

7.

Auflage



Christof Ebert

Systematisches Requirements Engineering

Anforderungen ermitteln, dokumentieren,
analysieren und verwalten

dpunkt.verlag

Inhalt

Cover

Über den Autor

Titel

Impressum

Vorwort zur 7. Auflage

Stimmen zum Buch

Inhaltsübersicht

Inhaltsverzeichnis

1 Motivation

1.1 Warum ein Buch über Requirements Engineering?

1.2 Projekte scheitern wegen unzureichender Anforderungen

1.3 Wirtschaftlicher Nutzen und Return on Investment (ROI)

1.4 Wie Sie von diesem Buch profitieren

1.5 Selbsttest

1.6 Ein Blick über den Tellerrand

2 Requirements Engineering – kurz und knapp

2.1 Was ist eine Anforderung?

2.2 Perspektiven: vom Markt zur Realisierung

2.3 Arten von Anforderungen

2.4 Was ist Requirements Engineering?

2.5 Requirements Engineering in der Praxis

2.6 Terminologie

2.7 Durchgängiges Beispiel: iHome

2.8 Tipps für die Praxis

2.9 Fragen und Impulse

3 Anforderungen ermitteln

- 3.1 Ziel und Nutzen
- 3.2 Bedürfnisse verstehen, Ziele vereinbaren
- 3.3 Stakeholder managen
- 3.4 In 10 Schritten zu guten Anforderungen
- 3.5 Qualitätsanforderungen und Randbedingungen
- 3.6 Fallstudie: Security Requirements Engineering
- 3.7 Checkliste für die Anforderungsermittlung
- 3.8 Tipps für die Praxis
- 3.9 Fragen und Impulse

4 Anforderungen dokumentieren

- 4.1 Ziel und Nutzen
- 4.2 Lasten und Pflichten: vom Was zum Wie
- 4.3 Dokumentation und Vorlagen
- 4.4 Struktur und Lesbarkeit
- 4.5 Attribute und Filter
- 4.6 Glossar
- 4.7 Checkliste für die Dokumentation
- 4.8 Tipps für die Praxis
- 4.9 Fragen und Impulse

5 Anforderungen modellieren und analysieren

- 5.1 Ziel und Nutzen
- 5.2 Modelle und Methoden
- 5.3 Systems Engineering und Architektur
- 5.4 UML, SysML und BPMN
- 5.5 Aufwandsschätzung
- 5.6 Analyse in zehn Schritten
- 5.7 Checkliste für die Anforderungsanalyse

5.8 Tipps für die Praxis

5.9 Fragen und Impulse

6 Anforderungen prüfen

6.1 Ziel und Nutzen

6.2 Qualitätskriterien für Anforderungen

6.3 Verfahren zur Prüfung

6.4 Test-Driven Requirements Engineering (TDRE)

6.5 Kriterien für Testende und Abnahme

6.6 Checkliste zur Prüfung von Anforderungen

6.7 Tipps für die Praxis

6.8 Fragen und Impulse

7 Anforderungen abstimmen

7.1 Ziel und Nutzen

7.2 Abstimmung im Kernteam

7.3 Risiken abschwächen

7.4 Priorisierung von Anforderungen

7.5 Recht, Compliance und Haftung

7.6 Verträge und Vertragsmodelle

7.7 Checkliste für Abstimmung und Verträge

7.8 Tipps für die Praxis

7.9 Fragen und Impulse

8 Anforderungen verwalten

8.1 Ziel und Nutzen

8.2 Änderungsmanagement

8.3 Verfolgbarkeit (Traceability)

8.4 Wartung und Altsysteme

8.5 Versionierung und Varianten von Anforderungen

8.6 Kennzahlen und KPI

8.7 Checkliste für die Verwaltung

8.8 Tipps für die Praxis

8.9 Fragen und Impulse

9 Agiles Requirements Engineering

9.1 Agile Entwicklung

9.2 Komplexität beherrschen

9.3 Praxis des agilen RE

9.4 Design Thinking

9.5 Skalierbare Agilität

9.6 Fallstudie: Agiles Systems Engineering bei Ford

9.7 Fallstudie: Agiles RE bei Siemens

9.8 Tipps für die Praxis

9.9 Fragen und Impulse

10 Werkzeuge

10.1 Ziel und Nutzen

10.2 Werkzeuge und Bewertung

10.3 Praxis: von DOORS bis PREEvision

10.4 Werkzeuge einführen

10.5 Checkliste für Werkzeuge

10.6 Tipps für die Praxis

10.7 Fragen und Impulse

11 Requirements Engineering leben

11.1 Organisation

11.2 Projektmanagement

11.3 Produktmanagement

11.4 Lieferantenmanagement

11.5 Serviceorientiertes RE

11.6 Fallstudie: Funktionsmodellierung und Produktlinien

11.7 Fallstudie: Besseres RE in 10 Schritten

11.8 Tipps für die Praxis

11.9 Fragen und Impulse

12 Soft Skills und persönliche Entwicklung

12.1 Aufgaben des »Requirements Engineer«

12.2 IREB und Zertifizierung

12.3 Soft Skills

12.4 Konflikte lösen

12.5 Tipps für Ihre persönliche Entwicklung

12.6 Fragen und Impulse

13 Stand der Technik und Trends

13.1 Der »Stand der Technik«

13.2 Standards und Normen

13.3 Benchmarks, Faustregeln und Kennzahlen

13.4 Trends in der IT und Software

13.5 Trends im Requirements Engineering

13.6 Top-10-Tipps

Anhang

A Internetressourcen

B Glossar

C Literatur

Index

4 Anforderungen dokumentieren

Ihr Nutzen aus diesem Kapitel:

»Die Dinge sind nie so, wie sie sind. Sie sind immer das, was man aus ihnen macht.« Der Philosoph Jean Anouilh stellt die eigene Wahrnehmung in den Vordergrund. Anforderungen existieren auch nicht als »Ding an sich«, sondern nur als dokumentierte Wahrnehmung. Doch oft endet diese Dokumentation in ausufernden Dokumenten und wird in E-Mails beschrieben, in denen sich niemand zurechtfindet. Hier zeige ich Ihnen, wie Anforderungen klar und verständlich beschrieben werden. Anforderungen werden strukturiert und prüfbar dargestellt, um den Test später zu erleichtern. Konsequenterweise unterscheiden wir Bedarf (oder Lastenheft) und Realisierung (oder Pflichtenheft). Sie lernen Marktanforderungen, Produkthanforderungen und Komponentenanforderungen mit allen relevanten Attributen zu beschreiben. Es geht darum, Missverständnisse zu vermeiden. Denn nur das, was klar beschrieben ist, wird auch richtig entwickelt.

4.1 Ziel und Nutzen

Nur was klar beschrieben ist, wird auch richtig entwickelt. Kunden und Marketing beschreiben Anforderungen, Wünsche und Geschäftsziele informell und erwarten umgehend Lösungen. Spezifikationen sind ein verständlicher Formalismus, um diese informellen Inhalte zu strukturieren und damit projektauglich zu machen. Eine strukturierte Dokumentation klärt Details, macht Zusammenhänge transparent und erleichtert den Zugriff für alle Beteiligten. Alle weiteren Projektergebnisse bauen auf dieser Dokumentation auf (Abb. 4-1). Die Kommunikation von Anforderungen in Form von Post-it-Stickern oder als verteilte E-Mails ist unzureichend und führt ins Chaos.

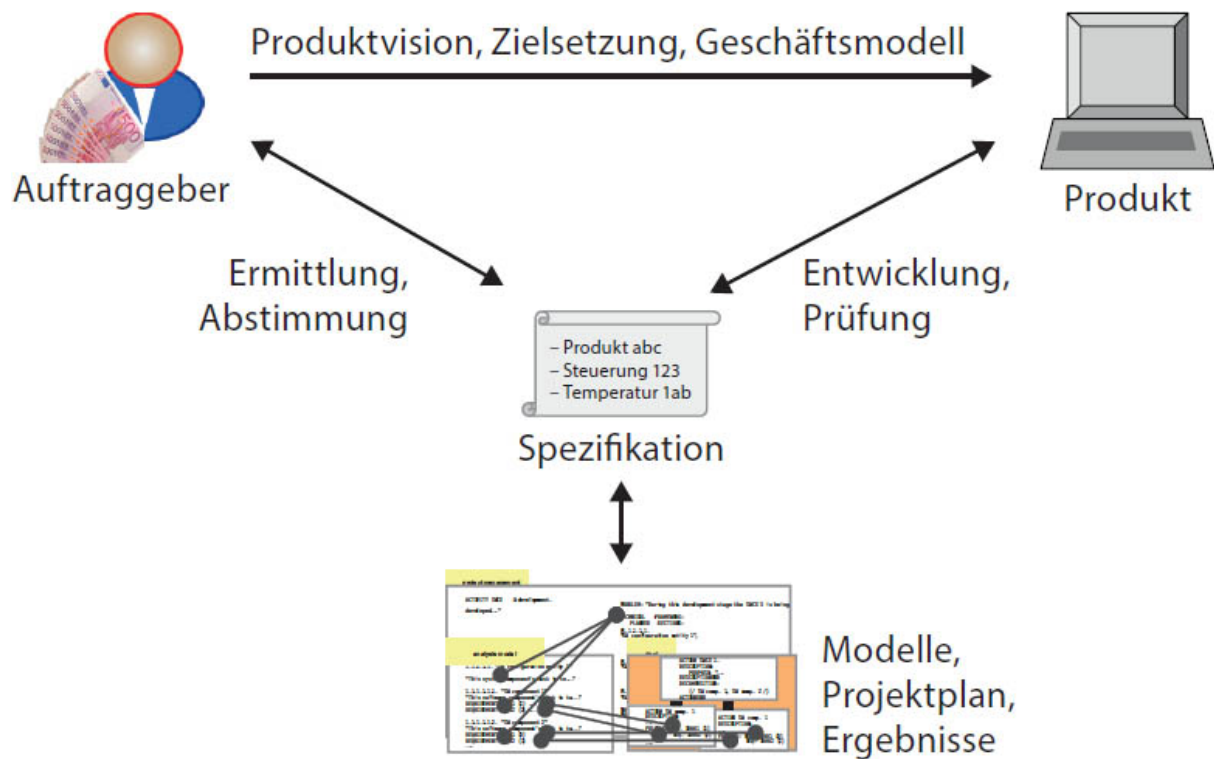


Abb. 4-1 Tragfähige Spezifikationen sind die Basis für alle weiteren Projektergebnisse.

Die Anforderungsspezifikation erwächst aus allen Anforderungen, die zum Projekt gehören (Abb. 4-2). Sie bündelt Kundenwünsche, aber auch interne Vorgaben oder solche Anforderungen, die aus gesetzlichen Randbedingungen erwachsen. Die Anforderungen stammen von allen Stakeholdern und werden auf dem Weg zur Spezifikation (oder Lastenheft) gefiltert, analysiert, bewertet und priorisiert. Häufig müssen sie zunächst einmal klar beschrieben werden, um überhaupt bewertet werden zu können.

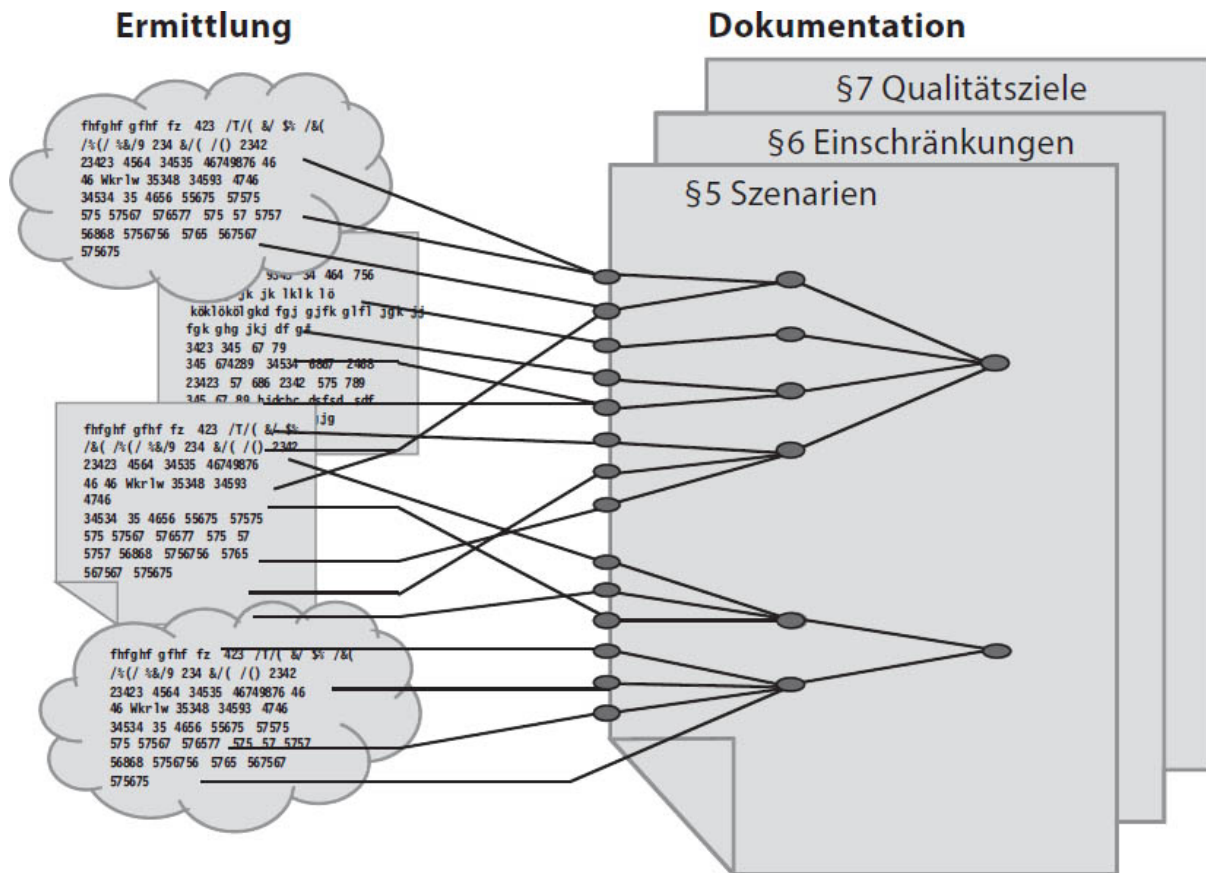


Abb. 4-2 Die Spezifikation strukturiert Anforderungen.

Nicht alle Anforderungen werden dokumentiert. Der weitaus größte Teil der Anforderungen ist impliziter Natur. Niemand geht zum Autohändler und lässt sich vertraglich dokumentieren, dass die Reifen schwarz sind, aus einer Gummimischung bestehen und ein Ventil haben. Dagegen ist eine Reifendruckwarnung Vertragsbestandteil, da sie Sicherheit bringt. Es gibt immer solch implizite Anforderungen, die für die Stakeholder auch ohne Dokumentation klar sind. Implizite Anforderungen sind entweder als wiederverwendete Standards oder Richtlinien referenziert oder sie werden weggelassen. Würden sie dokumentiert werden, bedeutet das Overhead und Ineffizienz im Projekt. Wesentlich ist, dass die relevanten Anforderungen als solche erkannt, ermittelt und dokumentiert werden.

Jeder Schritt im RE bearbeitet bestimmte Dokumente, verändert sie und führt damit schrittweise zum endgültigen Produkt. Definierte Meilensteine und Freigabekriterien in der Prüfung sichern eine ausreichende Qualität dieser Dokumente. Tabelle 4-1 bildet die wesentlichen Arbeitsergebnisse auf die Aktivitäten des Requirements Engineering ab.

Arbeitsergebnis	Ermittlung	Analyse	Prüfung	Abstimmung	Verwaltung
Produktvision	×				
Use Case/ Szenario	×				
Bewertung	×	×	×		×
Lastenheft, SLA	×	×	×	×	×
Projektplan		×	×	×	×
Teststrategie		×	×		×
Lösungsmodell		×	×		×
Pflichtenheft		×	×	×	×
Releaseplanung		×		×	×
Produktkatalog				×	×
Vertrag			×	×	×
Abnahme				×	×

Tab. 4-1 Arbeitsergebnisse und ihr Bezug zu Aktivitäten im Requirements Engineering

4.2 Lasten und Pflichten: vom Was zum Wie

Requirements Engineering übersetzt Bedürfnisse in Lösungen. Das hat sich im deutschsprachigen Raum bereits frühzeitig in der klaren Trennung zwischen Lastenheft und Pflichtenheft niedergeschlagen. Die angelsächsische »Spezifikation« ist unscharf, denn Anforderungen (aus Kunden- oder Benutzersicht) lassen sich nicht in einem Schritt direkt auf eine Lösung abbilden. **Wir unterscheiden zwei grundlegende Perspektiven in den Spezifikationen:**

- **Anforderungsspezifikation**

Sie beschreibt, was und wofür etwas gemacht werden soll. Diese Perspektive wird häufig als **Lastenheft** bezeichnet [VDI2001]. Sie deckt die Marktanforderungen ab. Das Lastenheft gehört dem Auftraggeber und ist vertragsrelevant.

- **Lösungsspezifikation**

Sie beschreibt, wie etwas gemacht werden soll. Diese Perspektive wird häufig als **Pflichtenheft**, Systembeschreibung oder Fachkonzept bezeichnet [VDI2001]. Sie deckt die Produkthanforderungen und Teile der

Komponentenanforderungen ab. Das Pflichtenheft gehört dem Auftragnehmer und ist Basis für alle weiteren Entwicklungsschritte.

Inhaltlich können Lasten und Pflichten sich überlappen – je nachdem, wer sie vorgibt.



Beispiel:

Beim iHome-Aufzug kann eine Schalterfunktion für die Stockwerksauswahl als Hardwareschalter oder als Softwarebutton auf einem Touchscreen realisiert werden. Die Umsetzung als Hardware- oder Softwarelösung kann der Kunde vorgeben, und damit ist es eine Last. Oder aber der Kunde will nur die Funktion der Stockwerksauswahl, und wir legen als Lieferant fest, ob dies durch Hardware oder Software realisiert wird. Dann ist die inhaltlich gleiche Anforderung eine Lösungsspezifikation.

Eine klare und konsistente Spezifikation von Anforderungen und Lösung, die frühzeitig und konsistent im Lastenheft und Pflichtenheft dokumentiert wird, hat viele Vorteile:

Einheitliche Basis für alle Anforderungen

Wer kennt nicht die Situation, in der ein Projektmanager über einen Satz von Anforderungen verfügt, während der Produktmanager oder Vertriebsbeauftragte bereits ganz andere Anforderungen mit dem Kunden verhandelt? Und was der Tester im Labor vorfindet, ist eine interne Interpretation der Anforderungen, die ihm der Entwickler geschickt hat. Ein solches Tohuwabohu führt nicht nur zu Inkonsistenzen, sondern auch zu viel Nacharbeit und Frustrationen. Die Anforderungsspezifikation ist ein Dokument, das für alle Beteiligten – intern oder extern – zentral und versioniert vereinbart, was zu tun ist.

Vertragsbasis mit klarer Trennung zwischen Aufgabe und Lösung (Abb. 2–2)

Je nach Projektcharakter existieren Verträge mit expliziten Vorgaben aller Anforderungen, oder aber sie beschreiben ein Geschäftsmodell und überlassen dem Lieferanten, wie er es realisiert. In beiden Fällen ist es aus Gründen der Planbarkeit und Kostenkontrolle wichtig, dass die Anforderungen herausgearbeitet und dann als »Anhang« des Vertrags spezifiziert werden. Oftmals ist es für den Kunden ein hilfreicher Meilenstein, die Anforderungen formalisiert beschrieben zu sehen und nochmals prüfen zu können, ob sein Geschäftsmodell oder seine Vorschläge zusammenhängend und vollständig übernommen wurden. Für den Lieferanten ist es fast eine Lebensversicherung,

denn nun hat er eine Basis, von der aus er sein Projekt planen und Aufwände schätzen kann. Inwieweit die Anforderungsliste Vertragsbestandteil ist oder zur Preisabstimmung genutzt wird, hängt von der beiderseitigen Verhandlungsführung ab. Kunden, Vertrieb oder Marketing spezifizieren ihre Anforderungen, Wünsche und Geschäftsziele zunehmend informell und erwarten umgehend Lösungen. Zykluszeiten sind zu kurz geworden, um vom Kunden oder vom Marketing eine präzise Anforderungsliste vor Projektstart zu erwarten. Der Ausweg ist ein verständlicher Formalismus (z.B. Use Cases, strukturierte Vorlage), der inkrementell von impliziten Anforderungen zu einer Spezifikation entwickelt wird. Wir werden solche Beschreibungstechniken in Abschnitt 5.4 vorstellen und charakterisieren.

Testbare und entscheidbare Beschreibung der Anforderungen

Anforderungen, die beschrieben sind, können als Basis für weitere Entwicklungsschritte dienen. Wenn die Anforderungen bereits in der Spezifikation darauf geprüft werden, wie sie nachher zu testen sind, ist dies ein gutes Hilfsmittel, um die Verständlichkeit und Widerspruchsfreiheit zu verbessern. Ein Testexperte wird schwammige Beschreibungen nicht akzeptieren, denn es ist nachher kaum möglich, daraus verlässliche Testfälle abzuleiten. Entscheidbarkeit bei Anforderungen heißt, dass binär am späteren Produkt entschieden werden kann, ob die Anforderungen erfüllt wurden oder nicht. Das ist insbesondere für die Abnahme wichtig. Abnahmekriterien in die Anforderungen einzubetten und die Anforderungen daraufhin abzuklopfen, ob sie entscheidbar sind, trägt maßgeblich zur Kundenzufriedenheit bei. Nehmen Sie die Anforderung »Die Raumtemperatur muss im Tagesbetrieb den Sollwert auf $\pm 0,5$ Grad Celsius einhalten«. Das klingt bereits hinreichend präzise, ist es aber noch nicht. Versetzen Sie sich in die Rolle eines Testers. Wie würde er vor der Übergabe entscheiden, dass diese Anforderung erfüllt ist? Offensichtlich nur, indem er noch weitere Attribute hinzufügt, beispielsweise nach welcher Zeit diese Temperatur erreicht wird. Testbare und entscheidbare Anforderungen sind umfangreicher als eine vage Charakterisierung, wie sie aus der Ermittlungsphase resultiert. Diese Präzisierung der Anforderung trägt maßgeblich zum Projekterfolg bei.

Kontrollierte Konfigurationsbasis

Anforderungsspezifikationen sind die erste Konfigurationsbasis in jedem Projekt. Auf dieser Basis bauen viele weitere Dokumente und Arbeitsergebnisse auf. Daher müssen Anforderungen nicht nur archiviert werden, sondern auch versioniert. Nur damit ist klar zu erkennen, welche Basis konkret implementiert wurde. Mit dieser Konfigurationsbasis können Verknüpfungen zu anderen Dokumenten erstellt werden, um Konsistenz und Durchgängigkeit zu sichern

(z.B. von Anforderungen zu Testfällen). Die Anforderungsspezifikation ist der Startpunkt für alle Änderungen am Projektinhalt.



Beispiel:

Innerhalb meiner Beratungsprojekte werde ich immer wieder auf das Problem angesprochen, wie man mit diffusen Kundenanforderungen umgehen soll, wenn man selbst einen Prozess anwenden will, der klar strukturiert zwischen Lasten (funktionale Anforderungen, Qualitätsanforderungen) und Pflichten (Komponentenanforderungen, Entwurfslösungen) unterscheidet. Die Kundenanforderungen sind in der Regel eine Mixtur aus Anforderungen (d.h., was zu tun ist) und Entwurfsvorgaben (d. h., wie etwas realisiert werden soll, z. B. genaue Beschaltungsvorgaben, die 1:1 ins Schaltbild übernommen werden). In solchen Situationen sollte man zunächst prüfen, was die tatsächlichen Kundenanforderungen in der Außensicht, also die »Lasten«, sind. Was als Realisierungsvorgaben zusätzlich zu den Produkthanforderungen übrig bleibt, beispielsweise Kosten, Zeitpunkte und serviceorientierte Geschäftsmodelle, sind Randbedingungen. Sie sollten dann als Anforderungen spezifiziert werden und auch in der Analyse und im Angebot explizit berücksichtigt werden.

4.3 Dokumentation und Vorlagen

Individuelle Anforderungen müssen klar und deutlich beschrieben werden. Dies ist eine notwendige Bedingung dafür, dass das richtige Produkt entwickelt wird. Der Anforderungstext selbst sollte auch nochmals klar strukturiert werden, damit die individuellen Vorgaben gut verständlich bleiben. Abbildung 4-3 beschreibt eine einfache Vorlage für den Satzaufbau einer einzelnen Anforderung. Die einzelnen Anforderungen müssen kurz und knapp sein. Nur einige wenige verschiedene Hilfsverben (muss, soll, kann) kommen zum Einsatz. Damit sind die Anforderungen verständlich und besser testbar.

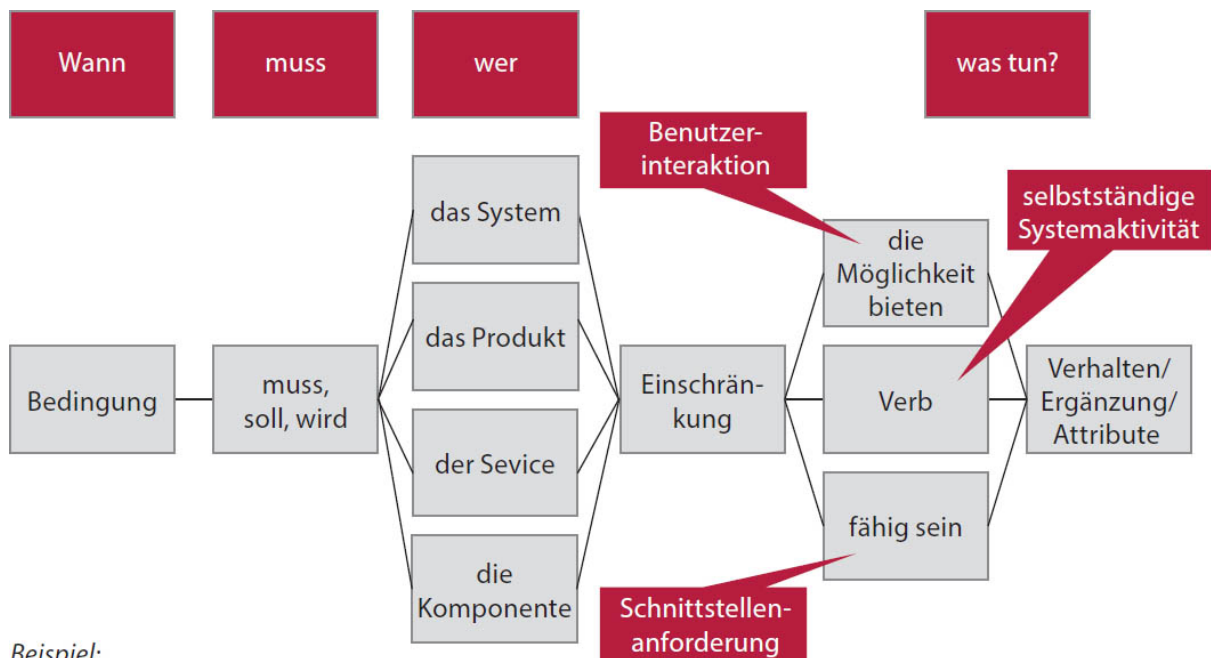


Beispiel:

Produktanforderung 1:

Wenn das Dashboard aktiviert ist, soll iHome die Parameter des aktuellen Raums zeigen.

Definierte Vorlagen und Satzschablonen helfen dabei, Anforderungen klar und rechtlich verbindlich auszudrücken, sie sauber zu strukturieren und sie testbar zu halten [ISO2011c, Robertson2012, Rupp2021]. Im Bestfall sind die Anforderungen selbst bereits als Geradeaus-Testfall spezifiziert, was mit Test-Driven Requirements Engineering (TDRE) zu einem sehr effizienten Arbeiten führt (siehe Kapitel 6) [Ebert2021c].



Beispiel:

Wenn das Dashboard aktiviert ist, soll iHome die Parameter des aktuellen Raums zeigen.

Abb. 4-3 Vorlage für die Beschreibung einer einzelnen Anforderung (Vorlage: www.vector.com/RE-Buch)

Tabelle 4-2 zeigt Mustertexte für verschiedene Arten von Anforderungen.

Anforderungstyp	Vorlage für den Text der Anforderung	Beispiel für den Text der Anforderung
Produktanforderung, Basis	Das <System> soll oder muss oder wird <Verhalten> <Verb>	Wenn die iHome-App aktiviert ist, soll iHome die Parameter des aktuellen Raums zeigen.
Produktanforderung, funktional, mit Schnittstelle	Das <System> soll oder muss oder wird <Person> <Verhalten> <Verb>	Bei Änderungen der Raumtemperatur soll die iHome-App die Änderung ansagen.

Produktanforderung, funktional, mit Schnittstelle, Randbedingung	Das <System> soll oder muss oder wird <Person> <Verhalten> <Randbedingung> <Verb>	Nach dem Start der iHome-App soll die Verbindung zu iHome hergestellt werden, wobei bei Störungen im WLAN automatisch auf Mobilfunk umgeschaltet wird.
Produktanforderung, Qualitätsanforderung, mit Schnittstelle, Randbedingung	Das <System> soll oder muss oder wird <Person> <Verhalten> <Randbedingung> <Verb>	Nach dem Start der iHome-App soll die Verbindung zu iHome innerhalb von 10 Sekunden hergestellt sein.
Produktanforderung, Qualitätsanforderung, mit Schnittstelle, Randbedingung	Der <Interesseneigner> soll oder muss oder wird in der Lage sein, <Eigenschaft> oder <Ziel> zu erreichen.	Der Betreiber soll neue Ansagen in 10 Minuten aufnehmen können.
Produktanforderung, Qualitätsanforderung, mit Schnittstelle, Randbedingung	Das <Prozessergebnis>/der <Prozess> soll oder muss oder wird <Fähigkeit> oder <Eigenschaft> haben.	Die Benutzerdokumentation soll in den Sprachen Deutsch, Englisch, Spanisch und Chinesisch vorliegen.
Produktanforderung, Qualitätsanforderung, mit Schnittstelle, Randbedingung	Das <Prozessergebnis>/der <Prozess> soll oder muss oder wird <Fähigkeit> oder <Eigenschaft> haben.	Die Projektorganisation soll aus einem Projektmanager, einem Produktmanager und einem Vertriebsbeauftragten bestehen.

Tab. 4-2 Mustertexte für verschiedene Arten von Anforderungen

Marktanforderungen werden im agilen Requirements Engineering (siehe auch Kap. 9) auch durch Epics und User Stories beschrieben.

Ein **Epic** beschreibt Marktanforderungen in ihren wesentlichen Ausprägungen. Epics erläutern fachliche Anforderungen aus Marktsicht und sind aus Geschäftszielen und der Vision abgeleitet. Epics werden häufig in User Stories oder auch Use Cases zerlegt und verfeinert. Beschrieben werden Epics in UML durch Use Cases und weitere passende Visualisierungen, beispielsweise Sequenzdiagramme. Der Begriff »Epic« wird vor allem im Umfeld des agilen Requirements Engineering verwendet. Epics dienen dabei zur Entwicklung eines Product Backlog im Rahmen von Scrum. Zunächst wird damit eine aggregierte, grobgranulare Sicht auf neue Produkthanforderungen entwickelt, ohne auf die Details einer Anforderung eingehen zu müssen.



Beispiel:

Epic 1:

Die Benutzerschnittstelle von iHome wird eine effiziente und diskriminierungsfreie Eingabe und Ausgabe bereitstellen.

Eine **User Story** detailliert Epics in einzelne Abläufe und Verhalten. Im Unterschied zum Use Case sind User Stories eher bruchstückhaft und zeigen nicht alle Aspekte auf, die zur Zielerreichung nötig sind. User Stories sind in Scrum sehr populär und substituieren vor allem die Use Cases, da der Begriff moderner ist, und man sich von etablierten Modellierungen abgrenzen will. Sie werden typischerweise informell beschrieben oder anhand einer einfachen Vorlage.



Beispiel:

User Story 1:

Als Benutzer von iHome möchte ich nach Drücken einer Taste der App sehen, in welchem Zustand mein aktueller Raum ist.

Der darunter liegende **Use Case** (deutsch: *Anwendungsfall, Benutzungsfall*) beschreibt, was das System tun soll, um einen Nutzen oder Wert zu schaffen. Damit ist der Use Case inhaltlich auf der gleichen Abstraktionsebene wie die Produkthanforderungen. Er beschreibt Akteure und ihr Verhalten gegenüber dem zu entwickelnden System (siehe auch Abschnitt 5.4). Use Cases sind in der UML standardisiert und unterstützen eine durchgängige Modellierung.

Qualitätsanforderungen (manchmal auch *nichtfunktionale Anforderungen* genannt) und Randbedingungen werden immer mit den nötigen Abnahmekriterien beschrieben.



Beispiel:

Qualitätsanforderung 1:

Der Benutzer der iHome-App erhält spätestens 5 Sekunden nach Drücken der Taste eine Reaktion des Systems.

Abnahmekriterium:

Test in der Simulationsumgebung für die Lastverhalten Nachtbetrieb, Standardbetrieb und Hochlastbetrieb. Drücken einer Taste. Messung der Zeit bis zum Erscheinen der Information.

Die Gesamtheit der Anforderungen wird in der Spezifikation strukturiert, um sie im Zusammenhang modellieren, analysieren, verfolgen und ändern zu können. Spezifikationen können schnell sehr umfangreich werden und brauchen daher eine klare Struktur und adäquate Werkzeugunterstützung. Beispielsweise umfasst die Anforderungsbeschreibung eines modernen Kombiinstruments im Kfz mehrere Hundert Seiten an Texten, Tabellen und Diagrammen. Ergänzt wird sie durch weitere Dokumente, wie unternehmenseigene Normen, Prüfprozeduren und Variantenbeschreibungen, die ein Mehrfaches dieses Umfangs haben.

Setzen Sie daher Vorlagen systematisch ein, um Ihren Projekten die Sicherheit zu geben, dass die Anforderungen klar und verständlich strukturiert werden. Vorlagen fördern die Disziplin während der Ermittlung und Spezifikation, denn sie geben eine Struktur vor, die eingehalten werden muss. Man sollte dieses Spezifikationsgerüst nicht für jedes Projekt neu erfinden.

Die wichtigste Vorlage für Anforderungsspezifikationen liefert ISO 29148 [ISO2011c]. Die Norm setzt sich aus einigen wenigen Kapiteln zusammen. Darauf aufbauende Vorlagen finden sich bei Vector [Ebert2014b] und Volere [Robertson2012].

Eine Vorlage sollte nicht nur das Gerüst anbieten, sondern auch eine Anleitung zum Ausfüllen. Abbildung 4-4 zeigt beispielhaft ein Lastenheft auf Basis der Vector-Vorlage. Es ist als Download erhältlich, um daraus eine eigene Vorlage zu erstellen. Sie können diese Vorlage als Kapitelstruktur für Ihre eigenen Bedürfnisse ergänzen und danach verbindlich für alle Projekte einführen.

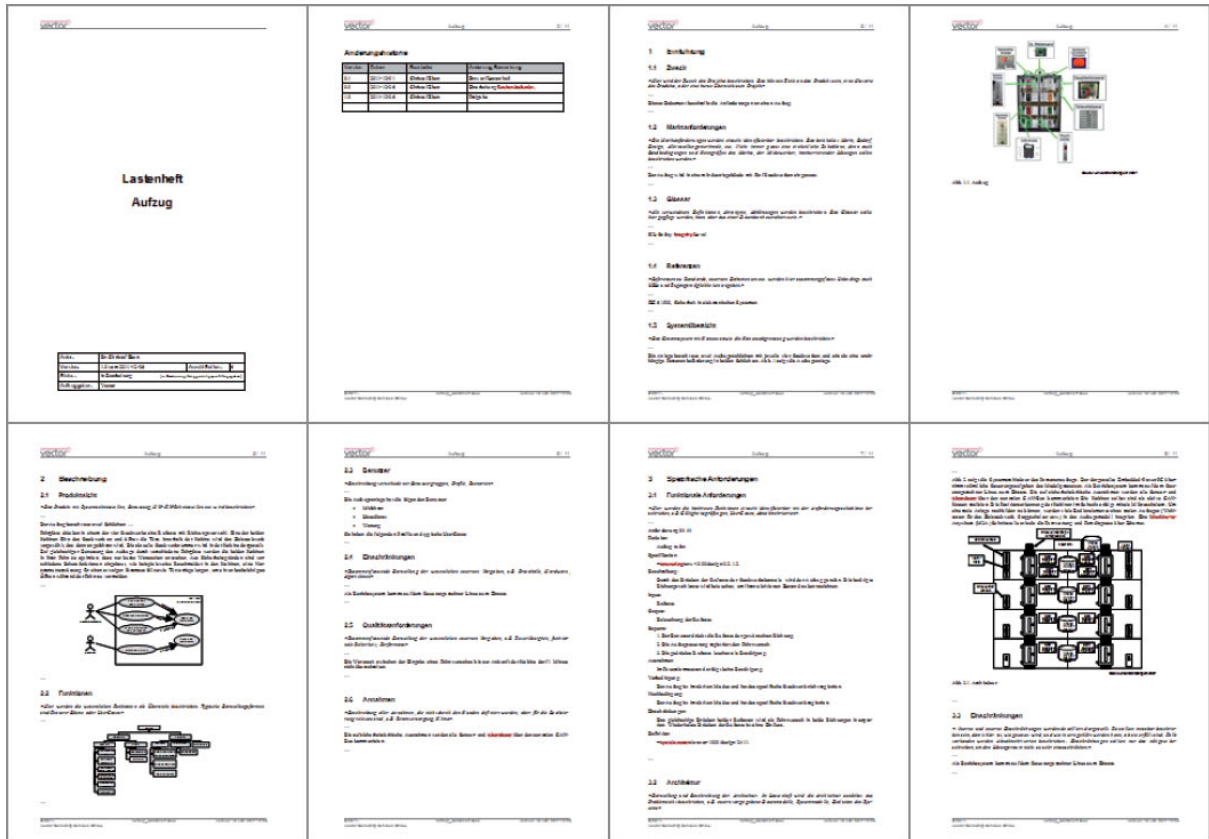


Abb. 4-4 Lastenheft mit der Vector-Vorlage (Vorlage: www.vector.com/RE-Buch)

Innerhalb der gegebenen Kapitelstruktur werden die Anforderungen einzeln so spezifiziert, wie in Abbildung 4-3 beschrieben. Wesentlich ist die frühzeitige Attributierung, um Metadaten zum Filtern zu setzen.



Beispiel:

Schlechte Formulierung einer Anforderung: »Der iHome-Aufzug soll nach Betätigung der Ruftaste umgehend das Stockwerk anfahren.« Als Anforderung ist der Text zu vage, zu unpräzise und zu abstrakt. Man erkennt die fehlerhafte Struktur sofort, wenn man versucht, einen Testfall dafür zu bilden. Was soll der Aufzug exakt machen? Welches Stockwerk soll er in Konfliktsituationen anfahren? Wie schnell ist »umgehend«? Welche Ruftaste? Vermeiden Sie die Vermischung verschiedener Anforderungen und bleiben Sie sehr konkret.

Verwenden Sie ausreichend Zeit und Energie, eine eigene Vorlage an Ihre Bedürfnisse anzupassen und sie als verbindlich zu kommunizieren. Vorlagen lassen sich sehr leicht mit einem Spreadsheet-Programm pilotieren (Abb. 4-5). Wir empfehlen Excel (oder ein anderes Tabellenprogramm), da es zu Struktur

und Systematik verpflichtet. Attribute lassen sich in einer Tabelle einfach filtern und verfolgen. Das macht später den Übergang zu einem professionellen Werkzeug einfacher.

ID	Prio	Status reached In					Source	Increment	WP	Customer Requirement	Functional specification
		Func Spec	Test Spec	Implemented	Validated	Released					
R0002	1	38	41	46	48	50	Kunden-LH	1	124	Der Benutzer stellt die Temperatur in der iHome-App ein.	Durch das Drücken von <Temperatur +/- Tastenfeld> wird die nach oben oder unten geänderte neue <Soll-Temperatur> angezeigt.

Inputs	Outputs	Constraints	Criteria of acceptance (Test case)	Novelty (1-4)	Complexity (1-4)	Functional Effort (1-10)	Effort (PD, Design)	Comment
Temperatur +/- Tastenfeld	Soll-Temperatur	Im Alarmzustand ist die Auswahl nicht aktiv.	<Temperatur +/- Tastenfeld> in Richtung unten kurz drücken. Die um 1 Grad Celsius reduzierte Temperatur ist sichtbar.	224			10 PD	Wird realisiert in der Panel App.

Abb. 4-5 Einfache Spreadsheet-Vorlage für Anforderungen (Vorlage: www.vector.com/RE-Buch)

Weitere Projektinformationen, wie Aufgaben, offene Punkte, Beschlüsse und Infos, finden ebenfalls im Spreadsheet Platz (Abb. 4-6). In größeren Projekten sind die Vorlagen umfangreicher, orientieren sich aber immer an der oben eingeführten Struktur.

No.	Date, creation	Source	Type	Content
	Initial date; frozen	Persons, not roles	Selection	Complete description of the agenda topic, action item, decision or information item
1	17.03.2016	ce	Agenda topic	Kick Off
2	17.03.2016	ce	Action item	Finalize Template for book
3	17.03.2016	ce	Meeting	Client workshop
4	22.03.2016	ce	Information	Circuits and logic will be provided by supplier
5	22.03.2016	ce	Decision	Software will be implemented in C language

Responsible	Status	Target date, initial	Target date, latest	Date, closed	Results, Remarks
Responsible for closing the action item	0 = open 1 = closed	initial target date to close	Latest target date to close	Completely finished/communicated	Information on status of action item, or details on progress, results
ce	1	17.03.2016		17.03.2016	
					Agreed in client meeting

Abb. 4-6 Agile Vorlage für offene Punkte, Beschlüsse und Infos (Vorlage: www.vector.com/RE-Buch)

Anforderungen sollten immer so kurz und prägnant wie möglich beschrieben werden. Erklärungen werden hinzugefügt, um die Anforderung und ihren Kontext zu verstehen. Anforderungen sollen sich auf Arbeitspakete

oder Inkremente im Projekt abbilden lassen. Jede Anforderung soll einen expliziten Nutzen beschreiben. Schließlich sollen Anforderungen auch psychologische Begrenzungen der Leser berücksichtigen, also beispielsweise die Lesbarkeit und Verständlichkeit maximieren und die Kompliziertheit reduzieren. Die Lesbarkeit kann durch eine Umfangsbegrenzung verbessert werden. Anforderungen sollten als Faustregel auf eine Druckoder Bildschirmseite passen. Viele Personen können Zusammenhänge, die auf einer einzigen Seite beschrieben sind, besser verstehen und gedanklich verarbeiten, als wenn es sich um mehrere Seiten handelt.



Beispiel:

In Kundenprojekten und vor allem in Altsystemen sehen wir häufig eine gewachsene, aber rudimentäre Struktur der Anforderungen. Vieles wird einfach in Word dokumentiert, ohne Systematik. Die Texte wachsen und verlieren an Übersichtlichkeit. Oft stecken sogar Vorlagen dahinter, die aber nicht konsequent genutzt werden. In der Folge wuchern die Anforderungen und können nicht gepflegt werden. Vor allem fehlt jegliche Verfolgbarkeit zwischen Lasten und Pflichten oder zu den Testfällen. Derart unzuverlässige Arbeitsergebnisse demotivieren die Mitarbeitenden, da man sich auf nichts verlassen kann.

Der Zusatzaufwand ist sehr viel höher, als wenn die Anforderungen systematisch gepflegt werden würden. Aber man packt es nicht an, da ständig Wichtiges im aktuellen Projekt zu tun ist.

Für solche Fälle haben wir eine Methodik entwickelt, um rasch die Anforderungen schrittweise dort zu verbessern, wo der Nutzen am größten ist. Das heißt, dass wir nicht »mit der Gießkanne« alles nacharbeiten, was immens Zeit kostet, sondern zielorientiert und fokussiert dort, wo aktuell kritische Änderungen sind oder wo Regressionstestfälle Sinn machen. Abbildung 4-7 zeigt die Methodik anhand eines konkreten Falls aus der Praxis.

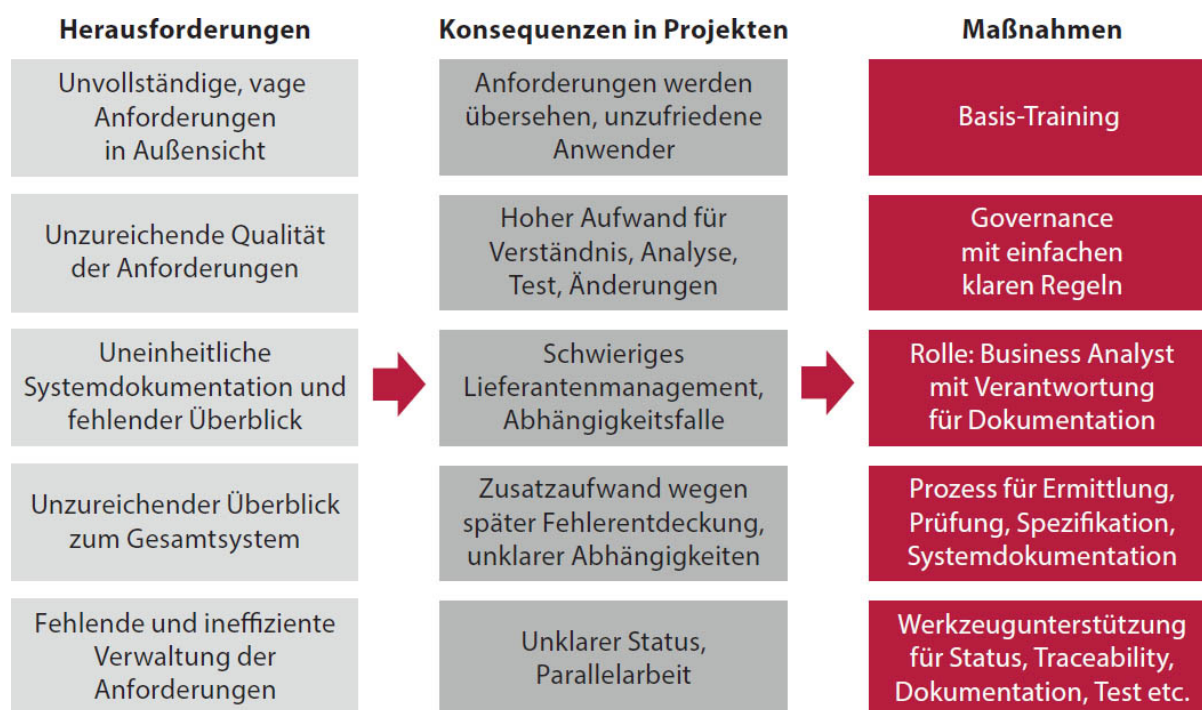


Abb. 4-7 Verbesserung der Dokumentation von Anforderungen

4.4 Struktur und Lesbarkeit

Spezifikationen müssen verständlich sein. Die natürliche Sprache ist vieldeutig und kann zu zahlreichen Missverständnissen führen. Das stört bereits im normalen Miteinander, wenn man sich missversteht. In Ausschreibungen und Verträgen ist es natürlich noch viel riskanter, denn durch Missverständnisse entstehen teure Fehler. Aus diesem Grund wurde frühzeitig die klare Differenzierung zwischen Lasten und Pflichten eingeführt. So ist sichtbar, was aus der Aufgabenstellung angekommen ist und umgesetzt wird.

Transformationsprozesse in der Kommunikation zwischen Darstellung und Wahrnehmung führen zu vielen Missverständnissen. Autoren transformieren von ihrer Wahrnehmung und ihren Konzepten zur Darstellung. Leser transformieren von der Darstellung zur eigenen Wahrnehmung. Wahrnehmung ist jedoch immer situativ und stark von der eigenen Erfahrung abhängig. So entstehen Abweichungen und damit Fehler. Diese Fehler pflanzen sich so als »stille Post« vom Benutzer zum Lastenheft, zum Pflichtenheft, zum Testfall etc. fort. Beispiele dazu sind:

- Nominalisierung (z.B. »Bei einem Systemabsturz ...«)
- Substantive ohne Bezugsindex (z.B. »die Daten«, besser »die Rechnungsdaten«)

- Universalquantoren (leichtfertiger Gebrauch von »immer«, »nie«, »jeder«, »alle« etc.)
- Unvollständig spezifizierte Bedingungen (z.B. »wenn ... dann« unzureichend spezifiziert)
- Unvollständig spezifizierte Prozesswörter (Aktiv-Formulierungen besser als Passiv-Formulierungen)

Eine formalisierte Spezifikation mit definierter Sprache schafft Klarheit. Sie erlaubt es, zu prüfen, ob die Anforderungen auch das wiedergeben, was die verschiedenen Stakeholder während der Ermittlung ausgesprochen hatten. Das reduziert oder eliminiert Widersprüche und Redundanzen. Das ist Handarbeit und muss immer wieder durch Reviews geprüft werden. Werkzeuge helfen dabei, die Anforderungen automatisch hinsichtlich verschiedener Qualitätsparameter zu prüfen, beispielsweise daraufhin, ob sie verständlich sind, wie ihre Lesbarkeit ist und ob wesentliche Attribute beschrieben sind. Unternehmen wie Mercedes prüfen so die eigenen Spezifikationen und erreichen einen gewaltigen Qualitätsschub und weniger Missverständnisse und Nachfragen während der Umsetzung. Die Anforderungsspezifikation ist eine Referenz für alle Stakeholder und Projektbeteiligten, was zu tun ist. Sie definiert eine einheitliche Terminologie und beschreibt alle projektrelevanten Inhalte und Randbedingungen in einem Dokument.

Einzelne Anforderungen können auf unterschiedliche Weise spezifiziert werden. Die Vorlage einer Spezifikation, die Sie im vorigen Kapitel kennengelernt haben, macht dazu keine Vorschriften. Wir wollen hier verschiedene Beschreibungsformen anhand ihrer jeweiligen Formalität und Präzision unterscheiden. In Kapitel 5 zeigen wir Modellierungstechniken.

Häufig gehen Zusammenhänge in einer unübersichtlichen Struktur der Anforderungen verloren. Vor allem bei Tabellen und Werkzeugen führt dies zu einer Betrachtung von Ausschnitten, und man riskiert, den Wald vor lauter Bäumen nicht mehr zu sehen. Abhilfe schafft dabei eine **gute Organisation der Anforderungen**:

- Klare Nummerierung der Anforderungen
- Änderungen und Beziehungen ständig pflegen und nicht abwarten
- Strukturierung anhand von Klassen und Gruppen
- Segmentierung gemäß Quellen und Funktionen
- Abhängigkeitsbeziehungen zu funktionalen Beschreibungen (z.B. Implementierungssicht, SW-Klassen, Komponenten)

- Attributierung zum Filtern, Sortieren etc.
- Metainformationen anhand definierter Begriffe aus dem Glossar

Die einfachste, aber auch missverständlichste Form einer Spezifikation ist informeller Text. Er ist leicht zu schreiben, denn es ist die Form, in der die Kunden oder Benutzer in aller Regel kommunizieren. Abbildung 4–8 zeigt eine solche informelle Beschreibung einer Anforderung. Offensichtlich ist Prosatext zur Beschreibung unzureichend, da Zusammenhänge nur sehr schwer herauszufinden sind. Selbst die Kapitelstruktur hilft nicht dabei, Querverweise praktikabel zu gestalten.

Beispiel

...

3.2. Benutzerfunktionen von iHome

iHome wird mit einem Control Panel gesteuert, das als App für Smartphone und Tablet verfügbar ist. Die typischen Parameter wie Licht und Temperatur werden dargestellt. Wenn das Dashboard aktiviert ist, soll iHome die Parameter des aktuellen Raums zeigen. Wenn der Button für Temperatur berührt wird, erscheint ein Display zum Steuern der Temperatur. Der Benutzer stellt die Temperatur mit einem Ziffernfeld in der App ein. Die neue Temperatur wird fett gedruckt, um ihren aktivierten Zustand zu kennzeichnen.

...

Abb. 4–8 Spezifikation in informellem unstrukturiertem Text

Die Verfolgbarkeit von Anforderungen zu Testfällen oder Entwurfsentscheidungen ist kaum möglich. Nachträgliche Änderungen sind mit großer Wahrscheinlichkeit inkonsistent mit Entwurfsdokumenten, denn die unscharfen Begriffe (z.B. »Button«) führen leicht dazu, dass sich die Entwickler ein eigenes Modell zurechtlegen, das nie mehr mit dem ursprünglichen und schwer lesbaren Text abgeglichen wird.

Strukturierte Spezifikationen mit expliziter Kapiteleinteilung verbessern die Lesbarkeit der Anforderungen. Textelemente werden strukturiert, um beispielsweise Begründungen für Anforderungen auffindbar zu machen oder um Querverweise zu ermöglichen. Abbildung 4–9 zeigt eine strukturierte Spezifikation einer Anforderung. Die Lesbarkeit hat sich im Vergleich zu Abbildung 4–8 verbessert, aber die Anforderung selbst ist noch immer sehr dicht. Dieses Format findet man in vielen Anforderungsbeschreibungen, mit dem Nachteil, dass eine Seite Anforderungstext nicht handhabbar ist.

Beispiel

...

3.2. Benutzerfunktionen von iHome

3.2.1. Wenn das Feld <Soll-Temperatur> berührt wird, erscheint das <Temperatur +/- Tastenfeld>. Durch das Drücken von <Temperatur +/- Tastenfeld> wird die nach oben oder unten geänderte neue <Soll-Temperatur> angezeigt.

Begründung: Der Benutzer muss unmittelbar erkennen können, dass iHome seine Änderungen angenommen hat.

Lösungsspezifikation: Kap. 5.3.

Systemarchitektur: Kap. 4.2.

Entwurf: ~/xyz/iHome/iHome-1000/design

...

Abb. 4-9 *Strukturierte Spezifikation*

Anforderungen werden auf ein Maß reduziert, in dem wesentliche Elemente einzeln identifizierbar und testbar sind. Damit ist eine explizite Verfolgbarkeit möglich, aber auch das Änderungsmanagement wird erleichtert. Abbildung 4-10 zeigt eine solche detaillierte strukturierte Spezifikation. Sie enthält bereits die strenge und klare Trennung der einzeln identifizierbaren Anforderungen. Abfolgen von Ereignissen werden aus Benutzersicht beschrieben.

Beispiel

...

3.2.1. Soll-Temperatur wählen

3.2.1.1. Wenn das Feld <Soll-Temperatur> berührt wird, erscheint das <Temperatur +/- Tastenfeld>. Durch das Drücken von <Temperatur +/- Tastenfeld> wird die nach oben oder unten geänderte neue <Soll-Temperatur> angezeigt.

3.2.1.2. Die Abfolge der Schritte ist wie folgt:

3.2.1.2.1. Der Benutzer berührt kurz das <Temperatur +/- Tastenfeld>.

3.2.1.2.2. Die iHome-App registriert die Änderung.

3.2.1.2.3. Die um ein Grad nach oben oder unten geänderte neue <Soll-Temperatur> wird angezeigt.

3.2.1.3. Die Eingabe einer Zahl außerhalb des Intervalls 10 bis 25 Grad Celsius wird nicht

akzeptiert.

3.2.1.4. Nach der erfolgreichen Eingabe wartet die iHome-App 1 Minute und gibt dann die Parameter an die Heizung weiter.

Begründung: Der Benutzer muss unmittelbar erkennen können, dass iHome seine Änderung angenommen hat.

Lösungsspezifikation: Kap. 5.3.

...

Abb. 4-10 *Strukturierte detaillierte Spezifikation*

In einer halbformalen Spezifikation (Abb. 4-11) wird eine definierte Vorlage eingesetzt, die die Anforderung intern stark strukturiert (z.B. Funktionalität, Informationsflüsse, Seiteneffekte, Bedingungen). Use Cases (Anwendungsfälle, Benutzungsfälle) haben eine solche Struktur. Sie bietet eine sehr gute Unterstützung für Validierung, Konsistenzprüfungen und automatisiertes Konfigurations- und Inhaltsmanagement.

Beispiel

...

Spezifikation: ~/iHome/anw-1000/design/3.2

Funktion: Rollladen steuern

Beschreibung: Das Rollladensteuergerät steuert den Rollladenmotor. Weicht die aktuelle Position des Rollladens von der gewünschten Position ab, aktiviert das Steuergerät den Motor, bis die Position erreicht ist. Wird dabei das Fenster geöffnet, wird zur Sicherheit der Motor abgeschaltet. Die Zielposition wird durch verschlüsselte Steuernachrichten vom Bedienfeld festgelegt.

Inputs: Steuernachricht, Rollladen,
 Positionssensor, Fenster Kontaktsensor

Outputs: Rollladenmotor

Sequenz: 1. Rollladensteuergerät empfängt eine
 Steuernachricht.
 2. Wenn diese korrekt entschlüsselt
 und decodiert wurde, übernimmt das
 Steuergerät die neue Zielposition
 und aktiviert den Rollladenmotor.

	3. Wenn die gewünschte Position erreicht ist, wird der Rollladenmotor abgeschaltet.
Ausnahmen:	Steuernachricht ungültig
Vorbedingung:	Bedienfeld und Rollladensteuergerät besitzen den gleichen, geheimen Schlüssel.
Nachbedingung:	-
Randbedingung:	Rollladen darf sich nicht bewegen, solange das Fenster geöffnet ist.
Definition:	~/xyz/iHome/iHome-1000/design/26.11.
...	

Abb. 4-11 *Halbformale Spezifikation*

Detaillierte Spezifikationen und halbformale Spezifikationen sind heute am gebräuchlichsten. Oftmals werden spezielle Vorlagen für eine halbformale Spezifikation eingesetzt, um die Inhalte automatisch weiterverarbeiten zu können. Die meisten kommerziellen RE-Werkzeuge legen eine solche Struktur nahe.

Formale Sprachen erlauben, Anforderungen und Lösungen ablauffähig und automatisch verifizierbar zu beschreiben und zu prüfen, beispielsweise auf Erreichbarkeit, Deadlocks, Widersprüche etc. Sie erlauben ablauffähige Simulationen und Prototypen, beispielsweise GUI mit Szenarien bzw. Matlab/Simulink für eingebettete Systeme.

Allerdings haben auch formale Spezifikationen nicht die gewünschte Fehlerfreiheit gebracht, da sie die Implementierung vorwegnehmen und das Risiko von Missverständnissen, Lücken etc. nur nach vorne verschieben (Abb. 4-12).

```

Beispiel
...
//Rollladensteuergerät Pseudocode
void loop() {
    if(packetReceived() )
    {
        decrypt() ;
    }
}

```

```

    parse() ;
    if(packet.isValid() )
        pos_target = packet.target() ;
    }
    motorSteuern() ;
}

```

Abb. 4-12 Formale Spezifikation

Zur Spezifikation empfehlen wir halbformale Notationen, die mit einem Werkzeug gepflegt und verwaltet werden. Einzelne Anforderungen müssen klar strukturiert sein. Großer Wert sollte auf eine klare Sprache gelegt werden. Die Grammatik kann eingeschränkt werden, um die Lesbarkeit und Prüfbarkeit zu verbessern. Beispielsweise werden Substantive als Daten, Verben als Aktionen, Adjektive und Attribute als Basis für Testfälle genutzt. Schlüsselwörter werden im Glossar spezifiziert und gezielt zur Filterung von bestimmten Inhalten eingesetzt. Die Verwaltung und Pflege solcher halbformalen Anforderungen kann bereits mit einem Spreadsheet umgesetzt werden.



Beispiel:

In einem Konsortialprojekt führten wir für einen Kunden ein durchgängiges Requirements Engineering ein. Unser Hauptaugenmerk lag auf einer einfach strukturierten Dokumentation aller Anforderungen und Testfälle, die zur Planung und Kontrolle der Lieferanten taugen sollte, aber nicht formal und schwer sein durfte. Hier zehn konkrete Hinweise zur Umsetzung:

1. Geben Sie für die Nummerierung eine exakte Spezifikation der Nummerierung vor. Anforderungen, Testfälle und Arbeitspakete im Projektmanagement sollten mit einem Kürzel vor der Nummer beginnen. Das Zahlenformat muss definiert sein, also z. B. Integer, 6 Stellen, führende Nullen werden geschrieben. Beispiel: REQ-123456, TST-123456, AP-123456.
2. Spezifizieren Sie, wie die Nummern bei Projektpartnern alloziert werden. Einerseits sollen die Nummern projektweit eindeutig sein, andererseits sollten sie dezentral verwaltet werden. Das wird

typischerweise über Nummernbänke gemacht, also z. B. REQ-10xxxx sind von Unternehmen 1, REQ-20xxxx sind von Unternehmen 2.

3. Führen Sie von den ersten Gesprächen im Projekt und Konsortium an ein zentrales Glossar, das für alle Projektpartner zugänglich ist und alle Begriffe, Akteure, Ereignisse, Arbeitsergebnisse etc. definiert. Beispielsweise sollten alle Akteure vor deren Erwähnung im Lastenheft oder in einem Use Case auch im Glossar definiert sein. Das erleichtert nachher die Konsistenzsicherung.
4. Pflegen Sie die Verfolgbarkeit ab der Spezifikation einzelner Szenarien, Business-Use-Cases und Testfälle. Dazu gehören Ziele und horizontale Querverweise auf andere Anforderungen der gleichen Abstraktionsebene sowie vertikale Querverweise zur späteren Implementierung.
5. Sehen Sie ein Feld für die Priorität vor. Sinnvoll sind 3 Prioritäten: hoch, mittel, niedrig. Die Prioritäten werden in der Analyse mit den Stakeholdern bewertet.
6. Sehen Sie ein Feld für die Aufwandsschätzung vor. Das ist für das spätere Projektmanagement unabdingbar. Geschätzt wird normalerweise auf Basis von Anforderungen.
7. Sehen Sie Möglichkeiten der Gruppierung und Hierarchisierung vor, wenn Sie bereits das Risiko einer Flut heterogener Anforderungen erkennen.
8. Definieren Sie klare Vorgaben an die erläuternden Detaildokumente in Word. Beispielsweise sollte das typische Volumen auf 1–2 Druckseiten begrenzt werden, Metadaten sollten auf die ID reduziert werden etc.
9. Definieren Sie zu Beginn der Spezifikation klare Regeln für das Versionieren der Anforderungstabelle sowie für das Änderungsmanagement. Dazu eignen sich verschiedene Werkzeuge. Bei Konsortien und verteilten Stakeholdern in verschiedenen Unternehmen bieten sich Cloud-Lösungen an.
10. Stellen Sie eine ausreichende Performance der Werkzeugunterstützung sicher. Wenn das Ändern eines Felds langsamer erfolgt, als man das von Excel gewohnt ist, geht die Zufriedenheit gegen null. Prüfen Sie das insbesondere auch bei einem abgesicherten VPN-Zugriff von außen.

In Zukunft wird es vermehrt anwendungsspezifische, teilweise formalisierte Spezifikationsumgebungen mit definierter Syntax und unterstützenden Werkzeugen geben, die in einem bestimmten Anwendungsbereich eingesetzt werden (z.B. Automobil- oder Flugzeugbau). Dafür werden spezielle Frameworks für die Modellierung und Bibliotheken zur Wiederverwendung angeboten. Zunehmend werden wir auch visuelle Prototyping-Ansätze sehen, um Spezifikationen schneller und effizienter zu erstellen. Diesen formalen Notationen ist gemeinsam, dass die Notation selbst »unsichtbar« wird und stattdessen Zusammenhänge und Modelle grafisch spezifiziert werden. Daraus wird die formale Beschreibung generiert, um später darauf aufbauend ablauffähigen Code zu erzeugen.

4.5 Attribute und Filter

Attribute werden eingesetzt, um die Anforderungen sortieren und filtern zu können. Attribute bilden die »Metainformationen« zu den Anforderungen, ähnlich wie dieses Buch Metainformationen zu Titel, Inhalt, Verlag etc. enthält, um in Suchmaschinen, beim Zitieren oder beim Buchhändler gefunden zu werden.

Ohne geeignete Attribute ist es nicht möglich, Anforderungen eine Struktur zu geben und sie in einen gemeinsamen Kontext zu bringen. Ohne gemeinsame Attribute ist weder der Projektfortschritt zu erkennen noch sind die Validierungsvorgaben nachzuvollziehen. Attribute werden während der Spezifikation eingeführt und zur Verwaltung der Anforderungen (z.B. Dokumentation, Verfolgbarkeit) genutzt (siehe auch Kap. 8). Spätestens bei der Projektkontrolle sind Attribute verpflichtend, um die Kontrolle zu behalten.

Anforderungen müssen strukturiert und verknüpft werden. Verfolgen Sie jene Attribute und Inhalte, die für Sie wichtig sind. Die typischen Attribute von Anforderungen sind in Tabelle 4–3 dargestellt:

Attribute	Beschreibung
Anforderungsnummer	Eindeutiger, referenzierbarer Schlüssel
Anforderungstitel	Kurz, aussagekräftig
Status	Status der Anforderung in der Umsetzung
Beschreibung	Kurz, präzise, verständlich auf Basis einer Vorlage mit Vor-/Nachbedingungen etc.: »wenn ... muss das System ...«

Quelle	Eigentümer der Anforderung
Referenzen	Standards, Normen, Kundendokumente, interne Dokumentation, Abnahmebedingungen
Erläuterung	Präzise, verständlich, Projektbezug; falls nötig, Abbildung mit Modellierung, Geschäftsprozess, Ablaufdiagramm
Randbedingungen	Anwendbare Randbedingungen und Qualitätsanforderungen
Nutzen	Externe Sicht, nicht pauschal, differenzieren
Priorität	Marktpriorität für agile Inkremente
Querbezüge	Betroffene oder bezogene Anforderungen, Abhängigkeiten
Einflüsse	Systemkomponenten, Funktionen, Hardware
Aufwand	Vorläufige Einflussanalyse, Kosten, make vs. buy
Akzeptanzkriterien	Testfälle, quantitative Vorgaben, Messbarkeit
...	
Kommentare	Ergänzungen während des Lebenslaufs der Anforderung

Tab. 4-3 *Attribute von Anforderungen gemäß ISO 29148*

4.6 Glossar

Das Glossar dient zur Dokumentation von Begriffen, Datenfeldern sowie zur Beschreibung der Datenverwendung (Abb. 4-13). Es listet alle Datenbezeichner auf, die im System oder in seiner Umgebung verwendet werden. Zu diesen Datenelementen werden die Beziehungen zwischen den Datenelementen sowie die Attribute der Datenelemente als »Single Source« zentral beschrieben – und gepflegt. Damit werden Inkonsistenzen oder Ungenauigkeiten vermieden.

Es existieren verschiedene Beschreibungsformen, wobei die verbale Form am häufigsten eingesetzt wird. Für formale Datenmodelle wird die BNF (Backus-Naur-Form) verwendet. Die meisten Werkzeuge des RE und der Anwendungsentwicklung oder Datenbankprogrammierung bieten ein Glossar bzw. Data Dictionary als grundlegendes Beschreibungsinstrument an.

Inhalte des Glossars

- Kontextspezifische Fachbegriffe

- Abkürzungen und Akronyme
- Alltägliche Begriffe, die im gegebenen Kontext eine spezifische Bedeutung haben
- Synonyme und Homonyme

Verwaltung und Nutzung des Glossars

- Es muss zentral verwaltet werden.
- Es muss projektbegleitend gepflegt werden.
- Es muss allgemein zugänglich sein.
- Es muss verbindlich verwendet werden.
- Es muss mit den Stakeholdern abgestimmt sein.
- Es muss eine einheitliche Struktur aufweisen.

Bezeichner	Informelle Beschreibung	Typ
Rollladen	Motorisierter Rollladen mit Sensoren	Entity
Positionssensor	Misst Rollladenposition	Entity
Rollladenmotor	Kann Rollladen anheben und absenken	Entity
...

Abb. 4-13 Glossar oder Data Dictionary

4.7 Checkliste für die Dokumentation

Die Anforderungsspezifikation sollte bereits während ihrer Entstehung daraufhin untersucht werden, dass sie gewisse Mindestanforderungen erfüllt. Hier ist eine kurze Checkliste, die solche Anforderungen an eine Spezifikation beschreibt.

- Beschreibt die Spezifikation ein explizites Ziel, das durch das Produkt erfüllt wird? Ist dieses Ziel konsistent mit der ursprünglichen Vision?
- Ist der Anwendungsbereich des Produkts ausreichend präzise beschrieben?
- Wird auch beschrieben, was das System nicht erfüllen muss?
- Wurden die Bedürfnisse verschiedener Stakeholder berücksichtigt?
- Trennt die Spezifikation klar zwischen Marktanforderungen (d.h. Kundenwünschen) und Produkt- bzw. Komponentenanforderungen

(Realisierung, Abbildung auf das spätere Produkt)?

- Trennt die Spezifikation klar zwischen Lastenheft (was muss getan werden) und Pflichtenheft (wie wird es getan)?
- Existiert ein ausreichend genaues Glossar?
- Sind verpflichtende Marktanforderungen ausreichend berücksichtigt (z.B. Gesetze, Standards, Vorschriften, wirtschaftliche Erfordernisse, physikalische Randbedingungen)?
- Wurde unnötige Komplexität vermieden? Gibt es einfachere Lösungen für die gleiche Frage? Lassen sich Verzierungen streichen?
- Sind die relevanten Qualitätsanforderungen konkret und messbar beschrieben?
- Wird der umgebende Geschäftsprozess bei IT-Systemen bzw. die Umgebung im Systemkontext (bei eingebetteten Systemen) hinreichend beschrieben?
- Sind die Schnittstellen zu anderen Systemen ausreichend dokumentiert?
- Ist die Ablösung eines etwaigen Vorgängersystems beschrieben?
- Sind Interoperabilitätsszenarien beschrieben?
- Wurden die richtigen Techniken zur Spezifikation eingesetzt?
- Ist die Dokumentation bei späteren Änderungen und Erweiterungen wartbar?
- Ist die Dokumentation lesbar und verständlich (Lesbarkeitsanalyse durchführen)?

4.8 Tipps für die Praxis

- Beschreiben Sie Produktvision und Projektauftrag zu Beginn der Spezifikation. Es hilft, wenn die Beteiligten verstehen, was sie eigentlich machen und was es bringt.
- Strukturieren Sie alle Anforderungen. Anforderungen werden individuell spezifiziert, referenziert und modelliert. Vermeiden Sie, dass mehrere Anforderungen in einem Satz beschrieben werden. Nur so lassen sich Anforderungen nachvollziehbar umsetzen, prüfen und ändern.
- Setzen Sie verständliche Sprache konsistent ein. Verwenden Sie in der Dokumentation kurze Sätze und verständliche Sprache.
- Fragen Sie in der Ermittlung direkt nach, wenn Sie etwas nicht verstanden haben.
- Nutzen Sie ausschließlich die Verben »sollen«, »müssen« und »werden«. Optionale Anforderungen müssen klar unterschieden werden (z.B. durch Prioritäten). Vermeiden Sie kryptischen Computerjargon und Mehrdeutigkeiten.

- Halten Sie jede Anforderung testbar und messbar. Anforderungen sind Vertragsbestandteile und sollten daher systematisch prüfbar sein. Alle relevanten Anforderungen werden dokumentiert. Implizite Anforderungen sind entweder als wiederverwendete Standards oder Richtlinien referenziert, oder sie werden weggelassen.
- Führen Sie ab Beginn der Anforderungsermittlung ein Glossar oder Wörterbuch für alle Fachwörter und Abkürzungen. Spezifizieren Sie die einzusetzenden Daten.
- Strukturieren Sie den Text mit Layout- und optischen Hilfsmitteln, beispielsweise Kapitelnummern im Standardformat, Markierungen, Unterstreichungen, Referenzen. Vermeiden Sie Farben und Farbcodes, die beim Kopieren verschwinden oder von manchen Menschen nicht wahrgenommen werden können.
- Nutzen Sie passende Methoden und Notationen für Spezifikation und Modellierung. Verwenden Sie für die Anforderungen geeignete standardisierte Vorlagen. Beispielsweise sollte ein GUI mit einem Prototyp beschrieben werden und nicht durch einen Use Case.
- Achten Sie bei der Spezifikation auf Attribute wie rechtlich verbindliche Normen, Randbedingungen etc. Verweisen Sie auf die exakten Quellen, da sich Normen, Standards und Gesetze ändern.
- Spezifizieren Sie Anforderungen mit einem Werkzeug. Die Anforderungsspezifikation ist ein Dokument, das für alle Stakeholder zentral und versioniert vereinbart, was im Projekt zu tun ist. Geben Sie Dokumente in einem robusten Standard wie PDF oder ReqIF weiter. Damit können Sie Leserechte, Kopierbarkeit und Weiterverteilung sehr gut kontrollieren, während gleichzeitig eine definierte Qualität beim Druck erreicht wird.
- Pflegen Sie Änderungsstände nicht in der Dokumentation, sondern mit einem speziellen Konfigurationswerkzeug, in dem komplette Versionsstände archiviert werden und die Änderungen per Change Request oder Änderungsbeschreibung beschrieben sind.
- Achten Sie beim Austausch von Dokumenten darauf, dass sie keine vertraulichen Informationen enthalten. Beispielsweise steckt in Office-Dokumenten oftmals eine Menge von Metadaten (z.B. Autorennamen, Unternehmensdaten) und Änderungshistorien, die nicht in fremde Hände gehören. Nutzen Sie das ReqIF-Schnittstellenformat für den Austausch.

4.9 Fragen und Impulse

- Entwickeln Sie eine Vorlage für Anforderungen. Geben Sie eine Struktur vor (z.B. Nummer, Titel, Status, Kundensicht vs. Lösungssicht). Wie viel Sichtbarkeit und Struktur brauchen Sie an welchen Stellen?
- Wie sehen Ihre Anforderungen aus? Sind sie sauber spezifiziert? Folgen Sie einer definierten Vorlage, wie Ihre Spezifikation auszusehen hat?

- Wer spezifiziert Ihre Anforderungen? Wer prüft sie? Werden alle Anforderungen formal geprüft? Falls nicht, weshalb nicht?
- Haben Ihre Anforderungen die richtige Qualität? Wie könnten Sie deren Qualität verbessern?
- Ändern sich Ihre Anforderungen häufig? Können Sie diese Änderungen auf bestimmte Auslöser zurückführen? Sind es immer wieder die gleichen Auslöser? Kommen Sie manches Mal in die Situation, dass Anforderungen unsauber spezifiziert werden, da sie sich sowieso noch einmal ändern? Wie lässt sich dieser Teufelskreis durchbrechen?
- Trennen Sie in der Spezifikation klar zwischen Lastenheft und Pflichtenheft?
- Wie entwickeln Sie die Architektur aus den Anforderungen?
- Übung »Vernetztes Not-Aus«: Der Kunde wünscht sich die Anbindung einer Notaus-Schaltung für eine verteilte Industrieanlage. Die Information zum Abschalten soll über einen vorhandenen Industriebus gesendet werden. Definieren Sie ein Szenario für System und Kontext. Ermitteln Sie wenige wesentliche Anforderungen an das System (Markt/Produkt). Erstellen Sie eine initiale (Top-Level-)Systemarchitektur mit einzelnen Systemkomponenten. Spezifizieren Sie 1–2 Marktanforderungen an das System sowie 2–3 damit verknüpfte Anforderungen an eine Komponente.

Index

A

- Abdeckungsanalyse 245
- Abhängigkeiten 183
- Ablaufmodell 73
- Ablauforganisation 323
- Abnahme 191
- Abnahmekriterien 182, 191
- Abstimmung 202
- Abuse Case 93, 143, 182
- ACE 284
- ACE-Framework 286
- ACES 395
- AGB 232
- Agil 40, 257, 283
 - Transformation 284
- Agile Entwicklung 258, 265, 276, 299, 404
- Agile für Critical Engineering *siehe ACE-Framework*
- Agile Schätzung 159, 166, 276
- Agile Skalierung 286
- Agiles Manifest 267
- Agiles Projektmanagement 331
- Agiles Requirements Engineering 67, 265, 276, 404
- Agilität, skalierbare 283, 286
- Aktivitätsdiagramm 157
- Alleinstellungsmerkmal 10, 68

Allocation before Commitment 391
ALM 301, 308
Altsystem 249
Ampel 210
Analogie 159
Analyse 76, 123, 125
Analysemethode 126, 129, 137
Analyst *siehe Requirements-Ingenieur*
Änderbarkeit 82
Änderung 249, 392
Änderungsmanagement 39, 237
Änderungsrate 39, 73, 237, 255
Anforderung 21
 Attribute 118, 137
 Delta-Anforderung 249
 Excel-Vorlage 109
 Fehler 175
 Maße 253
 Satzschablone 104
 Verfolgbarkeit 137
 Vorlage 104, 109
 Word-Vorlage 109
Anforderungsänderung 393
Anforderungsanzahl 254
Anforderungsermittlung 65–66
Anforderungsingenieur *siehe Requirements-Ingenieur*
Anforderungsmodell 126, 137
Anforderungsqualität 194
Anforderungsspezifikation 102
Anforderungsstatus 255
Anwendungsfall 31, 167
Anwendungslebenszyklus-Management *siehe ALM*
Apple 2, 23
Application Lifecycle Management *siehe ALM*
Äquivalenzklassen 193

Arbeitsergebnis 101
Architektur 134, 214
Attribute 118, 137
Attributierung 108, 251
 Aufbauorganisation 323, 328
Aufgabenbeschreibung 368
Auftraggeber 61
Aufwand 10, 243, 292
Aufwandsschätzung 159
Ausnahme 168
Ausnahmefälle 77
Autonomes System 77
Autonomy, Convergence, Ecology, Services *siehe ACES*

B

Backlog 310, 330
Backus-Naur-Form 119
Basisfaktoren 55, 79
Bedrohungsmodell 93
Bedrohungsszenario 92
Begeisterungsfaktoren 55, 79
Beispiel 43
Benchmarking 10, 166, 282, 390
Benutzbarkeit 81, 87, 313
Benutzer 22, 56, 61
Benutzeranforderung 24
Benutzerdokumentation 184
Benutzerschnittstelle 73
Beratung 346
Beschaffung 341
Besprechungen 377
Beste Praxis 383
Betrieb 393
Bewertbarkeit 198
Beziehung 139
BGB 222

Big Data 396
Blackbox-Tests 189–190
BPMN 137, 140
Brainstorming 73
Bugzilla 300
Burn-Down Chart 245
Business Analyst *siehe Requirements-Ingenieur*
Business Canvas 282
Business Case 6, 335
Business Model Canvas 282
Business Process Model and Notation *siehe BPMN*

C

Capability Maturity Model Integration *siehe CMMI*
CCB 65
Certified Professional for Requirements Engineering *siehe CPRE*
Change Control Board 65
Change Control Board *siehe CCB*
Change Management 311, 356
Checkliste
 Analyse 170
 Definition of Done 170
 Dokumentation 120
 Ermittlung 95
 Prüfung 194
 Verträge 230
 Verwaltung 261
 Werkzeuge 313
Claim Management 225, 239
Cloud-Lösungen 302
CMMI 386
Cockpit-Chart 260
COCOMO-Verfahren 160
Commercial off-the-shelf *siehe COTS*
Commitment 227
Community-Source-Software 398

Compliance 216
Concurrent Engineering 290
Confuse Case 143, 182
Continuous Integration 258, 277
Conway's Law 33
Corner Cases 77
COTS 341
CPRE 371
CRC-Karte 155
Cybersecurity *siehe Security*

D

DAD 284, 286
Dashboard 253, 258, 260
Data-Dictionary 119, 152, 275
Datenflussmodell 150
Definition of Done 171
Dekomposition 167
Delta-Anforderung 249
Demand Management 239
Demografische Analyse 73
Design for Change 209, 398
Design Thinking 241, 282, 286
Design to Cost 127, 398
Dienst 41, 129, 347, 398, 401
Dienstanforderung 129, 347
Dienstleistung 347, 398
Dienstleistungsvertrag 343
Dienstqualität 347
Differenzierung 398
DIN 69901 388
Disciplined Agile Delivery *siehe DAD*
Dokument 101
Dokumentation 99, 184
DOORS 301, 307, 353
Dopler 353

Dumpingangebot 223

E

Earned Value 245–246, 257, 292, 333

Earned-Value-Analyse 245, 257

Eclipse 301

Effizienz 82, 269

Effizienzsteigerung 404

Eindeutigkeit 198

Einflussanalyse 245

Einkauf 341, 346

Einschränkung *siehe Randbedingung*

Elevator Pitch 282

Enterprise Architect 301, 353

Entity Relationship Attribute *siehe ERA*

Entity Relationship Model *siehe ERM*

Entscheider 56

Entscheidung 79

Entscheidungstabellen 157, 168

Entwicklungsprozess 325

Entwurf 134

Epic 106

ERA 152

ERM 152

Ermittlungstechniken 65

Erwartete Funktion 77

Escrow 233

Ethnografische Analyse 73

Evolution 325, 393

Experiment 73

Explorativ 40

Extreme Programming *siehe XP*

E4-Messprozess 253

F

Failure Mode and Effect Analysis *siehe FMEA*

Failure Mode, Effect and Diagnostics Analysis *siehe FMEDA*

Fault Tree Analysis *siehe FTA*

Feature-Driven Development 270

Feature-Modell *siehe Funktionsmodell*

Fehler 10, 175, 185, 255

Fehlerarten 255

Fehlerbaumanalyse 189

Fehlerzahlen 255

Fibonacci-Folge 159

Flesch-Kincaid 183, 256

FMEA 73, 87, 93, 182, 189

FMEDA 13, 87

Fokusgruppen 73, 282

Ford 23

Fortschritt 253

Fragebogen 73

Freigabekriterien 191

Frontloading 13, 324

FTA 13, 87

Full Function Points 160, 163

Function Points 160, 163

Funktion 31

Funktionale Anforderung 30

Funktionale Dekomposition 147

Funktionalität 81

Funktionskorrelation 77

Funktionsmodell 147, 149

Funktionsperspektive 128

Funktionspunkte 150, 160, 163

G

Gartner 4

Gebrauchstauglichkeit 219–220

Gefährdung 77

Gefahrenanalyse 189

Geschäftsanforderung 24

Geschäftsleitung 63
Geschäftsmodelle 401, 404
Geschäftsnutzen 194
Geschäftsprozess 31, 140, 316, 347
Geschäftsvorfall 31
Gesetze 33, 216
Glossar 119, 152, 167, 179, 275, 413
Green Coding 397
Green IT 397
Gruppenarbeit 73
Gut genug 275

H

Haftungsfragen 222
Halbformale Spezifikation 114
Hardware 237
Harvard-Prinzip 380
Hierarchische Verfeinerung 147
Horizontale Verfolgbarkeit 246

I

IBM 307
IDE 301
Identifizierbarkeit 256
IEEE-Standard 830 389
iHome 43
Implizite Anforderung 101
Industrie 4.0 396
Industriestandards 34
Informationssicherheit 89
Informationsvernetzung 396
Inkrement 277, 288
Inkrementelle Entwicklung 211, 277
Innovation 2, 72, 339, 394
Innovationsmanagement 398
in-Step 301
Integrated Development Environment *siehe IDE*

Integration 135
Integrity 301, 308, 353
Intelligent Reuse 147
Intelligente Wiederverwendung 147
Interesseneigner *siehe Stakeholder*
Interessenkonflikt 56
Interessensphäre 56
International Organization für Standardization *siehe ISO*
International Requirements Engineering Board *siehe IREB*
Internet of Things *siehe IoT*
Internet of Thinking 399
Internetressourcen 411
Interview 73
IoT 399
IREB 371
ISO 385
ISO-Standard 12207 388
ISO-Standard 15288 387
ISO-Standard 19501 137
ISO-Standard 24766 303, 389
ISO-Standard 25010 389
ISO-Standard 25030 81
ISO-Standard 29148 107, 385, 389
ISO-Standard 33000 386
ISO-Standard 9001 388
ISO-Standard 9126 81
IT-Plattformen 317
IT-Recht 216

J

Jira 300
Jobs, Steve 23, 37
Joint Application Development 73

K

Kanban 269–270

Kano-Modell 37, 55, 72, 79, 276
Kaufvertrag 219, 343
Kennzahlen 10, 253, 390
Kernteam 63, 210
Key Account Manager 56
Klarheit 183, 194, 256
Klasse 153
Klassendiagramm 155
Klassenmodell 169
Kognitives Verfahren 73
Kollaboration 286, 298, 300, 316
Kommunikation 111, 374, 380
Kompetenz 363
Kompetenzprofil 368
Komplex 273
Komplexität 2, 147, 273
Kompliziert 273
Komponenten 341
Komponentenanforderung 24
Konfiguration 40
Konfigurationsbasis 65, 103
Konfigurationsmanagement 239, 355
Konfigurator 147
Konflikt 75, 78, 380
Konformität 82
Konsistenz 197
Kontext 69
Kontextbewertung 73
Kontextdiagramm 167
Kontextgrenze 69
Kontextmodell 139
Kontinuierliche Integration 258, 277
Kontinuierliche Verbesserung 270
Konzept 41, 73
Korrektheit 197

Kostenreduzierung 404
Kostenschätzung 161
KPI 253
Krisenmanagement 228
Kunde 22, 25, 56, 61, 230
Kundenorientierung 37, 404
Kundenumfrage 394

L

Large-Scale Scrum *siehe* LeSS
Lastenheft 43, 101–102
Lean 265
Lean Development 265, 290
Lean Requirements Engineering 276, 404
Lebenszyklus 239, 323
 Management 402
Leistungsfaktoren 55, 79
Leistungsvereinbarung 347
Leitungsgruppe 63
Lesbarkeit 183, 256
LeSS 284, 286
Lieferanten 56, 229, 343
Lieferantenbewertung 343
Lieferantenmanagement 166, 172, 208, 225, 232, 341, 399
Linguistische Analyse 183, 256
Lizenzvertrag 229
Lösung 21, 404
Lösungsmodell 123, 126, 137
Lösungsraum 23, 69
Lösungsspezifikation 102

M

Machbarkeit 136
Magic Quadrant 306
Magisches Dreieck 266, 394
Marketing 25, 52, 62, 395, 401
Marktanforderung 24

Markteintritt 325
Marktforschung 394
Marktregeln 401
Marktstudie 73
Maße 10, 253, 390
Medizintechnik 290
Metadaten 108
Metapher 287
Methode 37, 41, 124, 132, 153
Migration 249
Migrationsprojekt 2, 249
Minimum Viable Product *siehe MVP*
Missbrauch 182
Missbrauchsszenarien 168
Misuse Case 93, 143, 168, 182
Mock-up 301
Modell 41, 125, 129, 182
Modellbasiertes Requirements Engineering 126, 137
Modellierung 76, 125
Modellierungsgrad 255
Modifizierbarkeit 199
Moqups 301
MVP 266
Myers-Briggs-Indikator 377
Mylyn 300

N

Nachforderungen 225
Nachforderungsmanagement 239
Nachhaltigkeit 397
NDA 232
Negative Anforderung 74, 143, 168, 189
Netscape 326
Neues Normal 374, 383, 394
Nexus 284
Nichtfunktionale Anforderung *siehe Qualitätsanforderung*

Non Disclosure Agreement *siehe NDA*

Normen 33

Notation 41

Nutzen 257, 395

Nutzenanalyse 245

O

Object Management Group *siehe OMG*

Objekt 153

Objektmodelle 153

Objektorientierte Analyse 73

Objektorientierung 153

Oder-Dekomposition 167

Office-Werkzeug 299

Ökologie 397

Ökosystem 398–399

OMG 137

Organisation 323, 328

Orthogonalitäts-Check 184

P

Papyrus 301

Pareto-Prinzip 390

Pecha Kucha 282

Penetrationstest 93

Performance 165

Persona 73, 282

Perspektive 56

Pflege 343

Pflichtenheft 43, 101–102

Pflichtverletzung 222

Plan 159

Planning Game 166, 276

Planung 13, 68

Planungspoker 166, 276

Planungsspiel 288

Plattform 147

PLE 147
PLM 298, 301
PMO 331
Polarion Requirements 310, 330
Portfoliomanagement 337
Portierbarkeit 82
Positionsbeschreibung 368
Praxis 43
PREvision 301, 310, 353
Prinzipien 41
Priorisierung 93, 211, 214, 332, 354
Problemraum 23
Product Owner 330
Produkt 22, 41, 323
Produktanforderung 24, 26
Produktivität 159, 165, 393
Produktlebenszyklus-Management *siehe PLM*
Produktlinie 147, 291
Produktlinienentwicklung *siehe PLE*
Produktmanagement 335, 404
Produktmanager 62, 66–67, 330
Produktplanung 354
Produktroadmap 215
Produktvision 13, 68–69, 194
Prognose 394
Programmiermakro 318
Project Management Office *siehe PMO*
Projekt 253, 323, 331
Projekterfolg 4
Projektfortschritt 161, 333
Projektkernteam 63
Projektmanagement 36, 63, 205, 253, 331
 Referenzprozess 325
Projektmanager 6, 61, 66, 220, 330
Projektphasen 331

Projektplan 209, 215
Projektreview 210
Projektstatus 260
Projektumfang 69
Projektziel 53
Protokollanalyse 73
Prototyping 73, 184, 241, 282, 301
Prozess 40–41, 153, 312, 325, 356, 385
Prozessfähigkeit 34, 161
Prüfbarkeit 103, 177, 185, 188, 199
Prüfung 76, 175–176, 189, 194
Puffer 211
Pure Variants 353
Putnam-Formel 165

Q

QFD 73, 183, 213
Qualifikationstest 190
Qualität 10, 175, 194, 253, 405
Qualitätsanforderung 30, 32, 81, 92, 169, 193, 405
Qualitätskontrolle 175
Qualitätskriterien 177
Qualitätssicherung 64
Quality Function Deployment *siehe* QFD
QualityCenter 301

R

RACE 273
RAMSS 32
Randbedingung 30, 33, 78, 169, 188, 193
RCDA-Prinzip 227
Realisierbarkeit 199
Rechteverwaltung 318
Rechtliche Verbindlichkeit 66
Rechtsmangel 221
Redesign 290
Redmine 300

Reduce Accidents, Control Essence *siehe RACE*

Refactoring 249, 286

Referenzprozess 40, 325, 331

Reflexion 136

Reframing 282

Regressforderung 220

Release 337, 354

Releaseplanung 215, 277, 288, 337

Relevanz 195

Reliability, Availability, Modifiability, Safety, Security *siehe RAMSS*

Reporterzeugung 317

ReqIF 301, 318, 390

Requirements

- Änderungsrate 392
- Rechtsprechung 220
- Test 185

Requirements Engineering

- agiles 40, 265, 276, 404
- Arbeitsergebnisse 101
- Aufwand 10
- Ausblick 394
- Benchmarking 10, 390
- Daumenregeln 10, 390
- Definition 21, 34, 36
- Dienst, Service 347
- Einführung 21, 356
- Glossar 413
- Kennzahlen 10, 390
- Konfiguration 40
- Kosten 392
- Lean 276, 404
- Nutzen 10
- Organisation 323, 328
- Prozess 356
- Referenzprozess 331

Risiken 6
ROI 10
Selbsttest 16
Stand der Technik 383
Werkzeuge 298, 301, 303, 405
Requirements-Ingenieur 63, 367–368
 Aufgabenbeschreibung 368
 Kompetenzprofil 368
 Stellenausschreibung 368
Requirements Interchange Format *siehe RIF, ReqIF*
Return on Investment *siehe ROI*
Reverse Auctioning 225
Review 292
RIF *siehe ReqIF*
Risiko 6, 161
Risikomanagement 4, 166, 175, 205, 224, 239, 275, 331, 343
Roadmap 215, 277, 337, 354
ROI 10, 291, 335
Rollen 56, 66
Rollenspiele 73

S

Sachmangel 219
SAFe 267, 284, 286
Scaled Agile Framework *siehe SAFe*
Schadenersatzpflicht 222
Schätzung 159, 171
 agile 166
 Analogie 160
 Einflussfaktoren 162
 Kosten 159
 Referenzen 160
 Zeit 159
Schätzwerkzeuge 160
Schlüsselperson 57
Schnittstellen 69, 140

Scrum 270, 286, 291, 330
Scrum of Scrums *siehe SoS*
Security 42, 77, 81, 89, 92–93, 318, 396
Security Requirements Engineering 92
Selbsttest 16
Sequenzdiagramm 144
Service Level Agreement *siehe SLA*
Service *siehe Dienst*
Serviceorientierung 401
Sicherheit 87, 396

T

Siemens 292
Simulation 73
Skalierbare Agilität 283, 286
SLA 83, 233, 347–348
SlimControl 160
Soft Skills 374, 380
Software-as-a-Service 398
Softwarekomponenten 341
Sonnenschein-Szenario 141
SoS 284, 286
SOTIF 77, 87
Soziale Kompetenz 374
Spezifikation 43, 99, 112
SPICE 386
Spreadsheet 299
Stakeholder 2, 56, 61, 75
Stakeholder-Analyse 56
Stand der Technik 34, 383, 385
Standards 33, 385, 411
Standish Group 4
Status 260
Stellenausschreibung 368
Steuerkreis 61, 63
Strategie 324

Strukturbruch 28, 128
Strukturierte Analyse 73
Strukturierte Spezifikation 113
Strukturperspektive 128
SysML 137
System 69, 132
Systemanalyse 125
Systemanalytiker *siehe Requirements-Ingenieur*
Systemdenken 132
Systementwicklung 150
Systemkontext 69
Systemmodell 125
Systems Engineering 132
Systemschnittstellen 69
Systemtechnik 34
Systemtest 190
Systemumgebung 69
Szenario 73, 141, 167

T

TARA 92, 94
TDD 185, 270
TDRE 104, 168, 185, 270
Team 399
Technische Schulden 249, 273
Technologiemanagement 172, 339
Template *siehe Vorlage*
Terminschätzung 161
Termintreue 10
Tesla 2
Test 135–136, 185, 243
Testabdeckung 190
Testbarkeit 168, 188
Test-Driven Development *siehe TDD*
Test-Driven Requirements Engineering *siehe TDRE*
Testende-Kriterien 191

Tester 64, 185
Threat and Risk Analysis *siehe TARA*
Timeboxing 277
Time-to-Market 37
Time-to-Profit 37
Trac 300
Training 320
Transformationsprozesse 111
Trend 394
Triple Peaks 135
Twin Peaks 134

U

Umfang 69
UML 137, 275
Und-Dekomposition 167
Universalquantoren 111
Unschärferelation 136
Unsicherheit 36, 208, 404
Urheberrechte 318
Use Case 31, 106, 114, 142, 167
Use Case Points 160
User Story 106, 287

V

Validierung 76, 177, 185
Variabilitätsmodell 147, 149
VDI-Richtlinie 2519 389
Veränderungsmanagement 311, 319, 356, 393
Veränderungsprojekt 363
Verbesserungsmaßnahmen 363
Vereinbarung *siehe Abstimmung*
Vererbung 153
Verfolgbarkeit 39, 184, 198, 244, 246
Vergabe 346
Vergaberecht 217
Vergleich 160

Verhalten 153
Verhaltensperspektive 128
Verhandlung 229, 343
Verifikation 176, 189
Versionierung 251
Verständlichkeit 183, 194, 256
Verteilte Teams 283, 286, 298, 316, 331, 374, 399
Vertikale Verfolgbarkeit 246
Vertrag 66, 103, 216, 227, 229, 343, 390
Vertragsmodelle 225
Vertragspartner 227
Vertragsrecht 227
Vertragsverhandlung 332
Vision 35, 68
VisualStudio 301
V-Modell 325
Volere 107
Vollständigkeit 187, 195, 256
Vorgehensmodell 323
Vorgehensweise 356
Vorlage 104, 107, 109, 179

W

Wartung 249, 325, 343, 393
Watzlawick, Paul 374
Werbung 222
Werkvertrag 220
Werkzeug 41, 251, 298, 303
Aufwandsschätzung 160
 Bewertung 306
 DOORS 307
 Moqups 301
Werkzeugkosten 319
Wert 79, 257, 395
Wertorientierung 10, 68, 395, 404
Wertstromanalyse 269

Wettbewerb 52
Wiederverwendung 147, 184, 291, 315, 402
Wiki 298, 300
Win-win-Methode 56, 58, 374, 380
Win-win-Situation 228
Wissensmanagement 402
Workflow 31
Workflow-Management 316
Workshop 73, 75

X

XP 267

Y

YAGNI 286

Z

Ziel 53, 159
Zielmodell 167
Zukunft 394
Zustand 153
Zustandsanalyse 168
Zustandsmodell 146, 169
Zustimmung 202
Zuverlässigkeit 81
Zuweisung 202
Zykluszeit 404

Ziffern

150%-Modell 149
5 Why 282