wir heraus aus der selbstverschuldeten Unmündigkeit. Das Buch zeigt, wie Sie Anforderungen setzen und verfolgen. Denn im neuen Normal ist Neues normal. Anforderungen sind dafür die Basis.

Produkte sind, was wir liefern. Wert ist, was die Kunden wahrnehmen. Das läuft oft stark auseinander. Mehr Requirements bedeuten nicht unbedingt mehr Wert, aber definitiv mehr Kosten und Komplexität. Man startet mit unklaren Bedürfnissen und Zielen, rudimentären Ideen, Luftschlössern aus dem Vertrieb, nicht bewerteten Abhängigkeiten und vagen Vermutungen. Später rächt sich diese Nachlässigkeit in Gestalt von Änderungen und Nacharbeit. Entwickeln Sie nicht nur Anforderungen, sondern verfolgen Sie das Ziel, für Ihre Kunden Wert zu schaffen – und für Ihre Mitbewerber Hindernisse.

Mit diesem Buch bekommen Sie alles für gutes Requirements Engineering. Branchenübergreifend – von der Konzeption bis zur Evolution. Es ist aus der Praxis für die Praxis geschrieben und zeigt Ihnen, wie Sie Wert für sich und Ihr Unternehmen schaffen – und erhalten. Es adressiert Requirements Engineering für unterschiedliche Anwendungsbereiche, sei es agile Entwicklung versus formale Vorgehensweise für komplexe Projekte.

Diese 7. Auflage unterstreicht agile Requirements Engineering, testorientierte Anforderungen, Systems Engineering und aktuelle Trends. Sie enthält viele neue Abbildungen und ein komplett überarbeitetes Glossar. Mit aktualisierten Beispielen wird die Rolle des Requirements Engineering im Projekt gezeigt. Das Buch berück-

sichtigt den aktuellen Lehrplan des IREB-Zertifizierungsprogramms. Die URL-Verweise sind aktualisiert. Alle Templates sind online verfügbar. So bleibt diese Neuauflage für alle Lesenden eine wichtige Referenz auf dem Schreibtisch.

Die meisten Projekte umfassen Änderungen existierender Systeme. Das Buch adressiert diese Situation und bewegt sich nicht akademisch auf der »grünen Wiese«. Ein Selbsttest hilft bei der Bewertung Ihrer Fähigkeiten im Requirements Engineering. Der Nutzen und ROI von Requirements Engineering wird an verschiedenen Stellen herausgestellt. Damit können Sie zielorientiert eigene Herausforderungen adressieren. Die vorgestellten Vorlagen und viele aktuelle Tipps sowie Hinweise auf Trainings sind auf der Homepage dieses Buches verfügbar: www.requirements-excellence.com.

Mein Dank geht an Mitarbeitende und Kundenunternehmen von Vector Consulting, mit denen wir die genannten Praktiken umsetzen und ständig verbessern. Hervorheben möchte ich Frank Kirschke-Biller und Arnold Rudorfer, die mich in den Fallstudien bei Ford und Siemens unterstützten. Immer wieder erhalte ich Verbesserungsvorschläge von Lesenden. Danke dafür! Für bessere Lesbarkeit verwende ich dudengerechte Formen. Danke schließlich an Christa Preisendanz und den dpunkt.verlag für die gewohnte Professionalität.

Mein Freund Al Davis hatte mich als damaliger Chefredakteur von IEEE Software dazu angeregt, Requirements Engineering systematisch und dennoch agil (!) in der Industrie zu verankern. Dank an Al und die vielen, die dieses Thema in der Praxis antreiben, wie Ian Alexander, Anthony Finkelstein, Don Gause, Michael Goedicke, Martin Glinz, Matthias Jarke, Neil Maiden, Barbara Paech, Klaus Pohl, Mary Popenieck, Charles Symons, Suzanne Robertson, Ian Sommerville und Karl Wiegers.

Ihnen, liebe Leserinnen und Leser, wünsche ich viel Erfolg bei allem, was Sie neu anpacken! Dieses Buch gibt Ihnen mit lösungsorientiertem Denken die nötigen Impulse, um Ziele richtig offensiv anzugehen.

Christof Ebert
Stuttgart, 22.2.22

Stimmen zum Buch

»... ein hervorragendes Buch für den praxisnahen Einstieg in die vielschichtigen Themenkomplexe der Anforderungsanalyse und des Anforderungsmanagements.«

Chip.de (Juli 2010 zur 2. Auflage)

*

»Systematisches Requirements Engineering« ist die wertvollste Anleitung für Anforderungen, die Sie finden können. Christof Ebert deckt die gesamte Landschaft von Praktiken ab, die ein Requirements-Ingenieur, Projektmanager oder Produktmanager kennen sollte. Für Praktiker und Manager gleichermaßen kann ich dieses Buch nicht hoch genug empfehlen. Ich war schon immer ein Fan von seinen Schriften – und werde es auch weiterhin sein!«

*Alan M. Davis, Entrepreneur und Professor
University of Colorado, Colorado Springs*

*

»Christof Ebert schafft es, sowohl Frischlingen als auch alten Hasen Neues beizubringen. Anschaulich und ohne Besserwisserei zeigt er, wie Requirements Engineering im Unternehmen optimal aufgesetzt wird. Ein Muss für Entscheider und alle, die Erfolg bei Softwareprojekten haben wollen.«

*Gerhard Mack
Chief Technology Officer, Vodafone*

*

»Christof Ebert hat ein Talent dafür, zum Kern der Sache zu kommen und die einfache (aber schwer erkennbare) Wahrheit freizulegen. Danke für die immer klar und elegant formulierten Ratschläge!«

*Suzanne Robertson
Gründerin und Principal, The Atlantic Systems Guild Ltd.*

*

»Inzwischen ein Klassiker für den systematischen Umgang mit Anforderungen. Geschrieben von einem Praktiker für die Praxis – einfach, verständlich und anwendbar! Dass der Autor sein Metier versteht, durfte ich in einem gemeinsamen Praxisprojekt hautnah erleben.«

Hans Leibbrand
ehem. Chief Operating Officer und Vorstand, Thales

*

»Mit den Anforderungen werden die Weichen gestellt für den Projekterfolg – oder Misserfolg. Aus fehlerhaften, unvollständigen oder widersprüchlichen Anforderungen wird niemals gute Software entstehen. Das vorliegende Buch hilft, den richtigen Einstieg in die Softwareentwicklung zu finden und die vielen Klippen zielgerichtet zu vermeiden.«

Peter Liggesmeyer
Direktor Fraunhofer IESE, ehem. Vorsitzender der Gesellschaft für Informatik

Inhaltsübersicht

1	Motivation	1
2	Requirements Engineering – kurz und knapp	21
3	Anforderungen ermitteln	51
4	Anforderungen dokumentieren	99
5	Anforderungen modellieren und analysieren	123
6	Anforderungen prüfen	175
7	Anforderungen abstimmen	201
8	Anforderungen verwalten	237
9	Agiles Requirements Engineering	265
10	Werkzeuge	297
11	Requirements Engineering leben	323
12	Soft Skills und persönliche Entwicklung	367
13	Stand der Technik und Trends	383
	Anhang	409
A	Internetressourcen	411
B	Glossar	413
C	Literatur	445
	Index	453

Inhaltsverzeichnis

1	Motivation	1
1.1	Warum ein Buch über Requirements Engineering?	1
1.2	Projekte scheitern wegen unzureichender Anforderungen	4
1.3	Wirtschaftlicher Nutzen und Return on Investment (ROI)	10
1.4	Wie Sie von diesem Buch profitieren	13
1.5	Selbsttest	16
1.6	Ein Blick über den Tellerrand	18
2	Requirements Engineering – kurz und knapp	21
2.1	Was ist eine Anforderung?	21
2.2	Perspektiven: vom Markt zur Realisierung	24
2.3	Arten von Anforderungen	30
2.4	Was ist Requirements Engineering?	34
2.5	Requirements Engineering in der Praxis	36
2.6	Terminologie	40
2.7	Durchgängiges Beispiel: iHome	42
2.8	Tipps für die Praxis	48
2.9	Fragen und Impulse	49
3	Anforderungen ermitteln	51
3.1	Ziel und Nutzen	51
3.2	Bedürfnisse verstehen, Ziele vereinbaren	52
3.3	Stakeholder managen	55
3.4	In 10 Schritten zu guten Anforderungen	65
3.5	Qualitätsanforderungen und Randbedingungen	80
3.6	Fallstudie: Security Requirements Engineering	91
3.7	Checkliste für die Anforderungsermittlung	95
3.8	Tipps für die Praxis	97
3.9	Fragen und Impulse	98

4	Anforderungen dokumentieren	99
4.1	Ziel und Nutzen	99
4.2	Lasten und Pflichten: vom Was zum Wie	101
4.3	Dokumentation und Vorlagen	104
4.4	Struktur und Lesbarkeit	111
4.5	Attribute und Filter	118
4.6	Glossar	119
4.7	Checkliste für die Dokumentation	120
4.8	Tipps für die Praxis	121
4.9	Fragen und Impulse	122
5	Anforderungen modellieren und analysieren	123
5.1	Ziel und Nutzen	123
5.2	Modelle und Methoden	125
5.3	Systems Engineering und Architektur	132
5.4	UML, SysML und BPMN	137
5.5	Aufwandsschätzung	158
5.6	Analyse in zehn Schritten	167
5.7	Checkliste für die Anforderungsanalyse	170
5.8	Tipps für die Praxis	172
5.9	Fragen und Impulse	173
6	Anforderungen prüfen	175
6.1	Ziel und Nutzen	175
6.2	Qualitätskriterien für Anforderungen	177
6.3	Verfahren zur Prüfung	179
6.4	Test-Driven Requirements Engineering (TDRE)	185
6.5	Kriterien für Testende und Abnahme	191
6.6	Checkliste zur Prüfung von Anforderungen	194
6.7	Tipps für die Praxis	200
6.8	Fragen und Impulse	200
7	Anforderungen abstimmen	201
7.1	Ziel und Nutzen	201
7.2	Abstimmung im Kernteam	203
7.3	Risiken abschwächen	205
7.4	Priorisierung von Anforderungen	211

7.5	Recht, Compliance und Haftung	216
7.6	Verträge und Vertragsmodelle	224
7.7	Checkliste für Abstimmung und Verträge	230
7.8	Tipps für die Praxis	234
7.9	Fragen und Impulse	235
8	Anforderungen verwalten	237
8.1	Ziel und Nutzen	237
8.2	Änderungsmanagement	239
8.3	Verfolgbarkeit (Traceability)	244
8.4	Wartung und Altsysteme	248
8.5	Versionierung und Varianten von Anforderungen	251
8.6	Kennzahlen und KPI	253
8.7	Checkliste für die Verwaltung	261
8.8	Tipps für die Praxis	262
8.9	Fragen und Impulse	263
9	Agiles Requirements Engineering	265
9.1	Agile Entwicklung	265
9.2	Komplexität beherrschen	272
9.3	Praxis des agilen RE	276
9.4	Design Thinking	281
9.5	Skalierbare Agilität	283
9.6	Fallstudie: Agiles Systems Engineering bei Ford	286
9.7	Fallstudie: Agiles RE bei Siemens	290
9.8	Tipps für die Praxis	294
9.9	Fragen und Impulse	295
10	Werkzeuge	297
10.1	Ziel und Nutzen	297
10.2	Werkzeuge und Bewertung	298
10.3	Praxis: von DOORS bis PREEvision	307
10.4	Werkzeuge einführen	311
10.5	Checkliste für Werkzeuge	313
10.6	Tipps für die Praxis	321
10.7	Fragen und Impulse	322

11	Requirements Engineering leben	323
11.1	Organisation	323
11.2	Projektmanagement	330
11.3	Produktmanagement	334
11.4	Lieferantenmanagement	341
11.5	Serviceorientiertes RE	347
11.6	Fallstudie: Funktionsmodellierung und Produktlinien	350
11.7	Fallstudie: Besseres RE in 10 Schritten	356
11.8	Tipps für die Praxis	364
11.9	Fragen und Impulse	365
12	Soft Skills und persönliche Entwicklung	367
12.1	Aufgaben des »Requirements Engineer«	367
12.2	IREB und Zertifizierung	371
12.3	Soft Skills	374
12.4	Konflikte lösen	379
12.5	Tipps für Ihre persönliche Entwicklung	381
12.6	Fragen und Impulse	382
13	Stand der Technik und Trends	383
13.1	Der »Stand der Technik«	383
13.2	Standards und Normen	385
13.3	Benchmarks, Faustregeln und Kennzahlen	390
13.4	Trends in der IT und Software	394
13.5	Trends im Requirements Engineering	398
13.6	Top-10-Tipps	407
Anhang		409
A	Internetressourcen	411
B	Glossar	413
C	Literatur	445
	Index	453

1 Motivation

Ihr Nutzen aus diesem Kapitel:

»Wenn du nicht weißt, wohin du gehen willst, kannst du jeden Weg nehmen.« Alice im Wunderland wusste es, wie auch viele Denker vor und nach ihr. Klare Ziele werden erreicht, unklare Ziele werden sicher verfehlt. In diesem Kapitel zeige ich, wie Sie mit Requirements Engineering Ziele schrittweise klären und kommunizieren. Insbesondere möchte ich den Nutzen eines systematischen Requirements Engineering darstellen. Beantworten Sie die folgenden Fragen einfach spontan und ehrlich: Sind Ihre Anforderungen strukturiert und testbar dokumentiert? Hat Ihr derzeitiges Projekt einen expliziten Business Case, der auch geprüft wird? Gibt es für jede Anforderung eine kurze Begründung, die den Nutzen und Wert beschreibt? Falls nicht, ist das Buch das Richtige für Sie. Falls ja, lesen Sie die Fragen nochmals und gehen Sie in sich.

1.1 Warum ein Buch über Requirements Engineering?

Das Problem vieler Unternehmen ist, dass zu viele Funktionen verkauft werden, und zu wenig Wert. Oft zerbrechen wir uns den Kopf über eine Lösung, ohne verstanden zu haben, welches Problem wir lösen müssen. Wir gehen zu Besprechungen, ohne zu hinterfragen, was sie bringen. Wir entwickeln Funktionen und können deren Wert nicht darstellen. Wir optimieren und bemühen uns ständig, bessere Produkte zu entwickeln – und fühlen uns doch wie im Hamsterrad. Während des Projekts wundern wir uns, dass sich die Anforderungen ständig ändern. Dabei war niemals klar, was wir eigentlich konkret erreichen wollen, und was nicht.

Zeig mir deine Anforderungen und ich sage dir, wie dein Projekt endet. So lautet eine alte Weisheit, und wir wollen sie hier gleich mal anwenden.

Abbildung 1–1 zeigt ein typisches Beispiel einer Anforderung. Sie ist in Prosa, weil das vermeintlich verständlicher ist, und wirft mehr Fragen auf, als sie beantwortet:

- Wie hängen diese unstrukturierten Funktionen zusammen?
- Bringt die Anforderung Kundennutzen und damit Profit?

- Kann die Anforderung umgesetzt werden?
- Wie ist der Status der Anforderung?
- Wer ist für die Anforderung verantwortlich?
- Wie wird die Anforderung validiert?
- Wie wird die Anforderung umgesetzt?

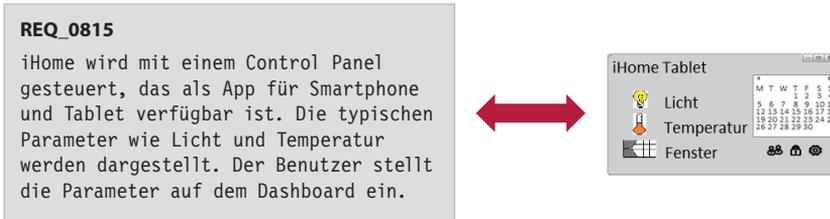


Abb. 1-1 Eine Anforderung – so viele Fragen

Software wird zunehmend komplexer. Abbildung 1–2 zeigt die Entwicklung verschiedener Softwaresysteme, die wir untersucht haben.¹ Auf der waagrechten Achse sind die Jahreszahlen angegeben, während senkrecht das Softwarevolumen in tausend Objektcodebefehlen dargestellt ist. Diese Darstellung erschien uns als die einzig praktikable, wenn wir so unterschiedliche Systeme wie Apps für Mobiltelefone und eingebettete Software vergleichen wollen. Der Umfang der Software verdoppelt sich alle zwei bis vier Jahre. Mit diesem Wachstum steigt auch der Umfang der Spezifikationen an. Gab es Anfang der Neunzigerjahre beispielsweise einige wenige Steuergeräte in einem Neuwagen mit ungefähr hundert Seiten an Spezifikationen, so sind es heute bereits fünfzig und mehr Steuergeräte mit über 100.000 Seiten an Spezifikationen. Diese schnell wachsende Komplexität fordert systematisches Requirements Engineering (RE), um die Qualität und Kosten nachhaltig kontrollieren zu können.

Ein Beispiel für hausgemachte Komplexität sind IT-Migrationsprojekte. Die Stakeholder, manchmal eingedeutscht als »Interesseneigner«, sind sich oft schnell darin einig, dass alle Funktionen des existierenden Altsystems übertragen werden müssen. Das ist ein großer Fehler. Erstens kann sowieso niemand mehr alle existierenden Altfunktionen im Zusammenhang beschreiben. Und zweitens ist gerade ein neues System die einzige Chance, alte Funktionen und Workflows über Bord zu werfen. Komplexität ist der Feind jeder Innovation. Neues entsteht nur durch Weglassen von Ballast, wie wir von Unternehmen wie Apple oder Tesla lernen können. Dass es anders geht, zeigen Start-ups und exzellente Unternehmen wie Apple. Bei ihnen lautet die erste Frage nicht, welche Komplexität noch zusätzlich entwickelt werden soll, sondern wie das Produkt hinreichend schlank bleibt.

1. Quellen der Daten: Eigene Studien des Autors, Vector Benchmarking Datenbank, NASA, Codeanalysen bei Windows und Linux. Konversionen soweit nötig gemäß den anerkannten Korrekturfaktoren [Hatton2005, Ebert2007].

Object Code in 1000 Instructions

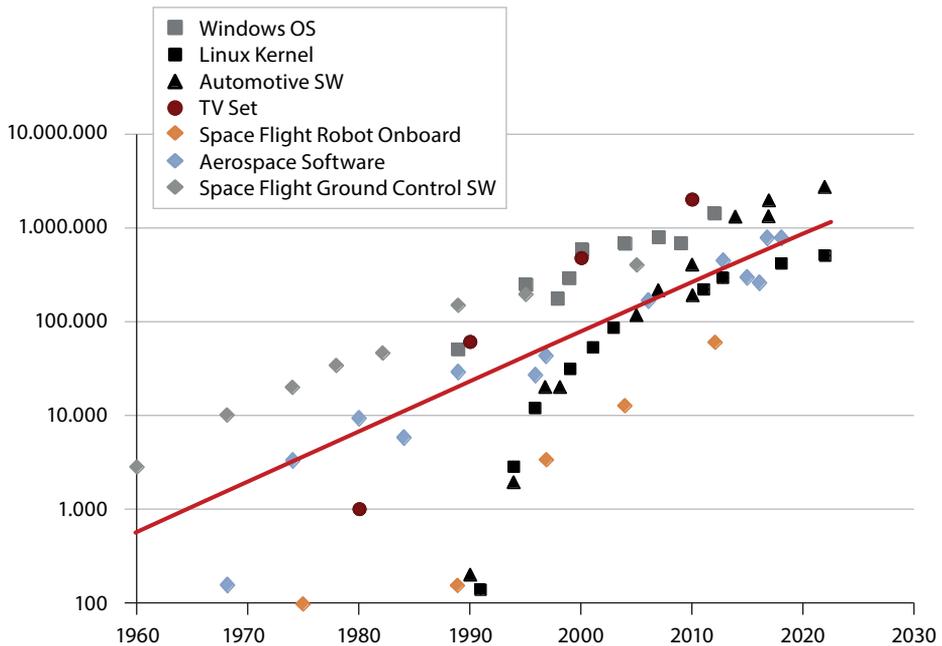


Abb. 1-2 Komplexität von Softwaresystemen über die Zeit

Erfahrene Projektmanager und Entwickler wissen, dass es erprobte Methoden sowie werkzeugunterstützte Hilfsmittel für das Requirements Engineering gibt. Häufig fehlt ihnen aber der Überblick über die Theorie und Praxis des Requirements Engineering, um die für ihre Situation passenden Methoden, Verfahren und Hilfsmittel auszuwählen, sowie die notwendige Kenntnis im Detail, um sie produktiv nutzen zu können.

Das Buch füllt diese Lücke. Es liefert umsetzungsorientiert Theorie und Praxis des Requirements Engineering. Die gängigen Verfahren der Anforderungsanalyse sind beschrieben. Die Leser erhalten Einblick in die Art und Weise, wie Anforderungen ermittelt, entwickelt, dokumentiert und im Projekt verfolgt werden. Die grundsätzlichen Methoden, Verfahren, Werkzeuge und Notationen des Requirements Engineering werden übersichtlich behandelt. Sie werden durch konkrete Beispiele aus der Projektarbeit illustriert. Notationen und Modelle sind in der Regel mit UML 2.0 beschrieben. Fallstudien demonstrieren die konkrete Umsetzung und Erfahrungen aus der Praxis.

1.2 Projekte scheitern wegen unzureichender Anforderungen

Zu viele Projekte scheitern, und Produkte erreichen die Marktziele nicht. Unzureichendes Requirements Engineering ist immer unter den Top-3-Ursachen. Die aktuelle Studie der Standish Group zeigt, dass ein gutes Drittel aller Projekte erfolgreich abgeschlossen wird. Ein Fünftel wird abgebrochen, und der Rest kommt zwar zu einem Abschluss, aber nur unter Aufgabe von ursprünglichen Zielen (Abb. 1–3) [Standish2021]. RE erhält im Schnitt nur 2–5% des Projektaufwands, aber Fehler in den Anforderungen haben die größten Effekte [Gartner2022, Lauesen2020, Standish2021, Charette2018, Garousi2019, Ebert2014a, Ebert2007].

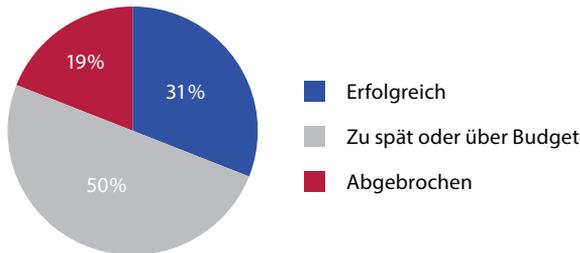


Abb. 1–3 Unzureichendes Requirements Engineering reduziert den Projekterfolg.

Ein wesentlicher Grund für Projektscheitern sind unklare Ziele. Bei einem Großteil aller abgebrochenen Projekte war unzureichendes RE ein wesentlicher Grund für das Scheitern. Häufiger wurden nur noch »unzureichende Prozesse« und »unklare Verantwortungen« genannt, aber das sind offensichtliche Allgemeinplätze.



Beispiel:

Viele Projekte scheitern, weil man versucht, die Interessen aller Stakeholder zu berücksichtigen, statt klare Ziele und Führung einzusetzen [Lauesen2020]. Aktuelle Beispiele sind die elektronische Patientenakte in Deutschland (ePA) oder die amerikanische Gesundheitsversicherung. An das Projekt wurden politisch hohe Erwartungen geknüpft, sollte es doch erstmals einen Basisschutz für alle amerikanischen Bürger schaffen. Doch die Webseite mit der Online-Registrierung lieferte nie die erwartete Performance. Kundendaten verschwanden oder mussten wiederholt eingegeben werden. Schließlich funktionierte die Webseite rudimentär, regte aber politische Gegner an, das gesamte Programm zu hinterfragen. Die Mehrkosten der IT lagen im Bereich von knapp 500 Mrd. US\$. Die Gründe für das Scheitern waren zu viele Stakeholder, die mitwirkten, sich stark ändernde Anforderungen, eine unzureichende Validierungsstrategie und ein monolithisches System, das ungeeignet für Teillösungen und inkrementelle Änderungen war.

Die wesentlichen Befunde aus Kundenprojekten, bei denen wir zur Unterstützung und Moderation gerufen wurden, sind im Folgenden aufgelistet – als Warnung, aber auch, damit Sie sich selbst prüfen können:

- Unbrauchbare Lastenhefte und Ausschreibungen (z.B. oberflächlich und missverständlich beschriebene Anforderungen)
- Implizite Anforderungen (Beispiel: Kunde erwähnt Funktionen nicht, da sie für ihn selbstverständlich sind, aber der Lieferant weiß das nicht.)
- Fehlende Anforderungen (z.B. schwammige Anforderungen, die zwar nötig sind, aber nicht geklärt werden, da sie teuer werden könnten; unklare Ausrichtung des Projekts: Was wird nicht geliefert?)
- Unsicherheiten und Unklarheiten (Beispiel: Schätzungen und Pläne basieren auf nicht verstandenen Risiken und oberflächlich dokumentierten Anforderungen.)
- Unzureichendes Änderungsmanagement (Beispiel: Kunde meldet sich beim Projektmanager oder Entwickler: »Wir brauchen das und das noch.«)
- Inkonsistente Dokumentation (Beispiel: Testfälle setzen auf einer anderen Basis auf als die Entwicklung.)
- Varianz und Komplexität (z.B. Mehrfachentwicklungen, die in der Codebasis später inkonsistent werden und einzeln verfolgt werden müssen)
- Fehlendes Wissensmanagement (Beispiel: Projektmanager übernimmt neues Kundenprojekt und hat nicht das implizite Wissen zum Kunden und dessen Hintergrund.)

Technologische Herausforderungen lassen sich beherrschen. Schlechtes Management nicht. In Krisenzeiten, wie 2001 und 2008, geht die Erfolgsquote zurück. Dann werden vermeintlich unnötige Ausgaben, wie für Requirements Engineering, reduziert – mit durchschlagenden Konsequenzen.

Erfolg ist machbar. Hier die wichtigsten Erfolgsfaktoren aus der Industriepraxis:

- Ergebnisorientierte Vorgaben
- Zielorientierte Prozesse
- Kompetentes Produkt- und Projektmanagement
- Standardisierte und optimierte Infrastruktur
- Agile Entwicklung

Wir wollen in diesem Buch darauf eingehen, welche Techniken des RE Sie einsetzen sollten, um mit Ihren Projekten und Produkten zu den Gewinnern zu gehören.

Auf was muss man beim RE achten? Aus unterschiedlichen Praxiserfahrungen lassen sich die wichtigsten Risiken im RE ableiten [Ebert2014a]. Die Risiken zu kennen, heißt, dass man sich darauf vorbereiten kann, um sie beim nächsten Mal zu vermeiden. Die folgende Liste hatte ich ursprünglich mit weiteren sehr

erfahrenen RE-Praktikern erstellt [Lawrence2001]. Sie ist bis heute gültig und nur inhaltlich etwas aktualisiert.

Risiko 1: Fehlende Anforderungen

Häufig werden bestimmte Anforderungen übersehen, und es werden nur greifbare und nachvollziehbare Funktionen spezifiziert. Aber Anforderungen haben verschiedene Ausprägungen, wie wir gesehen haben. Neben den funktionalen Anforderungen gibt es Qualitätsanforderungen und Randbedingungen. Neben den Produkthanforderungen gibt es auch Marktanforderungen und Komponentenanforderungen. Nur die Hinterfragung aller dieser Typen macht die Anforderungsdokumentation vollständig. Wichtig wird diese Vollständigkeit vor allem auch bei der Testspezifikation. Testfälle müssen alle diese Kategorien von Anforderungen abdecken.

Aus vertraglichen Gründen sollte man dem Kunden das liefern, was er will, und nicht das, was er braucht. Interpretieren Sie also nicht, was denn »passen könnte«, denn Sie kennen die Welt des Kunden und seine konkreten Bedürfnisse niemals so gut wie er selbst. Im Zweifelsfall zählt, was vertraglich abgestimmt wurde. Das ist vor allem dort wichtig, wo verschiedene Stakeholder auf Kunden-seite mitwirken und wo wir Anforderungen priorisieren. Ein Lieferant sollte im Interesse einer nachhaltigen Kundenbeziehung im Vorfeld klären, was der Kunde wirklich braucht, um dann vor Projektbeginn eine Abstimmung zu erreichen zwischen dem, was gebraucht wird, und dem, was gewünscht und damit vertraglich festgehalten wird. Eine wirksame Basis für erfolgreiches Kundenmanagement ist es, zuallererst den Business Case des Kunden zu verstehen. Dabei geht es darum, zu erkennen, was der Kunde – anders – machen will, wenn er erst einmal das gewünschte Produkt in den Händen hält. Den Business Case zu verstehen bedeutet, dass man als Produkt- oder Projektmanager erkennt, welche Funktionen oder Anforderungen an das Projekt den größten Nutzen bringen.

Risiko 2: Falsche Anforderungen

Anforderungen sind grundsätzlich unvollständig und fehlerhaft. Sie sind unvollständig, da wir das System nicht in jedem Detail vorab spezifizieren können – und dies auch niemand bezahlen wollte. Sie sind fehlerhaft, weil bei jeder Arbeit Fehler entstehen.

Wir Menschen machen pro zehn Zeilen Text ungefähr einen inhaltlichen Fehler, den wir nicht sofort entdecken. Die Hälfte dieser Fehler entdecken wir bei einer Schlussdurchsicht – sofern wir uns die Zeit dafür nehmen. Die andere Hälfte bleibt im Dokument und muss durch zusätzliche Techniken aufgedeckt und behoben werden. Das ist gerade bei Anforderungen kritisch, denn viele Fehler werden erst spät bei Test und Abnahme des Produkts entdeckt, und dann sind Korrekturen aufwendig.

Typische Fehler sind sowohl handwerklicher Natur, wie vage und ungenaue Beschreibungen, Widersprüche, Inkonsistenzen, Lücken, als auch inhaltlicher Natur, wie Denkfehler, falsche Priorisierung, falsche Abstraktionen und Vermischung von Was und Wie.

Fehler entstehen durch nicht beherrschte Komplexität. Kunden, der Vertrieb und das Marketing spezifizieren Funktionen, die niemand braucht [Pendo2019]. Entwickler versuchen, Anforderungen, die sie vermeintlich verstanden haben, mit Leben zu füllen, und entwickeln so Funktionen, die nie vereinbart wurden. Branchenübergreifend ist ungefähr die Hälfte der gelieferten Funktionalität ohne Wert und wird dementsprechend kaum oder nie verwendet (Abb. 1–4). Das gilt für ein Fahrzeug genauso wie für eine Office-Software. Jede unnötige Funktion bringt Abhängigkeiten von anderen Funktionen, Sonderfälle, Ausnahmesituationen – und damit zusätzliche Entwicklungs-, Korrektur- und Testaufwände, die sich über die Lebensdauer mit jedem Release vervielfachen.

Prüfen Sie Anforderungen mit Reviews (siehe dazu auch Kap. 6). Nutzen Sie dazu Checklisten und Szenarien wichtiger Abläufe. Spielen Sie vor allem die Szenarien durch, mit denen Ihr Kunde Geld verdient oder die dem Benutzer später Schwierigkeiten machen, wenn sie nicht optimal funktionieren. Prüfen Sie, ob Abhängigkeiten oder Fehlerszenarien übersehen wurden. Wer macht diese Reviews? Im Bestfall ein Tester. Tester haben einen Blick für Fehlermöglichkeiten und entdecken in Reviews sehr viel mehr Fehler als Designer oder Projektmanager. Achten Sie auch darauf, dass die Anforderungen kundenseitig geprüft und formal freigegeben wurden.

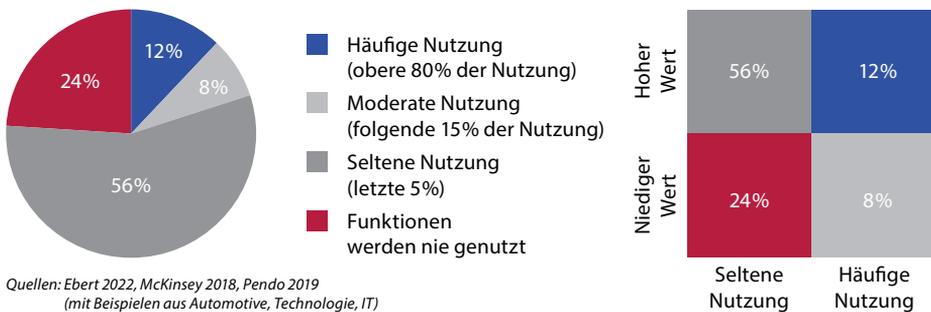


Abb. 1–4 Klasse statt Masse: Häufigkeit der Nutzung von Requirements

Risiko 3: Sich ändernde Anforderungen

Kontrollieren Sie Änderungen. Anforderungen, deren Änderungen nicht beherrscht werden, führen zu Kosten- und Terminüberschreitungen und reduzieren die Qualität. Anforderungen ändern sich in beinahe jedem Projekt. Die Änderungsrate hängt von verschiedenen Faktoren ab, beispielsweise vom Neuigkeitsgrad von Produkt und Technologie. Häufig existiert eine gewisse Basis von Anforderungen, mit denen ein

Projekt gestartet wird. Einige Punkte sind noch offen und klären sich im Laufe der Zeit. Auftraggeber haben allerdings oftmals kein großes Interesse daran, diese Punkte zu klären. Erstens ist es Zusatzaufwand und zweitens könnte der Auftraggeber bei der Abnahme davon profitieren, wenn nicht alles so läuft wie abgesprochen, denn das ist die Chance, komplett neue Anforderungen als Kompensation für diesen Projektfehler kostenlos zu erhalten.

Kontrollieren Sie die Änderungsrate im Projekt. Zu bestimmten Meilensteinen muss die Änderungsmenge reduziert werden, um die nächste Phase erfolgreich durchlaufen zu können. Üblich ist es, mit einem Puffer zu arbeiten, der sowohl Schätzungenauigkeiten als auch Anforderungsänderungen abfangen kann. Eine weitere Maßnahme ist die sogenannte Rückwärtsplanung von der Übergabe aus zurück ins Projekt, um zu erkennen, ab wann der kritische Pfad keine Parallelarbeit als Puffer mehr zulässt. Als Regel gilt, dass in der zweiten Projekthälfte nur noch sehr wichtige Änderungen ohne große Auswirkungen zugelassen werden. Eine dritte Maßnahme ist schließlich, Änderungen grundsätzlich nur zu diskutieren, wenn eine Analyse der Auswirkungen stattgefunden hat. Andernfalls verschwenden beide Seiten ihre Zeit. In diesem »Spiel« wird gern gepokert, nur um zu sehen, wie sich der Lieferant verhält. Oftmals genügt der Verweis auf die Einflüsse im Projektplan, um zu zeigen, dass die vorgeschlagene Änderung Kosten und Projektdauer unzulässig erhöhen wird.

Klären Sie frühzeitig und verbindlich Verantwortungen und Rollen der Kunden bei der Projektarbeit. Unzureichende Einbeziehung der Benutzer schafft viele Risiken. Oft werden Anforderungen interpretiert, ohne den Kunden direkt einzubeziehen. So wachsen die Divergenzen an, statt aufgelöst zu werden. Daraus hat sich das agile Prinzip »Kunde an Bord« entwickelt. Das kann jedoch auch schwierig werden, wenn kundenseitig nicht abgestimmte Anforderungen als Versuchsballons lanciert werden. Und die haben die Tendenz zu platzen. Viele Kunden haben sich nie zu Verantwortungen Gedanken gemacht. Abgestimmte Inhalte werden von uninformierten Kundenvertretern ständig neu hinterfragt und geändert.

Projekte scheitern wegen unzureichender Anforderungen. Abbildung 1–5 zeigt diesen Effekt, wie wir ihn bei einem Kunden beobachtet haben. Unzureichende Anforderungsqualität brachte dort nicht nur das Projekt in Schieflage, sondern reduzierte auch die Motivation dramatisch, denn viele wussten nicht mehr, wie sie die sich ständig ändernden Vorgaben meistern sollten.

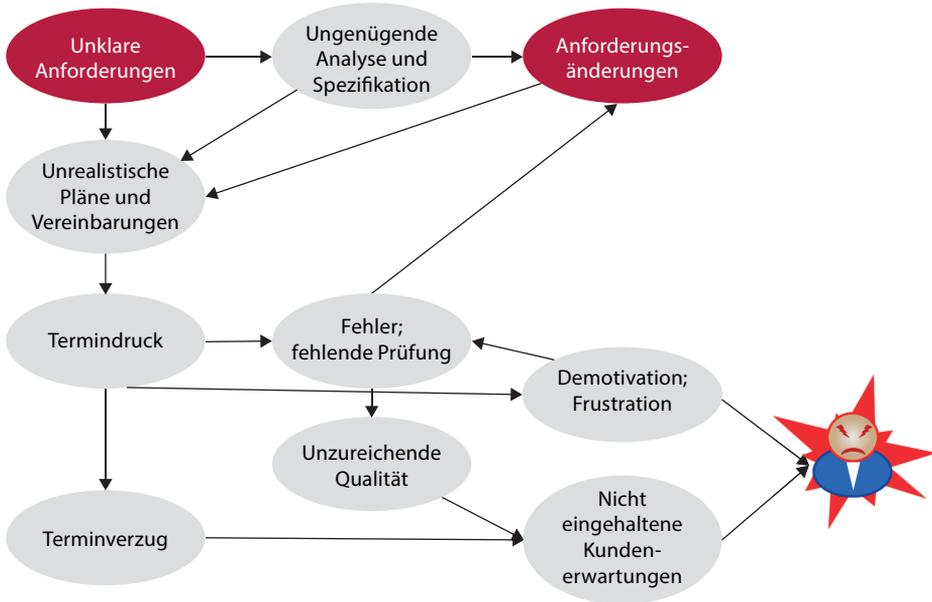


Abb. 1-5 Unzureichende Anforderungen und die Konsequenzen

Diese drei Risiken sind nicht softwarespezifisch und unterstreichen die breite Anwendbarkeit des Requirements Engineering – egal, ob IT, Medizin oder Maschinenbau. Branchenübergreifend ähneln sich die Herausforderungen: Anforderungen fehlen, sind falsch und ändern sich während des Projekts.



Beispiel:

Die Qualität der Anforderungen ist ein wesentlicher Erfolgsfaktor beim Projekterfolg. Eines der ersten kommerziellen Düsenflugzeuge, die Comet 1, hatte keine Anforderungen zu dynamischen Spannungen der Fenster – und stürzte sehr häufig ab. Die Tacoma-Narrows-Brücke hatte unzureichende Anforderungen und Lösungsmodelle bei der Berücksichtigung der Windlast – und stürzte ein. Beim Therac-25-Bestrahlungsgerät war die Benutzerschnittstelle fehlerhaft spezifiziert – und verursachte mehrere Todesfälle. Beim Bau der Ariane-5-Rakete wurden Anforderungen außerhalb des erlaubten Kontexts von Ariane 4 wiederverwendet, und die Rakete stürzte auf dem Jungfernflug ab. Die Liste könnte beliebig verlängert werden. Sie selbst haben bestimmt eigene Beispiele unzureichender Anforderungen. Achten Sie auf hinreichend gute Anforderungen!

1.3 Wirtschaftlicher Nutzen und Return on Investment (ROI)

Die systematische Umsetzung von RE erfordert Aufwand in der Entwicklung und an den Schnittstellen im Produktmanagement, Produktmarketing und Vertrieb. Häufig wird dieser Aufwand als zu hoch und zu zeitraubend angesehen. Aus unserer Beratungspraxis kennen wir das Dilemma: Verbesserungen in Methodik, Ausbildung und Werkzeugen werden nicht angegangen, da der Anfangsaufwand, um diese Verbesserungen anzustoßen, als zu hoch betrachtet wird.

Systematisches RE hat einen klaren Nutzen und ROI, den wir hier zeigen. Einige dieser Faktoren, wie Termintreue oder weniger Nacharbeiten, schaffen einen unmittelbar greifbaren Nutzen. Andere, wie beispielsweise die Kundenzufriedenheit, werden in Form nachhaltig guter Kundenbeziehungen und weiterer Projekte sichtbar. Die folgenden Daten stammen aus der Benchmark-Datenbank von Vector sowie verschiedenen Metastudien zum Effekt von RE [Hussain2016, Standish2021, Gartner2022, Ebert2007, Standish2003, Terzakis2013]:

■ Wertorientierung

50 % aller Funktionen werden nie verwendet. Diese Zahl gilt branchenübergreifend als Richtwert. Wenn einige dieser sowieso eher unnötigen Funktionen frühzeitig erkannt und nicht entwickelt werden, reduziert das die Komplexität und Kosten. RE richtet den Fokus auf das Wesentliche, sorgt für eine klare Positionierung des Produkts und seiner Alleinstellungsmerkmale und schafft damit auch die Gewissheit, dass alle Stakeholder am gleichen Strang ziehen.

■ Qualität

80 % der Fehler im Test und 43 % der Feldfehler resultieren aus unzureichendem RE. Diese Fehlerkosten werden durch ein besseres RE direkt reduziert – der Umfang hängt natürlich davon ab, wie Sie die Schwerpunkte setzen.

■ Kostenreduzierung

Typischerweise werden 2–5 % des Aufwands in das RE investiert. Eine Verdopplung reduziert die Lebenszykluskosten um typischerweise 20–40 %. Die Gründe dafür sind frühe Fehlerentdeckung, frühe Korrektur unzureichender Anforderungen, Fokus auf Erweiterbarkeit etc. Werden die Anforderungen so umgesetzt, wie sie vom Kunden oder Benutzer beschrieben werden, reduziert das die Nacharbeiten, die im Schnitt 45 % des Projektaufwands ausmachen.

■ Produktivitätsverbesserung

Typischerweise werden 30–50 % des Entwicklungsaufwands für die Fehlerbehebung und nicht entwicklungsbezogene Aktivitäten eingesetzt. Die Hälfte der Fehler ist das direkte Ergebnis unzureichender Anforderungen und unkontrollierter Änderungen. Im Systemtest sind es 80 % der Fehler, die aus unvollständigen (31 %) oder falschen (49 %) Anforderungen resultieren. Die Wiederverwendung von Anforderungen und davon abgeleiteten Arbeitsergebnissen (z. B. Mechanismen zur Systemsicherheit) schafft weitere Produktivitätsgewinne, insbesondere bei Produktvarianten.

■ Kürzere Durchlaufzeiten

Verbesserte Projektplanung und Ressourceneinteilung, weniger Verzögerungen vor dem Projektstart, eine schnellere Anlaufphase sowie Termintreue aufgrund von bekannten Anforderungen und klaren Verantwortungen im Projektteam und im Vertrieb reduzieren Wartezeiten sowie Terminverzug. Mehr Aufwand für die Entwicklung und die konsequente Umsetzung und das konsequente Testen der Anforderungen schaffen eine Verbesserung der Termintreue und reduzieren Verzögerungen auf unter 20 %.

Insgesamt zeigen unsere Erfahrungen, dass eine Verdoppelung des Aufwands für RE hin zu 10 % des Projektaufwands in den Bereichen Methodik, Prozesse, Training und Werkzeugunterstützung einen konkret realisierbaren Projektnutzen von über 20 % schafft (Abb. 1–6). Das ist ein ROI von mehr als 4, und bei diesem Wert sind nur die direkt messbaren Vorteile berücksichtigt.

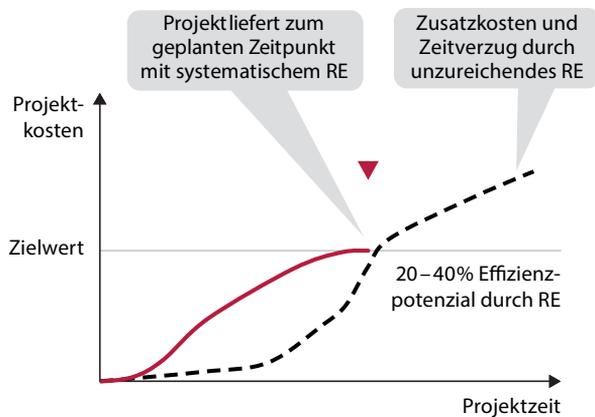


Abb. 1–6 Effizienzpotenzial mit systematischem Requirements Engineering

Die umfassendste Untersuchung zum Nutzen von RE stammt von Alcatel (heute Nokia). Über mehrere Jahre hinweg wurden in einer longitudinalen Feldstudie unterschiedliche Projektdaten systematisch erfasst und analysiert (Abb. 1–7) [Ebert 2006].

Gute Termintreue wird nur dann erreicht, wenn die vier folgenden Techniken gleichzeitig umgesetzt werden. Wurde nur einer dieser Faktoren vernachlässigt, führte das sofort zu Terminverzug:

- Ein verantwortliches Kernteam, bestehend aus Produktmanagement, Marketing, Entwicklung und Produktion, das das gesamte Projekt (oder das Produktrelease) steuert
- Konsequente Nutzung eines definierten Lebenszyklus mit Meilensteinen, Checklisten etc.
- Transparenz aller Projektvereinbarungen (z.B. Anforderungen) im Intranet

- Gemeinsame Anforderungsanalyse durch das Kernteam mit Produktmanagement, Marketing, Entwicklung und Produktion

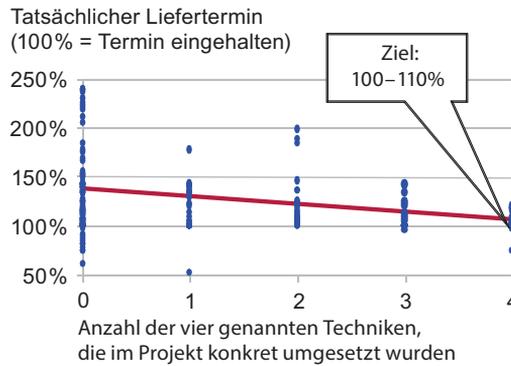


Abb. 1-7 Der gleichzeitige Einsatz von vier Requirements-Techniken verbessert den Projekterfolg.

Eine weitere umfassende Studie zum Nutzen von RE in Entwicklungsprojekten kommt von der NASA (Abb. 1-8) [Hooks2000]. Im Unterschied zu den beiden vorigen Studien wurde hier die Kosteneinhaltung in Abhängigkeit vom Aufwand für RE untersucht. Projekte mit 5% Aufwand für RE führen zu einer Kostenüberschreitung von 80% bis knapp 200%. Wird dieser Aufwand in Richtung 8-14% verdoppelt, liegt die Kostenüberschreitung bei unter 60%. Offensichtlich sind IT-Projekte sehr anfällig für eine unzureichende Anforderungsanalyse und -spezifikation, denn die Anforderungen werden sich im Projektverlauf zunehmend ändern und zu beträchtlichen Zusatzaufwänden führen. Auch hier gilt, dass die absoluten Zahlen für Überschreitungen von Kosten natürlich durch viele Faktoren bestimmt werden. Aber ein unzureichendes RE hat einen starken Anteil an überbordenden Kosten. Intel hat in einer aktuellen Langzeitstudie gezeigt, dass systematisches RE die Zahl der Fehler im Produkt um 30-50% senkt [Terzakis2013].

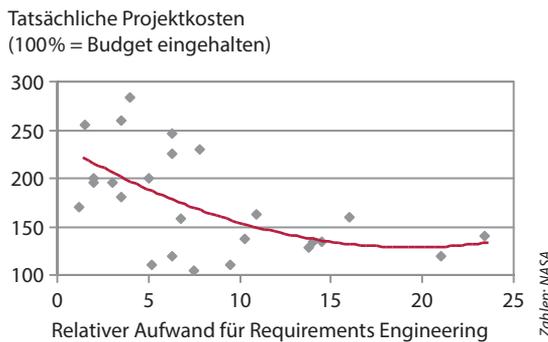


Abb. 1-8 Kosteneinhaltung in Abhängigkeit vom Aufwand für das Requirements Engineering

Viele Projekte geraten in Schwierigkeiten, da anfangs zu wenig Energie investiert wird und später in der »heißen« Phase die Zeit fehlt, um nachzudenken. Aufwendige Nacharbeit ist nun nötig, um Fehler und Entscheidungen zu korrigieren, die in der Startphase des Projekts viel leichter aufzulösen gewesen wären. Andere müssen Kapazität abgeben, um die jetzt dringend nötige Überlast abzudecken. Dieser Teufelskreis lässt sich nur durch frühzeitige Planung und kontinuierliches Requirements Engineering durchbrechen. Das wird als »Frontloading« bezeichnet und bedeutet, dass Projektaufgaben frühestmöglich erledigt werden, um Nacharbeiten und damit Terminverzug zu vermeiden. Abbildung 1–9 zeigt in der oberen Hälfte ein typisches Projekt. Anfangs wird zu wenig Energie in Requirements Engineering und Planung investiert.

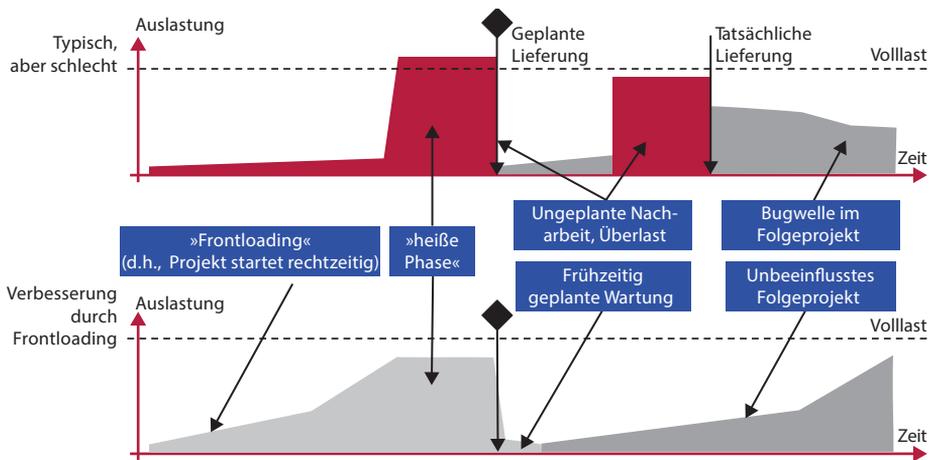


Abb. 1–9 Frühzeitige Lastbalance durch Frontloading

Daraus resultieren Nacharbeiten und die Kannibalisierung nachfolgender Projekte. Zudem führt die kontinuierliche Überlast in der »heißen Phase« zu Demotivation. Die untere Hälfte in der Abbildung zeigt den Effekt des »Frontloading« mit systematischem Requirements Engineering. Anforderungen werden frühzeitig ermittelt und das Projekt wird realistisch geplant. Aufwände werden über einen größeren Zeitraum verteilt und erlauben ein agiles Arbeiten mit inkrementellen Lieferungen. Auswirkungen auf Folgeprojekte werden vermieden.

1.4 Wie Sie von diesem Buch profitieren

Dies ist ein Buch für Einsteiger und Profis. Nach der Lektüre

- haben Sie einen Überblick über Theorie und Praxis des Requirements Engineering;
- verstehen Sie, wie moderne Werkzeuge und Notationen Sie beim Requirements Engineering praktisch unterstützen können;

- können Sie die wichtigen Elemente des Requirements Engineering in Ihren Projektalltag übertragen und dort produktiv einsetzen.

Abbildung 1–10 zeigt die Struktur des Buches. Das Thema RE wird zunächst anhand verschiedener Übersichtskapitel eingeführt. Kapitel 2 führt kurz und knapp in das Requirements Engineering ein und zeigt den Nutzen in der Praxis, aber auch die Risiken, wenn es nicht ausreichend gelebt wird. Die Kapitel 3 bis 8 beleuchten die einzelnen Aktivitäten innerhalb des RE systematisch. Kapitel 3 beschreibt, wie Anforderungen ermittelt werden. Der Nutzen und Wert aus der Sicht desjenigen, der bezahlt, steht im Vordergrund, denn das ist die wesentliche Basis für jedes erfolgreiche Projekt. Kapitel 4 betrachtet die Dokumentation, also das Beschreiben von Anforderungen. Dabei geht es um die Verbesserung der Anforderungsqualität und verschiedene formale Arten der Beschreibung, die hinsichtlich ihrer Praktikabilität und Schwierigkeit in der Umsetzung diskutiert werden.

Die relevanten Analyse- und Modellierungstechniken werden in Kapitel 5 eingeführt. Ein durchgängiges Beispiel erlaubt eine gute Vergleichbarkeit der Techniken und ihres Nutzens. Kapitel 6 zeigt, wie qualitativ gute Anforderungen erreicht werden. Wir zeigen hier Techniken zu Reviews, Prüfungen und konkrete Checklisten, um die Anforderungsqualität zu verbessern. Kapitel 7 unterstreicht die Abstimmung von Anforderungen mit den verschiedenen beteiligten Gruppen. Das Kapitel betrachtet auch rechtliche Zusammenhänge, beispielsweise Gewährleistungs- und Haftungsfragen. Schließlich beschreibt Kapitel 8 Techniken, um Anforderungen im Projekt zu kontrollieren und zu verwalten.

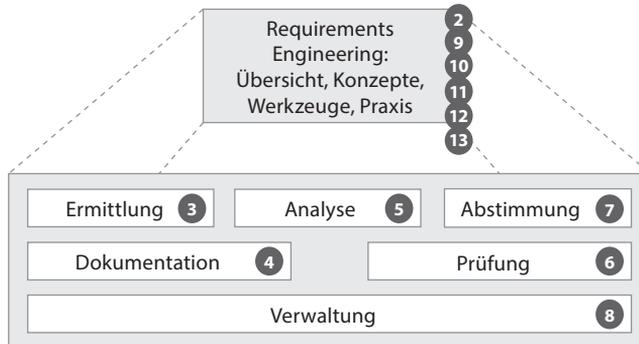


Abb. 1–10 Das Buch im Kontext des Requirements Engineering (Schwarze Kreise sind Kapitelnummern.)

Kapitel 9 ist komplett neu und beschreibt agiles Requirements Engineering. Wir gehen auf Methoden wie Design Thinking, Planning Poker und testorientiertes RE ein. Werkzeuge werden in Kapitel 10 charakterisiert und bewertet. Wir beschreiben einige populäre RE-Werkzeuge ganz praxisnah in unterschiedlichen Szenarien. Die Praxis des RE wird in Kapitel 11 dargestellt. Damit sehen Sie den praktischen Nutzen und die Abhängigkeiten der einzelnen Schritte im RE. In Kapitel 12 möchte ich Ihnen zeigen, wie Sie Ihre eigene Kompetenz ausbauen können. Das beinhaltet