

peter HRUSCHKA



2. Auflage

BUSINESS ANALYSIS

REQUIREMENTS ENGINEERING

Produkte und Prozesse
nachhaltig verbessern

HANSER



Ihr Weg zur Requirements-
Zertifizierung

Hruschka

Business Analysis und Requirements Engineering

Bleiben Sie auf dem Laufenden!



Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter



www.hanser-fachbuch.de/newsletter

Peter Hruschka

Business Analysis und Requirements Engineering

Produkte und Prozesse
nachhaltig verbessern

2., aktualisierte Auflage

HANSER

Der Autor:

Dr. Peter Hruschka, Aachen

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.



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.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2019 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Brigitte Bauer-Schiewek

Copy editing: Petra Kienle, Fürstenfeldbruck

Umschlagdesign: Marc Müller-Bremer, www.rebranding.de, München

Umschlagrealisation: Stephan Rönigk

Layout: Manuela Treindl, Fürth

Gesamtherstellung: Kösel, Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

Print-ISBN: 978-3-446-45589-4

E-Book-ISBN: 978-3-446-45734-8

Inhalt

Vorwort	XI
Vorwort zur zweiten Auflage.	XII
1 Probleme, Ziele, Ideen und Visionen	1
1.1 Wovon sprechen wir?	1
1.2 Quantitative Gründe	2
1.3 Qualitative Gründe	3
1.4 Warum macht es nicht jeder richtig?	4
1.5 Standardisierung und Zertifizierung	5
1.6 Drei Säulen erfolgreicher Projekte	6
1.7 Definition: Business Analysis und Requirements Engineering.	7
1.8 Definition: Requirement.	11
1.9 Arten von Anforderungen	12
1.10 Vier Hauptaufgaben eines Analytikers.	14
1.11 Benötigte Fähigkeiten.	16
1.12 Aufgabenverteilung im Team.	17
1.13 Der Aufwand für die Analyse.	20
1.14 Was erleichtert die Analyse?	22
1.15 Verschiedene Vorgehensweisen.	24
1.16 Zusammenfassung	27
2 Erfolgreich starten	29
2.1 Drei Zutaten zu einem erfolgreichen Projektstart.	29
2.2 Ziele	30
2.3 Ziele spezifizieren.	32
2.4 Stakeholder	34
2.5 Stakeholder finden	36
2.6 Die wichtigsten Stakeholder: die Nutzer.	39
2.7 Weitere Quellen für Anforderungen	41
2.8 Scope und Kontext	42
2.9 Scope und Analytiker	46
2.10 Umgang mit Grauzonen	49
2.11 Darstellung der System-/Produktgrenze	50
2.12 Alternative Notationen	56

2.13	Die drei Erfolgszutaten (nochmals)	58
2.14	Zusammenfassung	60
3	Geschäftsprozesse und Produktfunktionalität	61
3.1	Anforderungen unterschiedlicher Granularität	61
3.2	Funktionale Anforderungen gliedern und strukturieren	63
3.3	Prozesse: die Grundidee	65
3.4	Prozesse finden	68
3.5	Empfehlungen und Warnungen	72
3.6	Zusammenfassung	74
4	Funktionen genauer betrachtet	75
4.1	Zerlegungskriterien	75
4.2	Wo hört man auf?	78
4.3	Top-down oder bottom-up?	79
4.4	Zusammenfassung	82
	Intermezzo	83
5	Anforderungen in Umgangssprache	85
5.1	IEEE-Forderungen an Anforderungen	85
5.2	Zwischen Wahrnehmung und Niederschrift	87
5.3	Gute umgangssprachliche Anforderungen	90
5.3.1	User Storys	90
5.3.2	Alternative Satzschablonen	93
5.4	Generelle Stilregeln	97
5.5	Ein Glossar für die Daten	100
5.6	Gute Definitionen	101
5.7	Vorgehensweise bei Glossareinträgen	102
5.8	Zusammenfassung	104
6	Anforderungen modellieren	107
6.1	Use-Case-Modelle	108
6.1.1	Use Cases strukturieren	115
6.1.2	Use Cases und natürliche Sprache: ein Vergleich	117
6.1.3	Business Use Cases und Product Use Cases	119
6.1.4	Use Cases finden	120
6.1.5	Die Anzahl von Use Cases	121
6.1.6	Drei Tricks zur Vereinfachung	123
6.1.7	Use Cases beschreiben	126
6.1.8	Beschreibung auf Drachenniveau	128
6.1.9	Beschreibung auf Wellenniveau	129
6.1.10	Beschreibung auf Fischniveau	131
6.1.11	Der Stil auf Wellenniveau	132

6.1.12	Zusammenfassung Use-Case-Modelle	135
6.2	Datenmodelle	135
6.2.1	Eine kleine Geschichte	136
6.2.2	Datenmodelle als strukturiertes Glossar	138
6.2.3	(Entity-)Klassen	141
6.2.4	Entity-Klassen-Modelle	145
6.2.5	Beziehungen	146
6.2.6	Spezielle Beziehungen	152
6.2.7	Malen oder schreiben?	154
6.2.8	Noch drei Beispiele	156
6.2.9	Abläufe und Daten	160
6.2.10	Ein Ausblick auf die Erstellung von Datenmodellen	162
6.2.11	Zusammenfassung Datenmodelle	168
6.3	Wenn die Grobspezifikation von Prozessen nicht ausreicht	169
6.4	Aktivitätsdiagramme	171
6.4.1	Aktivitäten zerlegen	175
6.4.2	Swimlanes und Daten	177
6.4.3	Malen oder schreiben?	179
6.4.4	Wo hört man auf?	181
6.4.5	Nochmals: top-down oder bottom-up?	184
6.5	Alternative Funktionsmodelle	185
6.5.1	Datenflussdiagramme	185
6.5.2	Ereignisgesteuerte Prozessketten (EPK)	187
6.5.3	Business Process Model and Notation (BPMN)	188
6.5.4	Zusammenfassung feinerer Funktionsmodelle	189
6.6	Verhaltensmodelle	189
6.6.1	Warum noch ein Modell?	189
6.6.2	Grundlagen von Zustandsmodellen	191
6.6.3	Aktionen und Aktivitäten	195
6.6.4	Zustandsmodelle erstellen und prüfen	198
6.6.5	Komplexe Zustandsmodelle	200
6.6.6	Ein Beispiel	204
6.6.7	Malen oder schreiben?	207
6.6.8	Zustandsmodelle und Aktivitätsdiagramme	208
6.6.9	Use Cases und Zustandsmodelle	211
6.6.10	Zusammenfassung Zustandsmodelle	214
6.7	Zusammenfassung Requirements-Modelle	214
7	Qualitätseigenschaften und Randbedingungen	217
7.1	Was sind nichtfunktionale Anforderungen?	217
7.2	Kategorien nichtfunktionaler Anforderungen	221
7.3	Nichtfunktionale Anforderungen finden und zuordnen	225
7.4	Beispiele für äußere Qualitäten	228
7.5	Beispiele für innere Qualitäten	237

7.6	Beispiele für Randbedingungen	238
7.7	Messbarkeit von Anforderungen	242
7.8	Zusammenfassung	244
8	Anforderungsdokumente	245
8.1	Warum überhaupt Dokumente?	245
8.2	Viele Namen und mehrere Dokumente?	247
8.3	Anforderungen an Requirements-Dokumente	249
8.4	Beispiele für die Struktur von Requirements-Dokumenten	250
8.5	Mindestinhalte	257
8.6	Zusammenfassung	258
9	Anforderungen ermitteln	259
9.1	Das Kano-Modell	259
9.2	Arten von Erhebungsmethoden	263
9.3	Was beeinflusst die Auswahl?	264
9.4	Beispiele für Frage-Antwort-Techniken	266
9.5	Beispiele für Beobachtungstechniken	271
9.6	Beispiele für vergangenheitsorientierte Techniken	272
9.7	Beispiele für Kreativitätstechniken	274
9.8	Erhebungstechniken und Hilfsmittel	275
9.9	Noch eine Kreativitätstechnik	281
9.10	Überblick (Reprise)	283
9.11	Zusammenfassung	283
10	Anforderungen prüfen und abstimmen	285
10.1	Quality Gates	285
10.2	Ziele der Prüfung	288
10.3	Arten der Prüfung	289
10.4	Wer sollte beteiligt sein?	292
10.5	Was wird geprüft?	293
10.6	Checklisten für inhaltliche Prüfungen	295
10.7	Was tun bei Mängeln?	298
10.8	Konfliktmanagement	299
10.9	Zusammenfassung	302
11	Requirements-Management	303
11.1	Definition: Requirements-Management	303
11.2	Vorbereitende Tätigkeiten	306
11.3	Der Requirements-Prozess	307
11.4	Rollen	310
11.5	Laufende Tätigkeiten	312
11.6	Attributierung von Requirements	313
11.7	Sichtenbildung	318

11.8	Priorisierung	319
11.9	Baselines und Releases	322
11.10	Change Management	324
11.11	Traceability	327
11.12	Zusammenfassung	331
12	Requirements-Werkzeuge	333
12.1	Kategorien von Werkzeugen	333
12.2	Leistungen von Werkzeugen	334
12.3	Stärken und Schwächen der Kategorien	336
12.4	Werkzeugauswahl	337
12.5	Einführung von Werkzeugen	338
12.6	Zusammenfassung	339
Literatur		341
Stichwortverzeichnis		343

Vorwort

Als ich 1976 direkt nach dem Studium meinen ersten Arbeitsvertrag in Händen hielt, war ich überrascht. Als Berufsbezeichnung war „Systemanalytiker“ eingetragen. Ich war deshalb überrascht, weil ich „Programmierer“ erwartet hatte. Denn was wir an der Technischen Universität Wien im Informatikstudium gelernt hatten, war Programmieren, formale Sprachen, Mathematik und Logik, Hardware- und Elektrotechnikgrundlagen und vieles andere, aber keine Systemanalyse. Die Berufsbezeichnung „Systemanalytiker“ klang jedoch in meinen Ohren gut und ich fühlte mich aufgewertet. Ich habe mir jedoch vorgenommen, im Lauf meines Berufslebens herauszubekommen, was ein Systemanalytiker so tut.

In der Zwischenzeit haben wir die Berufsbezeichnung „Systemanalytiker“ oft durch „Business Analyst“ oder durch „Requirements Engineer“ ersetzt. Im Kern geht es immer noch um das gleiche Thema: herauszubekommen, wo unsere heutigen Produkte und Prozesse Schwachstellen haben, was Kunden und Nutzer wirklich brauchen, wie Prozesse durch neue Ideen effektiver gemacht werden können oder was IT-Systeme wirklich leisten sollten, um das Business besser zu unterstützen. Und das, was wir herausbekommen haben, wollen wir effektiv miteinander besprechen und verhandeln können, bevor wir es in Lösungen umsetzen (lassen), Produkte erstellen oder IT-Systeme bauen.

Jedem Vorhaben oder Projekt, d. h. jeder Korrektur, Verbesserung, Erweiterung, Änderung oder Diversifizierung Ihrer Produkte oder Ihrer betrieblichen Prozesse sollte ein klares Verständnis des Ist-Zustands, eine gründliche Analyse von Schwachstellen und Risiken und die Identifizierung von Verbesserungsoptionen vorausgehen.

Das ist das Thema dieses Buchs. Sie können es gerne Business Analysis, Requirements Engineering oder auch Systemanalyse nennen. Das sind nur verschiedene „Jahrgänge“ von Worten, die das gleiche Thema behandeln: analysieren und artikulieren, welche Anforderungen wir haben, um Produkte und Prozesse innovativ und nachhaltig zu verbessern.

Wie Sie alle bin auch ich beruflich und privat Nutzer von sehr vielen Softwaresystemen und Produkten. Das beginnt beim Mobiltelefon, bei den Programmen auf meinem Rechner, mit denen ich Kursfolien gestalte oder Bücher und Artikel schreibe; die Systeme, die die Abrechnungen erstellen, die ich jeden Monat erhalte und bezahlen muss, aber auch viele andere technische Systeme im Auto, mit denen ich täglich unterwegs bin, oder in meinem Kaffeevollautomat, um mich mit verschiedenen Spezialitäten bei Laune zu halten. Mit diesem Buch möchte ich ganz einfach meinen Beitrag dazu leisten, dass diese Systeme und Produkte den Benutzern mehr Freude als Ärger bereiten.

■ Vorwort zur zweiten Auflage

Vier Jahre sind vergangen seit der ersten Auflage. Business Analysis und Requirements Engineering hat sich durch die Verbreitung agiler Ansätze geändert – und doch auch nicht. Einiges hat neue Namen erhalten, einiges hat sich an den bevorzugten Notationen geändert und es sind auch einige neue Tools populär geworden, um Anforderungen zu erfassen und zu verwalten. Deshalb habe ich in dieser zweiten Auflage die Begriffe Notationen und Tools mit aufgenommen, über die man heute spricht, wenn man die Rolle Business Analyst, Product Owner oder Requirements Engineer im Rahmen des Unternehmens ausfüllen soll.

Vieles ist aber seit vielen Jahrzehnten einfach immer noch wahr und wichtig. Noch immer wollen wir als Analytiker die Probleme und Geschäftsideen verstehen und kommunizieren, bevor wir an Lösungen gehen. Nur ist unsere Welt immer schnelllebiger geworden: Wir wollen die Lösungen schneller und öfter. Daran müssen sich auch die Analyseprozesse anpassen.

Eine meiner Hauptbotschaften, die sich im Titel „Business Analysis und Requirements Engineering“ ausdrückt, ist immer noch nicht ganz in der täglichen Praxis angekommen: die möglichst reibungslose Zusammenarbeit zwischen Fachabteilungen und IT-Abteilungen, zwischen Auftraggebern und Auftragnehmern, zwischen denen, die das Business verbessern wollen, und denen, die IT-Systeme dafür entwickeln und liefern können.

Aber ich gebe die Hoffnung nicht auf, die Grenze zwischen Firmen oder Abteilungen weiter abzubauen: Denn, egal, auf welcher Seite Sie angesiedelt sind, wir ziehen doch (hoffentlich) am gleichen Strang (und auch am gleichen Ende des Strangs) und wollen, dass IT-Systeme unser Leben immer weiter erleichtern.

Aachen, Februar 2019

Peter Hruschka

1

Probleme, Ziele, Ideen und Visionen

Wenn man ein Projekt aufsetzt, dann hat man entweder im heutigen geschäftlichen Alltag ein Problem oder eine Schwachstelle identifiziert, die man beseitigen möchte, oder man hat eine Idee oder eine Vision, wie ein Produkt oder ein Prozess etwas besser, schneller, effizienter, billiger etc. werden könnte. Nun ja, manchmal hat man vielleicht auch einfach noch Geld übrig und möchte es für irgendetwas ausgeben. Das alles sind Ausgangspunkte für Business Analysis und Requirements Engineering.

Im ersten Teil wollen wir uns mit einigen Grundbegriffen und einer Einführung in das Thema Business Analysis und Requirements Engineering auseinandersetzen. Um Sie zu motivieren, betrachten wir einige quantitative und qualitative Gründe für den Einsatz derartiger Methoden. Wir erwähnen aber auch Ursachen, warum es trotz aller guten Gründe manchmal nicht ernsthaft betrieben wird.

Danach werden wir die drei Säulen erfolgreicher Projekte betrachten, von denen Business Analysis und Requirements Engineering eine ist.

Wir besprechen Definitionen: Was sind überhaupt Requirements oder auf Deutsch, was sind Anforderungen? Und was bedeutet Business Analysis und was Requirements Engineering? Was gehört dazu, was gehört nicht dazu?

Dann diskutieren wir die Berufsbilder Business Analyst, Requirements Engineer und Product Owner. Was haben diese Personen zu tun? Was müssen sie machen? Aber auch: Welche Fähigkeiten sollten sie mitbringen, welche Soft-Skills und welche methodischen Fähigkeiten?

Anschließend besprechen wir die drei wichtigen Arten von Anforderungen. Und zum Schluss des Kapitels betrachten wir verschiedene Vorgehensweisen, wie man unter unterschiedlichen Randbedingungen zu Anforderungen kommen kann – vom Wasserfallmodell bis zu sehr agilen Vorgehensweisen.

■ 1.1 Wovon sprechen wir?

In immer stärkerem Maß unterstützen heute technische Produkte oder IT-Systeme unser Leben. Sie helfen uns, Ziele schneller und effektiver zu erreichen, erleichtern oft vormals manuelle Tätigkeiten oder ermöglichen Dinge, von denen wir vor einiger Zeit nicht einmal zu träumen gewagt haben. Wir nehmen Geldtransaktionen von zuhause aus vor, wir überlassen das Schalten der Gänge und die Traktionskontrolle unseres Autos, wir telefonieren und

surfen mobil fast überall, wir nutzen medizinische Geräte zur Verbesserung der Diagnostik, wir gestalten Kundenprozesse benutzerfreundlicher und effektiver ...

Dazu ist es natürlich notwendig, unsere Probleme, Wünsche, Träume und Visionen so zu durchleuchten und aufzubereiten, dass die IT-Entwickler und Produktentwickler dafür geeignete Lösungen erstellen können oder bestehende Produkte und Prozesse nachhaltig verbessern können.

Ohne klare Kenntnis des Ist-Zustands und ohne klare Ideen, was wir erreichen wollen, verzetteln wir uns sonst in Lösungen, die an den Marktbedürfnissen oder den Bedürfnissen einer Organisation vorbeigehen und mühevoll und aufwendig nachgebessert werden müssen. Deshalb brauchen wir Methoden und Techniken, um herauszubekommen, was das Geschäft wirklich braucht oder was Produkte leisten sollen. Das nennen wir Business Analysis oder Requirements Engineering.

■ 1.2 Quantitative Gründe

Wir können sicherlich sehr viele quantitative und qualitative Argumente anführen, warum wir uns mit dem Thema auseinandersetzen sollen. Sprechen wir zuerst mal über die Manager. Die wollen immer Zahlen hören. Sehr bekannt ist die Studie der Standish-Group über Fehler in Softwareprojekten. Software, die nie fertig wird, die viel zu teuer ist, die zu spät ausgeliefert wird und die weit überteuert ist. Oder sogar total scheitert. In dieser Studie, die jährlich erneuert wird, hat man die Top-10-Risiken, die zehn größten Risiken, in Projekten festgehalten. In Bild 1.1 können Sie sehen, dass sich vier von diesen Top-10-Risiken mit dem Thema unseres Buchs beschäftigen, mit den Anforderungen.

	Executive Support	18
1.	User Involvement	16
	Experienced Project Manager	14
2.	Clear Business Objectives	12
3.	Minimized Scope	10
	Standard Software Infrastructure	8
4.	Firm Basic Requirements	6
	Formal Methodology	6
	Reliable Estimates	5
	Other	5

Bild 1.1

Die Top-10-Risiken in Projekten
gemäß Standish-Studie

Erstens die wirkliche Beteiligung der Benutzer, die das System haben wollen, an dem Prozess (heute eher Stakeholder Involvement genannt – mehr dazu in Abschnitt 2.4); zweitens klare geschäftliche Zielsetzungen – heute oft auch Vision genannt; drittens eine eindeutige Festlegung des Scope und viertens eine vernünftige Formulierung der Anforderungen. Die Zahlen hinten den Punkten sind Gewichtungen. Zusammengenommen machen diese vier Themen 44 von 100 Punkten aus, fast den halben Projekterfolg!

Aber ich glaube, noch deutlicher hat es der Großmeister der Zahlen, Capers Jones, ausgedrückt. Er hat auch untersucht, wie viele Fehler Teams machen, die keine guten Analysemethoden,

keine guten Requirements-Methoden oder andere Qualitätsmethoden für Anforderungen haben [Jon08]. Sein Ergebnis: Teams machen 0,23 Fehler pro Function Point. Sie brauchen jetzt nicht genau zu verstehen, was ein Function Point ist. Vergleichen Sie den Wert nur mit dem Ergebnis beim Einsatz guter Analysemethoden.

Mit guten Requirements-Methoden kann man die Fehlerrate von 0,23 auf 0,08 pro Function Point reduzieren. Das ist ein Drittel davon. Dreimal weniger Fehler. Und sicherlich kennen die meisten von Ihnen die Statistiken, was es uns kostet, Fehler erst im Betrieb eines Produkts oder eines Systems zu korrigieren. Es ist aufwendiger, es ist viel teurer, als wenn wir Fehler zu dem Zeitpunkt finden, wo sie gemacht werden, nämlich beim Beschreiben des Problems. Manche Chefs fangen an zuzuhören, wenn man sagt, dass drei Mal weniger Fehler gemacht werden und drei Mal weniger Kosten hinterher für die Korrektur dieser Fehler aufgewendet werden müssen. Das sind schlagende betriebswirtschaftliche Argumente für den Einsatz guter Analysemethoden.

■ 1.3 Qualitative Gründe

Aber es gibt auch qualitative Argumente, warum wir Business Analysis und Requirements Engineering betreiben sollten.

Der Hauptgrund ist: Falsch verstandene oder nicht korrekte Anforderungen führen zu falschen Systemen. Wenn also die Anforderungen ungenügend klar sind, erhalten wir die falsche Software und Produkte und meistens merken wir das erst im Betrieb. Also, ich brauche gute Anforderungen, um zu guten Systemen zu kommen.

Falsche Anforderungen
→ falsche Systeme

Ein zweiter Punkt, warum das Thema so wichtig ist, sind die vielen impliziten Annahmen. Das ist ohnehin klar, da brauchen wir nicht drüber sprechen, das weiß doch jeder. Naja, scheinbar doch nicht. Denn wenn es nicht vorher klar gesagt wurde, kommen Designer und Programmierer auf ihre eigenen Ideen und entwickeln wieder vorbei an den Bedürfnissen des Markts.

Implizite Anforderungen

Sehr viele Anforderungen sind impliziter Natur und es ist gar nicht leicht, diese aus den entsprechenden Personen herauszulocken. Wenn man es aber nicht tut, bleibt es eine implizite Annahme und jemand trifft Entscheidungen.

Selbst wenn Anforderungen explizit gemacht werden, ist der Projekterfolg noch nicht gewährleistet. Denn viele Anforderungen sind interpretierbar. Der Auftraggeber hält eine Aussage für absolut klar und verständlich; der Auftragnehmer interpretiert die Aussage jedoch ganz anders – und schon wieder haben wir unter Umständen eine falsche Lösung.

Interpretierbare
Anforderungen

Zuletzt wollen wir noch auf ein Problem hinweisen, das uns oft betrifft. Anforderungen ändern sich im laufenden Projekt. Wir verfolgen ein bewegliches Ziel. Saubere Methoden und Vorgehensweisen dieses „Requirements Creep“ sollten Ihnen daher einiges wert sein.

Requirements Creep

■ 1.4 Warum macht es nicht jeder richtig?

Man könnte sich jetzt die Frage stellen, warum es nicht jeder richtig macht, wenn wir doch wissen, dass schlampige Analyse sehr viele Fehler produziert und sehr teuer ist. Ich glaube, der Hauptgrund ist: Wir haben Kommunikationsprobleme miteinander. Es ist nicht leicht, Anforderungen in einer Weise auszudrücken, die alle Beteiligten verstehen. Das wird umso schwieriger, je weltweit verteilter unsere Projekte sind, wenn wir nicht mehr die gleiche Sprache sprechen oder in anderen Kulturen groß geworden sind – dann ist auch das gegenseitige Verständnis nicht so gut.

Das ist aber nur ein Argument. Ein weiteres Argument, warum es nicht jeder richtig macht, ist: Wir wollen einfach Geld sparen. Gute Analyse kostet Zeit und Aufwand. Zeit und Aufwand, den man vielleicht erst ersetzt bekommt, indem man keine Nachbesserungen machen muss. Das ist aber *nach* Abwicklung des Projekts. Mancher Projektleiter ist heilfroh, wenn er seinen Abnahmetest bestanden hat und fertig ist, und nach ihm die Sintflut. Was es dann noch kostet, die Fehler auszubessern, ist unter Umständen nicht mehr in der Verantwortung eines Projektleiters. Und deshalb versucht man, möglichst wenig Analyse zu machen, um möglichst schnell zu Lösungen zu kommen, was aber vielleicht dann teure Nachbesserungen erfordert.

Vielleicht noch ein weiteres Argument: Es geht mitunter nicht nur um das Sparen von Geld, sondern manchmal auch von Zeit. Wir wollen einfach rasche Lösungen und nehmen uns nicht die Zeit, vorher festzulegen, was es sein soll, das wir hier produzieren. Das ist aber ein falsch verstandenes Zeit sparen, denn die Zeit brauchen Sie hinterher mehrfach wieder.

Aus vielen dieser Gründe werden Business Analysis und Requirements Engineering nicht von allen Leuten ernst genommen, aber in den letzten 20 Jahren haben viele Unternehmen erkannt, dass es ein sehr wichtiges Thema ist, haben viel in dieses Thema investiert und es ist deutlich besser geworden.

Natürlich sind auch die Methoden in den letzten 35 Jahren deutlich besser geworden. Als ich angefangen habe zu arbeiten, gegen Ende der 1970er-Jahre, gab es nicht viel an Analysemethodik. In den letzten 35 Jahren ist hier eine Menge dazugekommen an Verständnis, wie man systematisch Geschäftsprozesse untersucht und Anforderungen erheben kann, wie man sie dokumentieren, prüfen und im Lauf der Zeit verwalten kann.

■ 1.5 Standardisierung und Zertifizierung

Business Analyst und Requirements Engineer sind heute keine geschützten Berufsbezeichnungen. Jeder, der möchte, kann sich diese Titel selbst verleihen, kann sich Systemanalytiker nennen oder Business Analyst oder auch Requirements Engineer, ohne dafür einen bestimmten Nachweis zu erbringen.

Zur Verbesserung dieser Situation hat sich 2006 eine Gruppe von Fachleuten zusammengeschlossen, um das IREB zu gründen, das International Requirements Engineering Board (www.ireb.org).

Die Hauptaufgabe dieses Boards war, das Wissen, das man zur Ausübung dieses Berufsbilds benötigt, zusammenzuschreiben und einen Lehrplan zu formulieren, zunächst für Basiskonntnisse, die man unbedingt nachweisen sollte, um den Beruf gut ausüben zu können. Neben diesem Lehrplan über das Basiswissen wurden Fragenkataloge erstellt, mit denen man nachweisen kann, das Wissen wirklich erworben zu haben. Sie können sich dieses Wissen bei unabhängigen Stellen zertifizieren lassen.



Dieses Buch ist IREB-konform. Es bietet alle Grundlagen, die man kennen muss, um die Prüfung zum CPRE (Certified Professional for Requirements Engineering) auch erfolgreich bestehen zu können. Bis 2018 wurden mehr als 50 000 Personen nach diesem Programm weltweit zertifiziert, in den letzten Jahren mehr als 7 500 pro Jahr. Der Verein bietet seit einigen Jahren auch entsprechende Aufbaumodule an, um das Basiswissen zu vertiefen. Das Buch deckt auch große Teile von Advanced Level Requirements Modeling und RE@Agile ab (Bild 1.2).

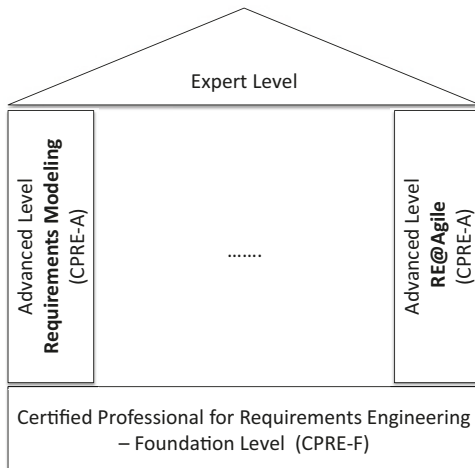


Bild 1.2

Die drei Ebenen des IREB-Zertifizierungsprogramms

Unabhängig davon ist auch in Kanada ein Programm entstanden: das IIBA (International Institute of Business Analysis, www.iiba.org), das ähnliche Ziele wie das IREB verfolgt. In diesem Buch lernen Sie auch alle relevanten Teile des BABOK (Business Analysis Book of Knowledge) kennen. Auch IIBA bietet ein mehrstufiges Zertifizierungsprogramm an.

■ 1.6 Drei Säulen erfolgreicher Projekte

Beginnen wir mit einer Gesamtbetrachtung von Projekten oder Produktentwicklungsvorhaben. Wie passt unser Thema „Business Analysis & Requirements Engineering“ in den gesamten Lebenszyklus hinein? Vielleicht hat Ihre Firma, Ihr Unternehmen, Ihre Organisation ein vorgeschriebenes Vorgehensmodell. Besonders populär im Behördenumfeld ist zum Beispiel das V-Modell, das von einigen Ministerien für staatliche Projekte vorgeschrieben wird. Ein anderes populäres Modell war der Rational Unified Process, heutzutage von IBM Rational vertreten. Wenn Sie nach einem dieser Vorgehensmodelle arbeiten müssen, haben Sie sicherlich festgestellt, dass Sie sehr, sehr viele Tätigkeiten machen müssen, viele Dokumente erzeugen und verschiedenste Rollen besetzen. Die Zahl der Dokumente bei diesen Vorgehensmodellen ist dreistellig; es sind auch zirka 150 – 170 getrennte Tätigkeiten.

Ich möchte versuchen, Ihnen ein einfacheres Weltbild nahezulegen, das leichter zu merken ist und aus nur drei Grundpfeilern besteht (vgl. Bild 1.3). Einer von diesen dreien ist, herauszufinden, welches Problem der Auftraggeber, ein Kunde oder eine Organisation hat; das Problem zu definieren, das Problem zu durchleuchten und zu verstehen. Das ist unser Thema Business Analysis und Requirements Engineering. Vielleicht sagen Sie auch Geschäftsprozessanalyse dazu; vielleicht verwenden Sie auch das Wort Systemanalyse. Wir meinen auf jeden Fall: herauszubekommen, was unser Geschäft oder unsere Kunden brauchen bzw. was unsere Anwender von IT-Systemen erwarten oder mit den Produkten machen wollen.

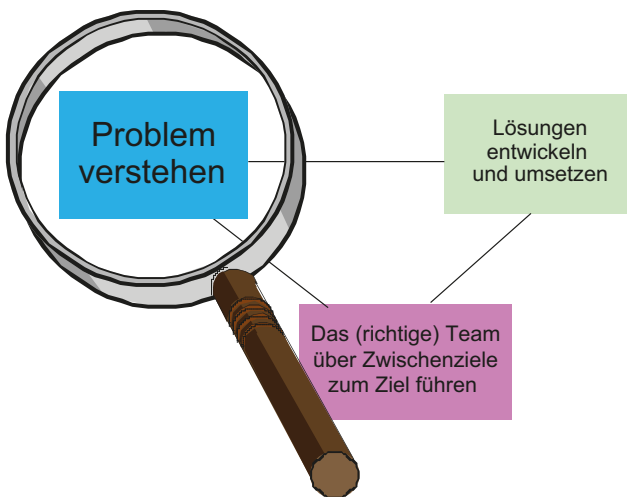


Bild 1.3

Drei Säulen für erfolgreiche Projekte

Die zweite Säule meines einfachen Weltbilds ist es, die Anforderungen in eine Lösung umzusetzen. Wenn diese Lösung eine IT-Lösung ist, dann gehört dazu der Aufbau einer Softwarearchitektur, vielleicht auch einer Hardwarearchitektur und die Entwicklung, der Test und die Inbetriebnahme von Software. Wenn die Lösungen organisatorisch bewerkstelligt werden sollen, dann umfasst dieser Block die detaillierten Festlegungen und die Umsetzung in Aufbau- und Ablauforganisation. Kurz gesagt: eine komplette Lösung für ein vorgegebenes Problem erstellen.

Und die dritte Säule in meinem einfachen Weltbild ist Projektmanagement: das richtige Team zu haben; diesem Team gute Ziele zu geben, vernünftige Pläne aufzustellen, Zwischenziele festzulegen, deren Erreichung man prüfen kann, und das Team erfolgreich über diese Zwischenziele zum Ziel zu führen.

Die Tätigkeiten in den drei Säulen werden heutzutage hochgradig iterativ durchgeführt. Ständig werden Sie gleichzeitig Probleme analysieren, Lösungen designen und Ihre Vorhaben oder Produktentwicklungen managen.

Mit diesen drei Säulen können Sie Projekte erfolgreich abwickeln, Ihr Business effektiver gestalten und Ihre Produkte erfolgreicher machen. Dieses Buch handelt hauptsächlich von der ersten dieser drei Säulen: von Business Analysis und Requirements Engineering. Wir werden aber auch die Schnittstellen zwischen diesen beiden Themen und der Umsetzung in Lösungen sowie die Schnittstellen zwischen Business Analysis/Requirements Engineering und Projektmanagement ausführlich behandeln.

■ 1.7 Definition: Business Analysis und Requirements Engineering

Bevor wir uns mit den Inhalten von Business Analysis und Requirements Engineering näher beschäftigen, sollten wir uns auf Definitionen einigen. Was also verbirgt sich hinter den beiden Begriffen?

Das International Requirements Engineering Board [IREB], dessen Programm diesem Buch zugrunde liegt, hat die Gesamtheit des Themas als Requirements Engineering bezeichnet. Die beiden Hauptteile, die dazugehören, sind einerseits die Requirements-Analyse – von vielen auch als Systemanalyse bezeichnet – und andererseits Requirements-Management, d. h., wenn ich einmal Anforderungen habe, diese auch zu managen, zu verwalten und zu pflegen. Zu den Unterpunkten der Requirements-Analyse gehört einerseits, Requirements aus Kunden herauszulocken, und andererseits, diese auch zu Papier zu bringen – also Requirements ermitteln und spezifizieren.

Sie sehen an dem einfachen Mengendiagramm in Bild 1.4 die drei Teile, aus denen dieses Gebiet besteht:

- Anforderungen ermitteln (oder herauslocken),
- Anforderungen spezifizieren (auf irgendeine Art skizzieren, niederschreiben oder zeichnen) und
- Anforderungen managen (verwalten, pflegen, versionieren, priorisieren).

Die ersten beiden davon werden oft auch als „Requirements-Analyse“ oder „Systemanalyse“ zusammengefasst. Welche Wörter Sie für die Summe aller dieser Tätigkeiten verwenden, überlasse ich Ihrer Entscheidung. Das IREB hat sich für „Requirements Engineering“ als Oberbegriff entschieden; ob Sie das in Ihrem Umfeld als Requirements Engineering bezeichnen oder Requirements-Management als Oberbegriff verwenden oder vielleicht Problemanalyse oder Business-Analyse, überlasse ich Ihnen.

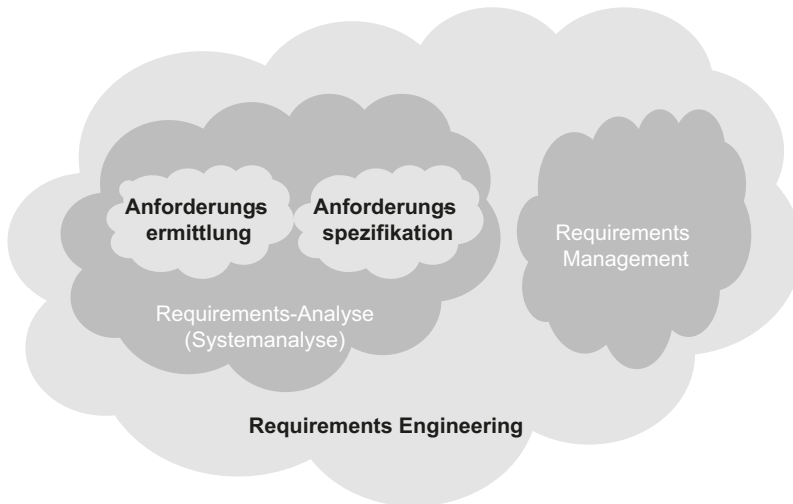


Bild 1.4 Die Hauptthemen im Requirements Engineering

Definition:
Requirements Engineering

Ich möchte Ihnen aber neben diesem Mengendiagramm doch noch eine ausführliche Definition geben. Requirements Engineering im heutigen Sinne ist ein kooperativer, iterativer und inkrementeller

Prozess mit folgenden drei Zielen:

1. Alle relevanten Anforderungen sollen bekannt und im erforderlichen Detaillierungsgrad verstanden sein.
2. Die involvierten Personen und Organisationen (Stakeholder) sollen ausreichende Übereinstimmung über die bekannten Anforderungen erzielen.
3. Die Anforderungen sollen konform zu den Dokumentationsvorschriften der Organisation spezifiziert sein.

Alleine in der Einleitung dieser Definition stecken drei Aussagen drin. Ein kooperativer Prozess heißt: Wir arbeiten zusammen und nicht gegeneinander. Wir wollen nicht Anforderungen über den Zaun werfen zu anderen Leuten, sondern wir wollen sie gemeinsam entwickeln zwischen Auftraggeber und Auftragnehmer oder in Scrum: zwischen Product Owner und Development Team.

Die Wörter „iterativ“ und „inkrementell“ richten sich gegen das alte Wasserfallmodell. Ich muss nicht alle Anforderungen am Anfang perfekt kennen. Ich kann in Iterationen (Releases oder Sprints) – schrittweise – Teile davon, weitere Teile davon, noch mehr Teile davon zeitgerecht entwickeln.

Danach lernen wir durch die Definition die drei wesentlichen Ziele dieses kooperativen Prozesses kennen.

Das erste Ziel ist: Ich möchte alle **relevanten** Anforderungen so gut und im **erforderlichen Detaillierungsgrad** kennen, dass ich eine Lösung darauf basieren lassen kann. Sehen Sie die Weichmacher in dieser Definition? Alle relevanten Anforderungen in hinreichendem Detaillierungsgrad? Was ist relevant? Was bedeutet hinreichender Detaillierungsgrad? Wir werden uns ausführlich mit beiden Fragen auseinandersetzen.

Das zweite Ziel besagt, dass die beteiligten und betroffenen Personen, die Stakeholder, die Organisationen, die mitspielen müssen, hinreichende Übereinstimmungen über diese Anforderungen erzielen sollten. Nun ja, in der Diktatur reicht es, wenn der Diktator ja sagt. Wenn Sie etwas demokratischer aufgestellt sind, brauchen Sie vielleicht die Zustimmung von ein paar mehr Leuten. Aber selbst in einer Demokratie sind 50,01 % eine Mehrheit. Es wird also immer wieder Personen geben, die dagegen sind, die andere Wünsche haben. Wir wollen aber, dass der Prozess dafür sorgt, dass wir eine hinreichende Übereinstimmung über die bekannten Anforderungen erzielt haben.

Und das dritte Ziel für diesen Prozess Requirements Engineering ist noch einfacher. Wir wollen, dass die Anforderungen nach den Regeln des Hauses, nach Ihren Spielregeln, nach Ihren Dokumentationsvorgaben aufgeschrieben werden. Wenn Sie in Ihrem Haus gar keine Vorschriften haben, wie man ein Pflichtenheft oder ein Lastenheft erstellt oder den Product Backlog erfasst, sind Sie auf jeden Fall konform dazu ☺. Aber die meisten Firmen haben natürlich Vorgaben, was alles festzuhalten ist und wie es festgehalten werden sollte.

Requirements Engineering ist also ein kooperativer Prozess, der in Zyklen durchgeführt wird. In jedem Zyklus muss ich genügend herausfinden für das, was ich demnächst umsetzen möchte. Ich brauche genügend Übereinstimmung und ich muss es hausgerecht festhalten, nach unseren Dokumentationsvorschriften.

Wenden wir uns nun dem etwas weiter gefassten Begriff „Business Analysis“ zu. Das International Institute of Business Analysis (IIBA) definiert in seinem Business Analysis Body of Knowledge (BABOK) Folgendes:

Business Analysis ist eine Menge von Aufgaben und Techniken, die genutzt werden, als Liaison zwischen Stakeholdern zu arbeiten, um die Struktur, die Direktiven (Policies) und die Arbeitsweise einer Organisation zu verstehen und (fachliche) Lösungen vorzuschlagen, die es der Organisation ermöglichen, Ihre Ziele zu erreichen.

Definition: Business Analysis

Das entsprechende Überblickbild über Business-Analyse identifiziert acht Kernbereiche, die dort als Wissensgebiete (Knowledge Areas) bezeichnet werden (vgl. Bild 1.5).

Sicherlich sehen Sie die starken Überlappungen zu der Definition von Requirements Engineering. Auch hier stehen die Erhebung (von Anforderungen) und das Managen von Anforderungen rechts und links im Fokus. Analysieren wurde hier dreigeteilt: Enterprise-Analyse und Requirements-Analyse, natürlich nicht ohne beides zu bewerten und zu validieren. Eingerahmt werden diese Kerntätigkeiten durch die Planung und Überwachung des Gesamt-vorgehens sowie die Forderungen nach grundlegenden Kompetenzen der Mitwirkenden, die wir später in diesem Kapitel genauer betrachten werden.

Was haben die beiden Definitionen gemeinsam? Sowohl bei Business Analysis wie auch beim Requirements Engineering geht es um das Verstehen eines Systems (d. h. eines Geschäfts, einer Organisation, eines Produkts oder eines Systems), mit dem Ziel, Schwachstellen zu identifizieren, Ideen für Verbesserungen zu entwickeln, damit zu besseren oder neuen Produkten bzw. zu optimierten Ablauf- und Aufbauorganisationen zu kommen. Bild 1.6 fasst die wesentlichen Aspekte von Business Analysis und Requirements Engineering zusammen:

- Wir gehen davon aus, dass es jemanden gibt, der mit dem Ist-Zustand von Produkten oder Organisationen nicht zufrieden ist. Jemand, der ehrgeizige Ziele hat, Wünsche und Ideen und Visionen, oder aber durch äußere Zwänge wie Gesetze und andere Auflagen gezwungen wird, am Status-quo etwas zu verändern.

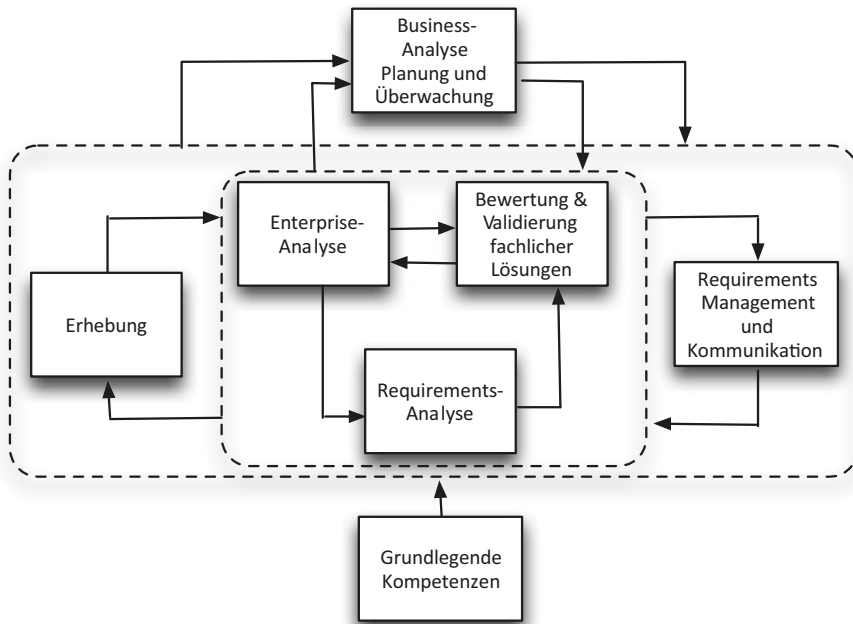


Bild 1.5 Wissensgebiete der Business-Analyse

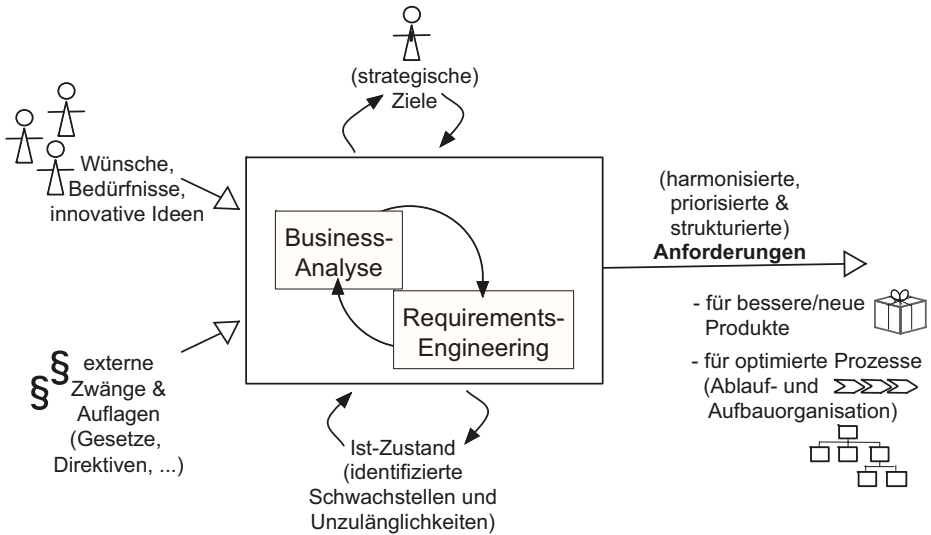


Bild 1.6 Business-Analyse und Requirements Engineering

- Durch Analyse des heutigen Zustands von Organisationen und Produkten identifiziert man Schwachstellen und entwickelt Lösungsideen und Verbesserungsvorschläge, die man in Form von möglichst präzisen Anforderungen an jemanden übergeben kann, der in der Lage ist, die Lösung in die Praxis umzusetzen. Entweder mittels IT-Systemen oder aber auch durch organisatorische Maßnahmen.
- Diese Tätigkeiten werden gemeinsam von allen Stakeholdern durchgeführt, die ein Interesse an der Sache haben, in einem kooperativen Prozess, wobei wir ausgewogen über kurz-, mittel- und langfristige Lösungen nachdenken. Je kurzfristiger wir etwas ändern wollen, desto genauer müssen wir Anforderungen spezifizieren. Je langfristiger Themen sind, desto vager können wir im Ergebnis bleiben. Durch iteratives und inkrementelles Vorgehen sichern wir jederzeit die Konzentration und Bündelung unserer Kräfte auf das zurzeit Notwendige.

■ 1.8 Definition: Requirement

Jetzt haben wir so häufig das Wort „Requirements“ erwähnt. Was überhaupt ist eine einzelne Anforderung, ein Requirement? Nicht alles, was uns Projektbeteiligte erzählen, ist automatisch eine Anforderung. Sie sprechen über viele Bedürfnisse, über Wünsche. Sie geben uns zahlreiche Hintergrundinformationen. Sie erklären uns vielleicht sogar, warum sie das Ganze haben wollen, und geben uns Erläuterungen aus dem bisherigen Leben, aus Altsystemen oder was in Zukunft sein soll. Zur Anforderung wird es erst dann, wenn Sie es halbwegs formalisieren und identifizierbar machen. Identifizierbar, denn im Zweifelsfall wollen wir sagen, das war eine Anforderung oder das war keine Anforderung. Es nur gesagt zu haben oder nur erwähnt zu haben, reicht oft nicht aus. Wir wollen eine identifizierbare Aussage haben und sie sollte auch halbwegs formalisiert sein.

Anforderungen
sind identifizierbar und
halbwegs formalisiert

Sie brauchen aber keine Angst zu haben. Sie müssen nicht unbedingt komplexe Diagramme lernen. Ein sauber geschriebener deutscher Satz mit Subjekt, Prädikat und Objekt oder ein sauber geschriebener englischer Satz oder eine gut formulierte User-Story erfüllt auch das Kriterium, halbwegs formalisiert zu sein. Aber die beiden Punkte brauchen wir, denn wir brauchen eine Einigung, ob irgendwas eine Anforderung war oder nicht, denn wir schreiben Anforderungen hauptsächlich deshalb, damit wir beim Übergeben einer Lösung, beim Abnahmetest, beim Sprintende feststellen können, ob die Anforderung auch erfüllt ist. Wenn wir zu diesem Zeitpunkt nichts hätten, wogegen wir prüfen können, nichts, was wir identifizieren können, nichts, was halbwegs eindeutig formuliert ist, dann hätten wir Probleme; wir könnten nicht feststellen, ob das geliefert wurde, was wir bestellt haben oder haben wollten. Für Formalisten hat IEEE, die amerikanische Ingenieursvereinigung, natürlich formale Definitionen festgehalten, was Anforderungen sind. Gemäß IEEE 610.12-1990 ist eine Anforderung

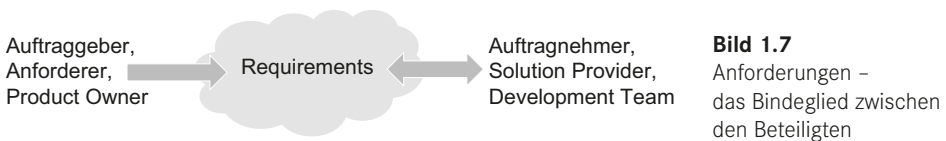
1. eine Bedingung oder Fähigkeit, die von einem Benutzer (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.

2. eine Bedingung oder Fähigkeit, die ein System oder Teilsystem erfüllen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.
3. eine dokumentierte Repräsentation einer Bedingung oder Fähigkeit gemäß (1) oder (2).

Umgangssprachlich ausgedrückt heißt der erste Punkt: etwas, was wir **wollen**, was irgendjemand will. Der zweite Teil heißt umgangssprachlich ausgedrückt: etwas, was wir haben **müssen**, ob wir wollen oder nicht, weil ein Gesetz oder eine Norm oder eine Vorschrift dahinter steckt. Und egal, ob es ein Wunsch war oder ob wir es haben müssen, jede Niederschrift einer solchen Aussage zählt natürlich auch als Anforderung. Also nicht nur das gesprochene Worte, sondern vor allem die dokumentierte Form von solchen Wünschen oder Zwängen zählen als Anforderung.

Der neue IEEE-Standard 29148 ist etwas pragmatischer. Er definiert: Eine Anforderung ist eine Aussage, die einen Wunsch samt seinen Einschränkungen und Randbedingungen ausdrückt oder übersetzt.

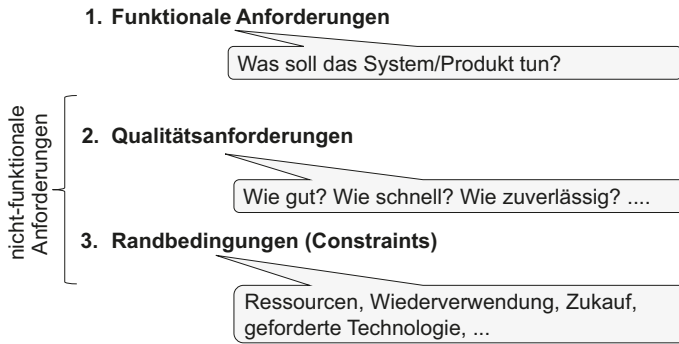
Bild 1.7 bringt diese Aussagen auf den Punkt: Anforderungen sind das Bindeglied zwischen zwei Parteien: jemanden, der etwas haben will (Auftraggeber, Kunde, Anforderer, Product Owner, ...), und jemanden, der es liefern kann (Auftragnehmer, Solution Provider, Development Team, ...).



Es ist egal, wer der Auftraggeber ist, Ihre interne Marketingabteilung, ein Kunde oder ein wirklicher Endnutzer. Es ist auch egal, ob Sie im gleichen Unternehmen sitzen oder unterschiedlichen Unternehmen angehören. Auftragnehmer ist jemand, der das Ganze umsetzen kann; jemand, der diese Anforderung nimmt und daraus eine gewünschte (organisatorische oder technische) Lösung erstellen kann. Anforderungen sind das Bindeglied zwischen diesen beiden Welten, egal ob sich dazwischen Firmengrenzen, Abteilungsgrenzen oder Systemgrenzen befinden. Es ist das, was die beiden Parteien zusammenbringt.

■ 1.9 Arten von Anforderungen

Wir haben jetzt geklärt, was Anforderungen sind. Als Nächstes wollen wir über unterschiedliche Arten von Anforderungen sprechen. Die Klassifizierung soll uns helfen, Anforderungen leichter zu finden und zu erfassen. Wir wollen zwei bis drei Arten von Anforderungen unterscheiden (vgl. Bild 1.8). Die klassische Zweiteilung erfolgt in funktionale und nichtfunktionale Anforderungen.

**Bild 1.8**

Kategorisierung
von Anforderungen

Funktionale Anforderungen sind eher einfach. Das sind die Funktionen, die das System leisten soll; die Informationen, die es verarbeiten soll; das gewünschte Verhalten, welches das System an den Tag legen soll.

Nichtfunktionale Anforderungen ist eigentlich ein Unwort. Auch im Englischen: Non-functional Requirements. Nichtfunktionale Anforderungen? Was soll das sein? Die Dinge, die hinterher nicht funktionieren? Nein. Das ist nicht damit gemeint. Es ist einfach das Gegenstück zu den funktionalen Anforderungen: alles andere außer den Funktionen und Informationen des Systems. Und es wird wesentlich klarer, wenn wir sie noch in zwei Teile unterteilen: in die Qualitätsanforderungen und in Randbedingungen.

Qualitätsanforderungen sind Ergänzungen zu den Funktionen; präziser gesagt: zu einer oder mehreren Funktionen. Sie sagen uns, wie gut die Funktionen ausgeführt werden sollen, wie schnell, wie zuverlässig, wie gesetzestreu, wie sicher, wie wartbar ... Es reicht nicht, zu sagen: „Mein Auto soll bremsen können.“ Sie wollen, dass es einen Meter vor der Mauer stehen bleibt und nicht erst einen Meter hinter der Mauer. Das ist eine Qualitätsanforderung an die Funktion „bremsen können“. Und diese Qualitätsanforderungen sind genauso wichtig und wir wollen sie genauso sorgfältig erforschen und erfassen wie die funktionalen Anforderungen.

Die dritte Kategorie Randbedingungen, im englischen Constraints genannt, ist etwas schwieriger. Denn viele Leute sind der Meinung, dass das eigentlich keine Anforderungen sind. Aber zu dieser Kategorie gehören technische und organisatorische Vorgaben, die ein Auftraggeber natürlich machen kann. Wenn der Auftraggeber sagt: „Du darfst maximal 300.000,- € ausgeben“, so ist das weder eine funktionale Anforderung noch eine Qualitätsanforderung. Es ist einfach eine Randbedingung. „Du musst bis Dezember fertig sein“ ist eine zeitliche Randbedingung. „Du darfst die Aufbauorganisation nicht verändern“ ist eine organisatorische Randbedingung. „Du sollst Rechner von Hersteller X einsetzen“ oder „Du sollst diese Datenbank verwenden“ sind technische Randbedingungen. „Du sollst dieses Framework verwenden“. Alles Randbedingungen. Viele betrachten sie als unechte Anforderungen; sie gehören aber definitiv zu der Menge der Anforderungen dazu.

Etwas formaler ausgedrückt: Funktionale Anforderungen definieren Funktionen, die vom Gesamtsystem oder von einem Teilsystem, von einem Modul, von einer Komponente ausgeführt werden sollen.

Qualitätsanforderungen definieren qualitative Eigenschaften, die diese Funktionen haben sollen. Wiederum sind das entweder qualitative Eigenschaften vom Gesamtsystem oder von einem großen Teilsystem oder von einer einzelnen Funktion; aber sie stehen niemals im

luftleeren Raum. Sie gehören immer zu einer oder mehreren funktionalen Anforderungen dazu. Und die Randbedingungen, die Constraints, sind organisatorische oder technische Vorgaben, die dem Designer, den Lösungsverantwortlichen, die Hände fesseln. Sie haben nicht jegliche Freiheitsgrade, eine Lösung zu entwickeln, sondern sie müssen sich bei dieser Lösung an solche Randbedingungen halten, an Zeitvorgaben, an Budgetvorgaben, an Technologievorgaben. Freiheitsgrade, die dem Designer durch den Auftraggeber entzogen werden, aus gutem Grund. Das sind unsere drei Kategorien von Anforderungen.

■ 1.10 Vier Hauptaufgaben eines Analytikers

Im nächsten Abschnitt wollen wir uns die Hauptaufgaben eines Business Analyst, eines Requirements Engineer oder eines Product Owners ansehen. Bild 1.9 zeigt, dass wir sie auf vier wesentliche Punkte reduzieren können.

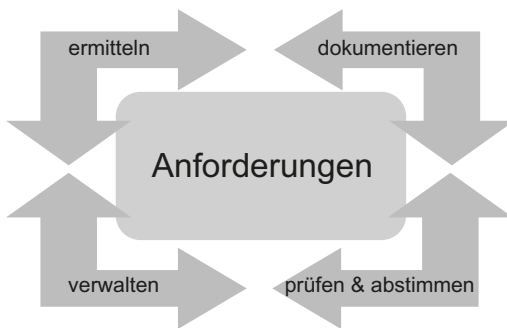


Bild 1.9

Die Hauptaufgaben eines Analytikers

Ermitteln

Erster wesentlicher Punkt: Ich muss Anforderungen ermitteln können. Ich muss sie herauskitzeln, ich muss sie begreifen und verstehen können, ich muss wissen, was der Kunde haben möchte, und Methoden haben, um darauf zu kommen. Als ich vor 30 Jahren gelernt habe, waren Interviews die einzige Technik, um aus Kunden Wünsche herauszubekommen. Wir werden später im Buch (in Kapitel 10) diskutieren, dass das bei Weitem nicht die einzige Technik ist. Wir haben heute Dutzende von Erhebungstechniken, um Anforderungen aus Stakeholdern herauszulocken, nicht nur Interviews.

Dokumentieren

Die zweite wesentliche Tätigkeit ist, verstandene Anforderungen zu dokumentieren. Ich habe sie begriffen, ich muss sie aber noch irgendwie schriftlich festhalten. Und dazu stehen uns im Wesentlichen drei Möglichkeiten zur Verfügung: schreiben, malen oder simulieren. Schreiben heißt, Sätze schreiben. Deutsche, englische, natürlichsprachige Sätze schreiben, die die Anforderung wiedergeben. Malen heißt, formale grafische Modelle einzusetzen: BPMN-Diagramme, Use-Case-Diagramme, Datenmodelle, Aktivitätsdiagramme; die vielen Modelle, die uns die Methoden heute zur Verfügung stellen, um vielleicht etwas präziser als nur in Umgangssprache auszudrücken,

was gewollt wird. Und simulieren heißt, wir machen Screenshots von IT-Lösungen, wir zeigen Masken, wir zeigen Maskenfolgen, um zum Beispiel die Akzeptanz zu prüfen, ob eine derartige Umsetzung den Wünschen des Auftraggebers gerecht würde. Oder wir bauen Prototypen und Mock-Ups, die wir als Diskussionsgrundlage vorführen können. Sie haben also drei unterschiedliche Arten zur Verfügung, um zu spezifizieren, was Anforderer wollen oder brauchen. Wir werden uns sehr ausführlich mit allen drei Arten beschäftigen und die alternativen Dokumentationstechniken sowie ihre Vorteile und Nachteile genauer erläutern.

Jetzt haben wir Anforderungen ermittelt und niedergeschrieben.

Als Nächstes müssen wir sie prüfen und eventuell konsolidieren.

Prüfen und konsolidieren

Gerade wenn viele Leute gleichzeitig arbeiten und unabhängig voneinander Anforderungen niederschreiben, kann es schon vorkommen – ja nicht nur kann, es wird wahrscheinlich vorkommen –, dass wir Widersprüche haben, dass unterschiedliche Wünsche von unterschiedlichen Leuten nicht zusammenpassen. Und genau daran müssen wir arbeiten. Wer sollte jetzt Recht bekommen? Welche Anforderung wollen wir wirklich erfüllen? Sind wir schon widerspruchsfrei, sind wir in uns konsistent? Wir wollen den Lösungspartnern ja keine widersprüchlichen Wünsche übergeben. Das sicherzustellen, ist die dritte Haupttätigkeit.

Und für die vierte Haupttätigkeit habe ich leider noch eine sehr schlechte Nachricht für Sie. Anforderungen ändern sich mit ein bis

Verwalten, managen

drei Prozent pro Monat. Wenn wir also ein längeres Projekt machen, über mehr als einen Monat hinweg, ist die Wahrscheinlichkeit sehr groß, dass Anforderungen auch geändert werden – mit ein bis drei Prozent Wahrscheinlichkeit. Eine Projektdauer von zwölf Monaten und Sie haben die Chance auf $12 \times 2\%$ – im Schnitt also 24% – geänderte Anforderungen. Deshalb brauchen wir als weitere Tätigkeit Requirements-Management bzw. Requirements verwalten. Das Verwalten von Requirements fasst alle Tätigkeiten mit einmal geschriebenen Anforderungen zusammen. Wie versionieren wir sie? Wie stellen wir fest, welche momentan gültig ist? Wie legen wir Prioritäten fest? Wie gehen wir mit Change-Requests, mit Änderungsanträgen, um? Und wie machen wir Traceability zwischen Analyse und Design und Sourcecode und Testdaten? Alle diese Themen fallen in den Bereich „Requirements verwalten“ oder „Requirements managen“.

Herauskitzeln, Niederschreiben, Prüfen und Verwalten sind die vier Haupttätigkeiten, die ein Systemanalytiker durchzuführen hat.

Lassen Sie mich eine kleine Geschichte erzählen. Als ich 1976 meinen ersten Job angetreten habe und – wie im Vorort erwähnt – von meinem Chef meinen Arbeitsvertrag mit der Berufsbezeichnung Systemanalytiker überreicht bekam, wollte ich das, was wir an der Technischen Universität Wien nicht gelernt hatten, natürlich nachholen. Wir haben damals Kollegen zu 5-Tage-Kursen geschickt, um über Systemanalyse zu lernen. Als die zurückkamen, habe ich neugierig gefragt: Was habt ihr denn gelernt? Was ist und tut denn ein Systemanalytiker? Und die haben mir die Essenz des Kurses mitgeteilt: „Zieh’ Dich anständig an, wenn Du zum Kunden gehst, sei höflich, tritt dem Kunden nicht auf die Füße, hör zu und schreib auf, was er sagt. Und das wichtigste Handwerkszeug sind Radiergummi, Bleistift und gesunder Menschenverstand.“ Nun, die Welt hat sich seither kräftig weiterentwickelt. Wir wissen heute viel mehr über dieses Berufsbild, und das alles sollen Sie im Rest des Buchs erfahren.

■ 1.11 Benötigte Fähigkeiten

Vorher wollen wir uns allerdings ansehen, welche Eigenschaften ein Systemanalytiker, Business Analyst, ein Requirements Engineer oder ein Product Owner haben sollte, um den Job wirklich gut ausführen zu können.

Analytisches Denken

Ich glaube, allem voran ist analytisches Denken gefragt. Ein Problem durchleuchten zu können, abstrahieren zu können, auf den Punkt zu bringen, das Wesentliche herauszufinden; ein gerütteltes Maß an analytischem Denken ist Voraussetzung für die Ausübung dieses Berufs.

Selbstbewusstes Auftreten und Kommunizieren

Außerdem sollte ein Systemanalytiker ein halbwegs selbstbewusstes Auftreten haben. Als Programmierer können Sie sich hinter dem Rechner verstecken. Sie müssen nur mit Ihrem Rechner kämpfen. Sie müssen nicht mit Menschen sprechen. Ich übertreibe bewusst. Als Systemanalytiker haben Sie ständig Kontakt mit vielen Leuten und Sie müssen diese auch koordinieren. Dazu gehören Moderationsfähigkeiten, aber auch ein gesundes Einfühlungsvermögen und auf jeden Fall gestandenes Selbstbewusstsein. Sie müssen sich mit größeren Gruppen von Leuten auseinandersetzen. Sie brauchen Überzeugungsfähigkeiten und Kommunikationsfähigkeiten. Sie müssen auf die Ausdrucksweise, die Sprechweise, das Verständnis Ihrer Gesprächspartner eingehen können. Manchmal kann man nur über Prototypen kommunizieren, manchmal über deutsche Sätze, manchmal mit grafischen Modellen. Nicht Sie bestimmen, was gute Kommunikation ist. Meistens bestimmen Ihre Projektpartner, wie man am besten und am effizientesten kommuniziert. Sie sollten all diese Variationen der Kommunikation beherrschen. Dazu gehört auch sprachliche Kompetenz. Sie müssen sich in der Projektsprache vernünftig ausdrücken und klare, eindeutige Sätze formulieren können.

Methodische Kompetenz

Zu den Fähigkeiten gehört auch methodische Kompetenz – das ist der Schwerpunkt, den Ihnen dieses Buch vermitteln kann. In den letzten 35 Jahren wurden viele Methoden entwickelt, wie man Anforderungen ermittelt, dokumentiert, prüft und verwaltet. Die können Sie lernen. Viele von den anderen Fähigkeiten sollten Sie mitbringen.

Domänenkenntnisse?

Lange diskutiert haben wir im IREB über Domänenkenntnisse. Muss ein Systemanalytiker gute Domänenkenntnisse haben, also zum Beispiel Versicherungsexperte sein, wenn wir ein Versicherungsprojekt durchführen, oder Getriebeexperte, wenn Sie über Getriebesteuerungen sprechen? Wir haben uns schließlich darauf geeinigt, Domänenkenntnisse helfen, sind aber nicht unbedingt Voraussetzung für einen guten Systemanalytiker. Sehen Sie mich an. Ich bin das beste Beispiel dafür. Ich arbeite in vielen Branchen, von denen ich anfangs wenig Ahnung habe. Irgendjemand in der Gruppe von Personen, mit denen Sie arbeiten, sollte allerdings Ahnung haben. Und Ihr Geschick als Systemanalytiker ist es, dieses Wissen aus den anderen herauszuholen und in geordneter Form zu kommunizieren. Sie müssen nicht Topspezialist in dieser Domäne sein. Aber ein gesundes Verständnis des Umfelds, in dem Sie arbeiten, hilft auf jeden Fall und macht Systemanalyse leichter.

Überlegen Sie einmal, was ein Systemanalytiker überhaupt nicht können muss, um seinen Job gut auszuüben: C# zum Beispiel, die neuesten NOSQL-Datenbanken, die letzten Versionen von irgendwelchen Hardwaresystemen. Dafür haben wir Designer, Programmierer,

Hardwareingenieure. Deren Job ist es, eine Lösung zu bauen. Also, Sie brauchen kein Technologiespezialist sein. Es schadet natürlich auch nicht, wenn Sie die Sprache der Leute, die hinterher weiterarbeiten, verstehen. Aber das ist nicht Ihre Hauptaufgabe.

Ein paar andere Punkte noch: Sie müssen nicht auf jedem Gebiet, auf dem Sie Anforderungen erfassen, Topspezialist sein. Sie werden juristische Anforderungen behandeln. Für deren Präzisierung holt man sich einen Rechtsanwalt oder Hausjuristen zu Hilfe. Sie werden Sicherheitsanforderungen behandeln. Dazu haben Sie im Haus vielleicht einen Sicherheitsbeauftragten oder einen Sicherheitsspezialisten. Sie müssen nur in der Lage sein, mit solchen Personengruppen vernünftig umzugehen und deren Wissen transparent zu machen. Also bitte suchen Sie sich für Spezialgebiete Unterstützung. Sie müssen nicht alle Themen alleine beherrschen.

Ist also ein Systemanalytiker eher eine extrovertierte Person oder eine introvertierte Person? Ich glaube, Sie haben es herausgehört. Introvertiert können Sie als Programmierer sein; als Systemanalytiker sollten Sie extrovertiert sein und mit Menschen gerne und gut umgehen können.

■ 1.12 Aufgabenverteilung im Team

Wir haben bis jetzt immer über die Person eines Requirements Engineer, Business Engineer oder Product Owner gesprochen. Sehen wir uns die Aufgabenverteilung im Team einmal an, wer alles eine Rolle in Zusammenhang mit Systemanalyse hat. Wenn Sie ein explizites Rollenbild eines Requirements Engineer oder eines Business Analyst haben, dann wird diese Person hauptsächlich die vier Tätigkeiten durchführen (ermitteln, dokumentieren, verwalten und auch prüfen und abgleichen). Ich kenne aber Projekte, wo es diese Rolle gar nicht explizit gibt. Und trotzdem muss irgendjemand diese Tätigkeiten durchführen. Irgendjemand im Team macht das dann. Ob das Entwickler oder der Product Owner oder wer auch immer übernimmt, ist egal.

Trotzdem, selbst wenn Sie diese Spezialrolle haben, haben auch andere Personen Aufgaben im Zusammenhang mit Business Analysis und Requirements Engineering, wie Bild 1.10 zeigt.

Nehmen wir einmal den Auftraggeber. Der trägt schließlich die Verantwortung für das Ganze; der will es haben; der ist der Geldgeber dafür und der wird vielleicht Ziele und grobe Anforderungen vorgeben. Und unsere Benutzer, die Personen, die später die Lösung nutzen sollen. Von denen – zusammen mit anderen Stakeholdern – wird der größte Teil der Anforderungen kommen, die Sie als Systemanalytiker aufschreiben sollen.

Als Analytiker haben Sie meistens nicht sehr viel Macht, denn die Entscheidung über Prioritäten, wie wichtig diese Anforderung ist, ob sie frühzeitig oder später erledigt werden sollte, trifft sehr oft der Projektleiter in Abstimmung mit dem Auftraggeber. Sie bereiten vor, Sie geben Abhängigkeiten an, Sie sagen, das eine kann nicht gemacht werden ohne das andere; aber die Entscheidung darüber trifft ein Projektleiter. Also hat auch ein Projektleiter eine Rolle in Zusammenhang mit Anforderungen. Und Projektleiter werden meistens auch Soll und Ist überwachen. Wie weit sind wir fortgeschritten? Und was ist das Nächstwichtige.