

Konform zum  
CPRE Foundation  
Level –  
Version 3.0

# Basiswissen



# Requirements Engineering

Aus- und Weiterbildung nach  
IREB-Standard zum  
Certified Professional for  
Requirements Engineering  
Foundation Level

 dpunkt.verlag

5., überarbeitete und aktualisierte Auflage

# Inhalt

**Cover**

**Titel**

**Impressum**

**Die Zertifizierung zum Certified Professional for Requirements Engineering (CPRE)**

**Vorwort**

**Unter Mitwirkung von**

**Unsere Familie Meyer**

**Inhaltsübersicht**

**Inhaltsverzeichnis**

## **1 Einleitung und Grundlagen**

1.1 Requirements Engineering – Was?

Exkurs: Qualitätsanforderungen

Exkurs: Randbedingungen (Constraints)

1.2 Requirements Engineering – Warum?

Exkurs: Kommunikationsprobleme

1.3 Requirements Engineering – Wo?

Exkurs: Ziele und Szenarien

1.4 Requirements Engineering – Wie?

1.5 Die Rolle und Aufgaben eines Requirements Engineer

Exkurs: Persönlichkeitsprofil eines Requirements Engineer

1.6 Was über Requirements Engineering zu lernen ist

## **2 Grundlegende Prinzipien des Requirements Engineering**

2.1 Neun grundlegende Prinzipien

2.1.1 Prinzip 1: Wertorientierung – Anforderungen sind Mittel zum Zweck, kein Selbstzweck

Exkurs: Anforderungen sind kein Selbstzweck

2.1.2 Prinzip 2: Stakeholder – Im Requirements Engineering geht es darum, die Wünsche und Bedürfnisse der Stakeholder zu befriedigen

Exkurs: Wichtige Anforderungsquellen zusätzlich zu Stakeholdern

2.1.3 Prinzip 3: Gemeinsames Verständnis – Erfolgreiche Systementwicklung ist ohne eine gemeinsame Basis nicht möglich

Exkurs: Gemeinsames Verständnis schaffen

2.1.4 Prinzip 4: Kontext – Systeme können nicht isoliert verstanden werden

Exkurs: Probleme mit getrennter Umfang-(Scope-)Abgrenzung

2.1.5 Prinzip 5: Problem · Anforderung · Lösung – Ein unausweichlich ineinandergreifendes Tripel

Exkurs: Twin-Peaks-Modell

2.1.6 Prinzip 6: Validierung – Nicht validierte Anforderungen sind nutzlos

Exkurs: Validierung

2.1.7 Prinzip 7: Evolution – Sich ändernde Anforderungen sind kein Unfall, sondern der Normalfall

Exkurs: Änderungen vs. Stabilität

2.1.8 Prinzip 8: Innovation – Mehr vom Gleichen ist nicht genug

2.1.9 Prinzip 9: Systematische und disziplinierte Arbeit – im Requirements Engineering unverzichtbar

Exkurs: Auswahl von Praktiken und Techniken

2.2 System und Kontextabgrenzung

2.2.1 Systemkontext

2.2.2 System- und Kontextgrenzen bestimmen

Exkurs: Dokumentation des Systemkontexts

### **3 Arbeitsergebnisse und Dokumentationspraktiken**

3.1 Arbeitsergebnisse im Requirements Engineering

3.1.1 Merkmale von Arbeitsergebnissen

3.1.2 Kategorien und Abstraktionsebenen

3.1.3 Detaillierungsgrad von Anforderungen

3.1.4 Aspekte von Arbeitsergebnissen

3.1.5 Allgemeine Dokumentationsrichtlinien

3.1.6 Planung der zu verwendenden Arbeitsergebnisse

3.2 Natürlichsprachige Arbeitsergebnisse

Exkurs: Sprachliche Effekte durch Transformation

3.2.1 Dokumentationsrichtlinien für natürlichsprachige Anforderungen

3.2.2 Sprachliche Effekte, auf die zu achten ist

3.3 Vorlagenbasierte Arbeitsergebnisse

3.3.1 Satzschablonen

Exkurs: Schritt für Schritt zur Anforderung

3.3.2 Formularvorlagen

Exkurs: Vorlagenbasierte Spezifikation von Use Cases

3.3.3 Dokumentvorlagen

Exkurs: Standardisierte Dokumentvorlagen

3.4 Modellbasierte Arbeitsergebnisse

3.4.1 Die Rolle von Modellen im Requirements Engineering

Exkurs: Eigenschaften von Modellen

Exkurs: Modellierungssprachen

3.4.2 Kontextmodellierung

Exkurs: Zielmodellierung im Requirements Engineering

Exkurs: Kontextmodellierung mit SysML-Blockdiagrammen

3.4.3 Modellierung von Struktur und Daten

Exkurs: Fortgeschrittene Modellierung von Struktur und Daten

3.4.4 Modellierung von Funktion und Ablauf

Exkurs: Use-Case-Modelle und -Diagramme

3.4.5 Modellierung von Zustand und Verhalten

Exkurs: Fortgeschrittene Zustandsmaschinendiagramme

Exkurs: Integration der Perspektiven auf funktionale Anforderungen

### 3.5 Glossare

3.5.1 Grundlagen von Glossaren

3.5.2 Regeln für den Umgang mit einem Glossar

### 3.6 Dokumentationsstrukturen für Anforderungen

### 3.7 Prototypen im Requirements Engineering

3.7.1 Explorative Prototypen

3.7.2 Evolutionäre Prototypen

### 3.8 Qualitätskriterien für Arbeitsergebnisse und Anforderungen

3.8.1 Qualitätskriterien für einzelne Anforderungen

3.8.2 Qualitätskriterien für ein Menge von Anforderungen

## **4 Praktiken für die Erarbeitung von Anforderungen**

### 4.1 Quellen für Anforderungen

4.1.1 Stakeholder und deren Bedeutung

Exkurs: Personas

4.1.2 Der Umgang mit Stakeholdern im Projekt

Exkurs: Stakeholder-Relationship-Management

4.1.3 Weitere Anforderungsquellen

### 4.2 Ermittlung von Anforderungen

4.2.1 Anforderungskategorisierung nach dem Kano-Modell

4.2.2 Arten von Ermittlungstechniken

4.2.2.1 Erhebungstechniken

Exkurs: Befragungstechniken

Exkurs: Kollaborationstechniken als Hilfstechiken

Exkurs: Beobachtungstechniken

4.2.2.2 Entwurfs- und Ideenfindungstechniken

Exkurs: Kreativitätstechniken

Exkurs: Szenarien

#### 4.2.2.3 Auswahl der richtigen Ermittlungstechnik

#### 4.3 Abstimmung und Konfliktlösung

##### 4.3.1 Konfliktidentifikation

##### 4.3.2 Konfliktanalyse

##### 4.3.3 Konfliktlösung

##### 4.3.4 Dokumentation der Konfliktlösung

#### 4.4 Validieren von Anforderungen

##### 4.4.1 Grundlagen der Validierung von Anforderungen

Exkurs: Qualitätsaspekte von Anforderungen

Exkurs: Qualitätsaspekt »Inhalt«

Exkurs: Qualitätsaspekt »Dokumentation«

Exkurs: Qualitätsaspekt »Abgestimmtheit«

##### 4.4.2 Wichtige Aspekte der Anforderungvalidierung

Exkurs: Auswahl der Validierer

Exkurs: Konzentration auf Aufdeckung von Fehlern

##### 4.4.3 Reviewtechniken zur Validierung von Anforderungen

###### 4.4.3.1 Walkthrough

Exkurs: Stellungnahme als Sonderfall

###### 4.4.3.2 Inspektion

Exkurs: Rollen bei einer Inspektion

Exkurs: Assistenztechniken zur Unterstützung des Reviews

##### 4.4.4 Explorationstechniken

###### 4.4.4.1 Prüfung durch Prototypen

Exkurs: Durchführung einer Prüfung mittels Prototyping

###### 4.4.4.2 Prüfung durch kontrollierte Experimente

###### 4.4.4.3 Probe-Entwicklung (Konstruktion von Entwicklungsartefakten)

Exkurs: Wechsel der Dokumentationsform

## **5 Prozess und Arbeitsstruktur**

## 5.1 Einflussfaktoren

5.1.1 Eignung des Gesamtprozesses

5.1.2 Entwicklungskontext

5.1.3 Fähigkeiten und Verfügbarkeit von Stakeholdern

5.1.4 Gemeinsames Verständnis

5.1.5 Komplexität und Kritikalität des zu entwickelnden Systems

5.1.6 Vorgegebene Randbedingungen

5.1.7 Verfügbare Zeit und Budget

5.1.8 Volatilität der Anforderungen

5.1.9 Erfahrungen des Requirements Engineer

## 5.2 Facetten der Requirements-Engineering-Prozesskonfiguration

5.2.1 Zeitfacette: linear versus iterativ

5.2.2 Zweckfacette: präskriptiv versus explorativ

5.2.3 Zielfacette: kundenspezifisch versus marktorientiert

5.2.4 Hinweis und Warnungen

## 5.3 Konfigurieren eines Requirements-Engineering-Prozesses

5.3.1 Partizipativer Requirements-Engineering-Prozess: iterativ, explorativ und kundenspezifisch

5.3.2 Vertraglich regulierter Requirements-Engineering-Prozess: typischerweise linear, präskriptiv und kundenspezifisch

5.3.3 Produktorientierter Requirements-Engineering-Prozess: iterativ, explorativ und marktorientiert

5.3.4 Weitere zu berücksichtigende Aspekte

## **6 Praktiken für das Requirements Management**

6.1 Was ist Requirements Management?

6.2 Verwaltung des Lebenszyklus

6.3 Versionskontrolle

6.4 Konfigurationen und Basislinien

Exkurs: Dimensionen von Anforderungskonfigurationen

Exkurs: Basislinien

## 6.5 Attribute und Sichten

6.5.1 Attribuierung von natürlichsprachigen Anforderungen und Anforderungsmodellen

Exkurs: Attribuierungsschemata

6.5.2 Sichten auf Anforderungen

Exkurs: Selektive Sicht

Exkurs: Verdichtende Sicht

## 6.6 Verfolgbarkeit von Anforderungen

Exkurs: Nutzen und Arten der Verfolgbarkeit

6.6.1 Verwendungszweckbezogene Definition der Verfolgbarkeit

6.6.2 Repräsentation der Verfolgbarkeit

## 6.7 Umgang mit Änderungen

Exkurs: Change Control Board

Exkurs: Änderungsantrag für Anforderungen

## 6.8 Priorisierung von Anforderungen

6.8.1 Vorgehen zur Priorisierung von Anforderungen

6.8.2 Techniken zur Priorisierung von Anforderungen

# **7 Werkzeugunterstützung**

## 7.1 Werkzeuge im Requirements Engineering

Exkurs: Nutzung von nicht für das Requirements Engineering entwickelten Werkzeugen

Exkurs: Requirements-Management-Werkzeuge

Exkurs: Spezialisierte Werkzeuge für das Requirements Management

Exkurs: Standard-Büroanwendungen

Exkurs: Modellierungswerkzeuge

## 7.2 Werkzeugeinführung

# **Anhang**

Videoverzeichnis

Animationsverzeichnis

Kernfaktenverzeichnis

Literatur

Index

## 4 Praktiken für die Erarbeitung von Anforderungen

### 4.1 Quellen für Anforderungen

Die Qualität der Anforderungen begründet sich vor allem auf den einbezogenen Anforderungsquellen. Wird eine relevante Anforderungsquelle außer Acht gelassen, kann dies gravierende Auswirkungen auf die Vollständigkeit und Qualität der Anforderungen und damit auf den Projekterfolg haben. Die Identifikation von Anforderungsquellen basiert auf einem gemeinsamen Verständnis des Kontexts (Prinzip 3, Abschnitt 2.1.3) und auf der korrekten Abgrenzung von System und Kontext (Prinzip 4, Abschnitt 2.1.4).

Es werden drei verschiedene Arten von Anforderungsquellen unterschieden:

*Drei Arten von  
Anforderungsquellen*

- *Stakeholder* (siehe Abschnitt 1.1) ist eine Person oder Organisation, die (direkt oder indirekt) Einfluss auf die Anforderungen hat oder von den Anforderungen betroffen ist. Beispiele für Stakeholder sind Benutzer des Systems, Betreiber des Systems, Sponsoren, Entwickler, Architekten, Auftraggeber oder Behörden (die Standards, Regularien, Normen herausgeben oder über Genehmigungen entscheiden).
- *Dokumente* enthalten oft wichtige Informationen, aus denen Anforderungen gewonnen werden können. Beispiele für Dokumente sind z. B. rechtliche und regulatorische Dokumente (Normen/Standards oder Gesetzestexte) sowie branchen-/organisationsspezifische Dokumente, aber auch Anforderungsspezifikationen oder Fehlerberichte des Altsystems.
- *Systeme* bezeichnen zum einen Nachbarsysteme, d.h. Systeme, mit denen das in Entwicklung befindliche System operativ interagieren wird, zum anderen können aber auch sowohl Alt- bzw. Vorgängersysteme als auch

Konkurrenzsysteme relevante Quellen für Anforderungen darstellen. Beispielsweise wird den Stakeholdern durch die Möglichkeit der Nutzung des derzeitigen Systems ein Eindruck vermittelt, auf dessen Basis die Stakeholder Erweiterungen oder Änderungen fordern können.

#### Kernfakten 4-1: Quellen für Anforderungen



[www.cpre-buch.de/pk4w1](http://www.cpre-buch.de/pk4w1)

### 4.1.1 Stakeholder und deren Bedeutung

Stakeholder sind gemeinhin die wichtigste Quelle für Anforderungen (Prinzip 2, Abschnitt 2.1.2). Somit ist das Identifizieren relevanter Stakeholder eine bedeutende Aufgabe des Requirements Engineering. Sie sollte zu Beginn einer Entwicklung stattfinden, und die Ergebnisse sollten im Rahmen einer kontinuierlichen Aktivität verwaltet werden.

*Bedeutung von Stakeholdern*

Wichtige Stakeholder eines Systems sind u.a.:

- Die erwarteten (*Be-*)Nutzer des Systems
- *Sponsoren* und andere Geldgeber
- Die *Entwickler* des Systems. Zum Beispiel haben die Fähigkeiten der Entwickler unmittelbaren Einfluss auf die Anforderungen, da diese u.a. Technologieentscheidungen beeinflussen. Neben den Entwicklern sind auch weitere am Entwicklungsprozess beteiligte Personen als Stakeholder relevant, beispielsweise der Business- und/ oder Prozessmanager, der Qualitätssicherungsverantwortliche, Vertriebsmitarbeiter und viele mehr.
- *Behörden*, die die rechtlichen und regulatorischen Rahmenbedingungen des Systems beeinflussen.
- Und natürlich die *Kunden* bzw. der Auftraggeber

Der Requirements Engineer hat die Aufgabe, die sich teils widersprechenden Ziele und Anforderungen der unterschiedlichen Stakeholder zu sammeln, zu dokumentieren und mit allen Beteiligten zu konsolidieren [Potts et al. 1994] (siehe Abschnitt 4.3).

Bleiben Stakeholder unberücksichtigt oder werden wichtige Stakeholder nicht identifiziert, hat dies signifikante negative Auswirkungen auf den

*Auswirkung fehlender Stakeholder*

gesamten Projektverlauf, da hierdurch Anforderungen nicht erkannt werden. Spätestens im Betrieb des Systems treten die übersehenen Anforderungen jedoch als Änderungsanträge zutage, deren nachträgliche Integration hohe Zusatzkosten verursacht. Das kann sogar so weit gehen, dass ein implementiertes System nach der ersten Benutzung wieder außer Betrieb genommen werden muss. Es ist daher essenziell, alle Stakeholder zu identifizieren und effektiv in die Anforderungsermittlung einzubeziehen.

Die folgenden Dokumente und Personen unterstützen bei der systematischen Identifikation von Stakeholdern:

#### Identifikation der Stakeholder

- *Checklisten* enthalten typische Stakeholder-Gruppen und -Rollen. Für jede Rolle sollte dann mindestens ein konkreter Stakeholder gefunden und verpflichtet werden. Außerdem muss sichergestellt werden, dass alle relevanten Stakeholder-Gruppen in ausreichendem Maße Berücksichtigung finden.
- *Organisationsstrukturen* zeigen die Verantwortlichen in der Hierarchie. Beispielsweise können Organigramme verwendet werden, um notwendige Entscheider zu identifizieren, die als Anforderungsquelle berücksichtigt werden müssen.
- *Geschäftsprozessdokumentationen* zeigen die wichtigen Geschäftsprozesse (z.B. mit BPMN-Modellen). Zum einen werden hier häufig die relevanten Process Owner genannt, zum anderen können ausgehend von den Prozessen die relevanten Prozessbeteiligten identifiziert werden.
- *Marktberichte* oder auch Marktstudien können beispielsweise genutzt werden, um Trends und potenzielle Kunden vorherzusehen, die mit dem System bedient werden sollen, oder um Wettbewerber zu identifizieren.
- *Anfängliche Stakeholder*, die nach zusätzlichen relevanten Stakeholdern befragt werden können. Zum Beispiel könnte bei der Identifikation weiterer Stakeholder mit der Befragung des Managements oder von Fachexperten begonnen werden. Von diesen ausgehend, können weitere relevante Stakeholder ermittelt werden.

Hat das System eine Benutzerschnittstelle, so stellen die Endbenutzer des Systems eine sehr wichtige Stakeholder-Gruppe dar. Bei sehr vielen Benutzern ist es unmöglich, mit allen zu sprechen. Diese sollten in Benutzergruppen zusammengefasst werden und für sie sollten entweder geeignete Repräsentanten gewählt oder eine Persona für die Benutzergruppe erstellt werden [Cooper 2004].

*Personas* sind fiktive Charaktere, die vom Requirements Engineer erschaffen werden, wenn kein Zugriff auf die echten Stakeholder einer Rolle besteht. So

können beispielsweise spezielle Systembenutzer simuliert werden.

### **Exkurs: Personas**

*Eine Persona ist ein fiktiver Charakter einer Gruppe von Nutzern mit ähnlichen Bedürfnissen, Werten oder Gewohnheiten, bei denen erwartet wird, dass sie ein System auf ähnliche Art nutzen.*

Übersetzt aus [IREB-Glossar 2020]

Personas können nicht nur genutzt werden, um echte Stakeholder zu ersetzen, sondern auch um Stakeholder-Gruppen zu bündeln. Stakeholder mit gleichartigen Interessen werden mit einer Persona zusammengefasst [Cooper 2004]. Hier mag es sich anbieten, zusätzlich zu der Persona aus der Gruppe der Stakeholder einen repräsentativen Stakeholder auszuwählen, der die Rolle dieser Persona spielt. Dennoch ist immer zu bedenken, dass die Persona selbst fiktiv ist, da sie nicht nur einen Stakeholder repräsentiert, sondern eine Gruppe unterschiedlicher Stakeholder, die zwar eine Schnittmenge in den gemeinsamen Interessen haben, in der Regel aber an anderer Stelle unterschiedliche Ziele verfolgen.

#### **4.1.2 Der Umgang mit Stakeholdern im Projekt**

In der Praxis ist häufig festzustellen, dass bei umfangreichen bzw. komplexen Projekten viele Stakeholder involviert sind. Dies hat zur Folge, dass aufgrund von Ressourcenbeschränkungen, die für die Ermittlung der Anforderung geeignetsten Stakeholder selektiert werden müssen. Für die Identifikation der geeignetsten Stakeholder empfehlen sich Stakeholder-Listen. Solche Stakeholder-Listen werden auch zur Dokumentation der beteiligten Stakeholder benötigt. Für die Dokumentation der Stakeholder eines Entwicklungsprojekts eignen sich Tabellen, die mindestens die folgenden Daten beinhalten müssen:

Verwaltung der Stakeholder

- Name,
- Funktion (Rollenbezeichnung),
- zusätzliche persönliche Daten und Kontaktdaten,
- zeitliche und räumliche Verfügbarkeit während der Projektlaufzeit,
- Relevanz (Bedeutung des Stakeholders),
- Fachgebiet und Umfang an Fachwissen sowie
- Ziele und Interessen bezogen auf das Projekt.

Tabelle 4-1 zeigt eine beispielhafte Attributbelegung für oben genannte Kriterien der Stakeholder-Dokumentation.



**Tab. 4-1** Attribute von Stakeholdern

<b>Stakeholder-ID</b>	2
<b>Name</b>	Opa Dieter
<b>Kontaktdaten</b>	opa.dieter@mail.de 0121/1234567
<b>Verfügbarkeit</b>	Mo – Do 13–16 Uhr im Haus der Meyers
<b>Relevanz</b>	Meinung ist wichtig für den Auftraggeber
<b>Fachgebiet und Umfang an Fachwissen</b>	Unterstützt im Haushalt
<b>Ziele und Interessen</b>	Will das Haus der Meyers auch weiterhin unkompliziert betreten

Stakeholder haben Bedürfnisse, die im Projekt beachtet werden müssen, und hier setzt das Stakeholder-Relationship-Management an. Es hat sich bewährt, die Rechte und Pflichten der Stakeholder, aber auch des Requirements Engineer zu Beginn des Projekts klar zu definieren, um die Zusammenarbeit von vornherein auf sichere Füße zu stellen.

Trotz klarer Regeln können persönliche Konflikte zwischen Stakeholdern die Ermittlung von Anforderungen erschweren. Hier ist es die Aufgabe des Requirements Engineer, mit Fingerspitzengefühl die richtigen Techniken auszuwählen, um zu vermeiden, dass persönliche Konflikte die Ermittlung von Anforderungen negativ beeinflussen. Beispielsweise könnten in Fällen persönlicher Konflikte getrennte Meetings durchgeführt werden, um die Anforderungen getrennt zu erheben, Konflikte herauszufiltern und dann nur diese mit einer Konfliktlösungstechnik gezielt zu bearbeiten.

**Exkurs: Stakeholder-Relationship-Management**

Ein effektiver Umgang mit den Stakeholdern beinhaltet einen kontinuierlichen Informationsfluss: Regelmäßige Statusbesprechungen sowie eine dauerhafte

Integration der Stakeholder helfen dem Requirements Engineer, aus Projektbetroffenen (d.h. prinzipiell betroffene Stakeholder) Projektbeteiligte (d.h. gut integrierte, mitverantwortliche Stakeholder) zu machen.

Stakeholder, denen der Requirements Engineer zu wenig Aufmerksamkeit widmet, können sich dem Projekt kritisch entgegenstellen. Zudem weisen manche Stakeholder eventuell einen Mangel an Motivation auf, da sie entweder mit dem Altsystem zufrieden sind, Angst vor Veränderungen haben oder durch negative Erfahrungen in früheren Projekten vorbelastet sind.

Es ist Aufgabe des Requirements Engineer, den Projektleiter dabei zu unterstützen, alle Stakeholder vom Nutzen des Projekts zu überzeugen. Um Missverständnissen, Kompetenzstreitigkeiten etc. vorzubeugen, ist es zweckmäßig, dass der Requirements Engineer mit jedem Stakeholder jeweils formal vereinbart, welche Aufgaben, Verantwortungsbereiche, Weisungsbefugnisse und individuellen Ziele bestehen und welche Kommunikationswege und Feedbackschleifen von dem Stakeholder genutzt werden können. Je nach Unternehmenskultur kann diese Vereinbarung in einfacher Form durch eine mündliche Vereinbarung oder auch formeller ausgestaltet und schriftlich getroffen werden. Die Vorgesetzten sollten den jeweiligen Vereinbarungen zustimmen.

Aus der Stakeholder-Vereinbarung resultiert für den Requirements Engineer sowie für jeden Stakeholder eine Reihe von Rechten und Pflichten.

#### Der Requirements Engineer

- spricht die Sprache der Stakeholder,
- arbeitet sich in das Fachgebiet gründlich ein,
- erstellt ein Anforderungsdokument,
- kann die Arbeitsergebnisse verständlich machen (z.B. mithilfe von Diagrammen oder Grafiken),
- pflegt einen respektvollen Umgang mit den Stakeholdern,
- präsentiert seine Ideen und Alternativen und deren Realisierung,
- ermöglicht es den Stakeholdern, Eigenschaften zu fordern (z.B. um das System einfach und benutzerfreundlicher zu machen),
- sorgt dafür, dass die Anforderungsspezifikation des zu entwickelnden Systems den funktionalen und qualitativen Ansprüchen der Stakeholder gerecht wird.

#### Der Stakeholder

- führt den Requirements Engineer in das Fachgebiet ein,

- versorgt den Requirements Engineer mit Anforderungen,
- formuliert die Anforderungen zielgerecht und gewissenhaft,
- trifft Entscheidungen zeitgerecht,
- respektiert die Einschätzung des Requirements Engineer bezüglich der Kosten, Machbarkeit und Priorisierung,
- priorisiert die Anforderungen aus seiner Sicht (die sich nicht notwendigerweise mit der Sicht der anderen Stakeholder decken muss),
- überprüft die dokumentierten Anforderungen des Requirements Engineer,
- kommuniziert unverzüglich Änderungen in seinen Anforderungen,
- befolgt den vorgegebenen Änderungsprozess.

Neben der Abgrenzung von Rechten und Pflichten ist es im Stakeholder-Relationship-Management Aufgabe des Requirements Engineer, die Planung und Koordination der Kommunikationswege sowie eine strukturierte Terminplanung für die durchzuführenden Requirements-Engineering-Aktivitäten mit den Stakeholdern zu vereinbaren. Diese Planung und die Art des Kommunikationswegs werden durch die Wahl der vom Requirements Engineer eingesetzten Ermittlungstechniken beeinflusst.

#### Kernfakten 4-2: Stakeholder



[www.cpre-buch.de/pk4w2](http://www.cpre-buch.de/pk4w2)

#### 4.1.3 Weitere Anforderungsquellen

Neben den Stakeholdern existieren weitere Quellen für Anforderungen, die durch den Requirements Engineer aufgedeckt und berücksichtigt werden müssen:

- *Bestehende Systeme und Altsysteme* geben Aufschluss über eine Vielzahl von Anforderungen, die der Kunde oftmals als selbstverständlich ansieht. Bestehende Systeme können beispielsweise Nachbarsysteme oder Konkurrenzsysteme sein. Nachbarsysteme liefern Anforderungen an Schnittstellenkompatibilität, Konkurrenzsysteme geben Auskunft über das Mindestmaß des erwarteten Leistungsumfangs. Altsysteme bezeichnen häufig existierende Vorgängersysteme, die einen guten Startpunkt für die Erhebung von Anforderungen bieten, da von diesen typischerweise umfangreiche Funktionalität übernommen werden soll.

- *Prozessdokumente* beschreiben den aktuellen Zustand der durch das System zu unterstützenden Prozesse, aber auch den Wunschzustand der Prozesse nach Einführung des Systems.
- *Rechtliche und regulatorische Dokumente* sind beispielsweise Gesetze, Normen und Standards, die während der Entwicklung unbedingt einzuhalten sind. Dies gilt sowohl für Einschränkungen des Entwicklungsprozesses (also wie das System zu entwickeln ist) als auch für Einschränkungen der Systemeigenschaften (also Vorgaben bzgl. des Funktionsumfangs oder der Qualität).
- *Unternehmensspezifische Vorschriften* ergänzen rechtliche und regulatorische Dokumente um weitere Richtlinien des Unternehmens.
- *Nutzerstudien oder Marktanalysen* geben Aufschluss über mögliche Nutzer des Systems.

Die verschiedenen Dokumente, die als Quelle von Anforderungen dienen, sind immer auch eng mit Personen verbunden, beispielsweise der Dokumentersteller, der Dokumentverantwortliche, die im Dokument genannten verantwortlichen Personen oder Rollenbeschreibungen, die von Personen ausgeführt werden. Diese Personen sind in vielen Fällen wertvolle Stakeholder, die im Projekt berücksichtigt werden sollten. Als solches können Dokumente auch als Ausgangsbasis für die Identifikation (weiterer) Stakeholder genutzt werden.

#### Kernfakten 4-3: Weitere Quellen für Anforderungen



[www.cpre-buch.de/pk4w3](http://www.cpre-buch.de/pk4w3)

## 4.2 Ermittlung von Anforderungen

Es existieren zahlreiche Ermittlungstechniken, die Unterstützung bei der Ermittlung von Wünschen und Bedürfnissen aller Stakeholder und der Berücksichtigung aller weiteren Anforderungsquellen bieten. Dabei ist es die Aufgabe des Requirements Engineer, sicherzustellen, dass die Anforderungen aller relevanten Anforderungsquellen mit angemessenen Techniken ermittelt werden. Die bewusste und situationsangepasste Nutzung von Ermittlungstechniken bietet die Möglichkeit, die Anforderungsermittlung an die in einem Projekt auftretenden Randbedingungen anzupassen und möglichst vollständige Anforderungen zu erhalten. Es ist dabei wichtig, die impliziten Forderungen, Wünsche und Erwartungen in explizite Anforderungen in der benötigten Qualität zu überführen.

*Anforderungsermittlung: kein Universalrezept*

#### 4.2.1 Anforderungskategorisierung nach dem Kano-Modell

Für die Anforderungsermittlung ist das Wissen, welche Bedeutung die Anforderungen für die Zufriedenheit der Stakeholder haben, sehr hilfreich.

*Einfluss der Anforderungen auf die Zufriedenheit*

Das Modell von Kano (siehe Abb. 4-1) teilt diese Zufriedenheit mit den jeweiligen Merkmalen eines Produkts in drei Kategorien ein [Kano et al. 1984]:

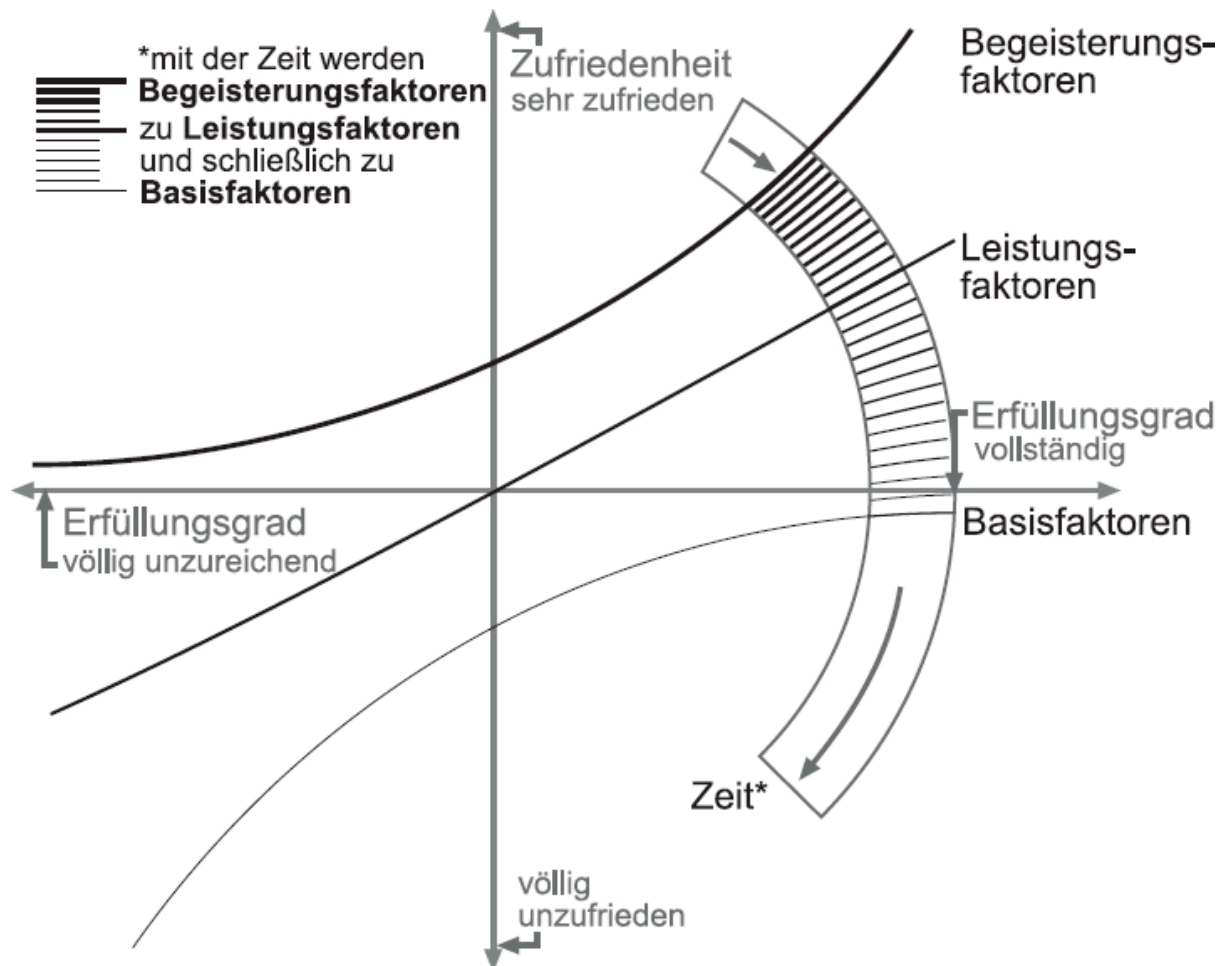


Abb. 4-1

Das Kano-Modell grafisch dargestellt



[www.cpre-buch.de/pk4v1](http://www.cpre-buch.de/pk4v1)

- *Basisfaktoren (auch Dissatisfiers, unterbewusste Anforderungen)* sind als selbstverständlich vorausgesetzte Systemmerkmale. Das System muss diese in jedem Fall vollständig erfüllen, sonst stellt sich beim Stakeholder massive Unzufriedenheit ein. Vollständig erfüllte Basisfaktoren erzeugen keine positive Stimmung, sondern vermeiden lediglich, dass starke Unzufriedenheit entsteht. Basisfaktoren werden vor allem durch bereits vorhandene Systeme (Vorgänger oder Konkurrenzsysteme) geprägt. Für

deren Ermittlung eignen sich deshalb besonders Beobachtungstechniken und artefaktbasierte Techniken.

- *Leistungsfaktoren (auch Satisfiers, bewusste Anforderungen)* sind die explizit geforderten Systemmerkmale. *Leistungsfaktoren* sind die Merkmale, die der Stakeholder bewusst und explizit fordert. Die Erfüllung dieser Merkmale erzeugt Stakeholder-Zufriedenheit und ist erstrebenswert. Fehlen einige der geforderten Merkmale, akzeptiert der Stakeholder das Produkt vermutlich, doch seine Zufriedenheit sinkt mit jedem fehlenden Leistungsfaktor. Leistungsfaktoren lassen sich gut durch Befragungstechniken ermitteln, da damit auf das bewusste Wissen des Stakeholders zugegriffen wird.
- *Begeisterungsfaktoren (auch Delighters, unbewusste Anforderungen)* sind Systemmerkmale, die der Stakeholder nicht kennt und erst während der Benutzung als angenehme und nützliche Überraschungen entdeckt. Für die Ermittlung solcher Begeisterungsfaktoren sind insbesondere Entwurfs- und Ideenfindungstechniken geeignet.

Im Laufe der Zeit werden aus Begeisterungsfaktoren Leistungsfaktoren und schließlich Basisfaktoren, denn der Nutzer gewöhnt sich an Merkmale eines Systems. Bei der Ermittlung der Anforderungen sind alle drei Anforderungskategorien zu berücksichtigen, um eine möglichst vollständige Sammlung von Anforderungen zu erhalten.

#### **4.2.2 Arten von Ermittlungstechniken**

Ermittlungstechniken erfüllen den Zweck, die bewussten, unbewussten und unterbewussten Anforderungen der Stakeholder herauszufinden. Es existiert jedoch keine universelle Methode, um Anforderungen zu ermitteln. Jedes Projekt besitzt zwar individuelle Randbedingungen, einen eigenen Charakter und ist einzigartig, ist jedoch stets mit einigen Ermittlungstechniken kompatibel.

*Einflussfaktoren auf die Wahl der Ermittlungstechnik*

Grob lassen sich die Techniken zur Ermittlung von Anforderungen in zwei Kategorien einteilen:

- *Erhebungstechniken*
- *Entwurfs- und Ideenfindungstechniken*

Wie wir im Folgenden sehen werden, können diese Kategorien noch weiter unterteilt werden. Abbildung 4–2 zeigt diese Kategorien und ordnet diesen beispielhaft konkrete Techniken zu. Hierbei ist zu beachten, dass unterschiedliche

Techniken durchaus – abhängig von der Verwendung – auch noch weiteren Kategorien zugeordnet werden könnten.



**Abb. 4-2**

Arten von Ermittlungstechniken

#### 4.2.2.1 Erhebungstechniken

Erhebungstechniken sind Techniken zur Ermittlung [Baxter et al. 2015] von Leistungsfaktoren und Basisfaktoren. Diese Erhebungstechniken lassen sich wiederum in vier Hauptkategorien unterteilen:

- **Befragungstechniken**  
 Mit Befragungstechniken wird versucht, direkt vom Stakeholder eine möglichst genaue und unverfälschte Aussage über seine Anforderungen an das System zu erhalten. Alle Befragungstechniken setzen voraus, dass der Befragte sein Wissen explizit ausdrücken kann und dass er bereit ist, Zeit und Engagement in die Ermittlung zu investieren. Auch in diesem Fall besteht jedoch das Problem, dass viele Anforderungen aus implizitem Wissen resultieren; also Wissen, das Stakeholder haben und auch formulieren können und wollen, sich aber nicht bewusst sind, dass dieses wichtig ist, oder es schlicht im Gespräch vergessen. An dieser Stelle müssen Befragungstechniken einsetzen, um dieses unterbewusste Wissen aufzudecken. Befragungstechniken sind tendenziell vom Requirements Engineer getrieben, da dieser die Fragen vorgibt. Dadurch können Anliegen der Stakeholder eventuell verdrängt, vergessen oder vernachlässigt werden. Beispiele von Befragungstechniken sind Interviews und Fragebögen.

#### Exkurs: Befragungstechniken

Die Klassiker der Befragungstechnik sind das Interview und der Fragebogen.

- Im Interview stellt der Requirements Engineer einem oder mehreren Stakeholdern vorgegebene Fragen und protokolliert deren Antworten. Im Gespräch auftretende Fragen können sofort geklärt werden, und der Requirements Engineer hat die Möglichkeit, durch weitergehende Fragen auch implizite Anforderungen aufzudecken. Interviews können während der gesamten Systementwicklung eingesetzt werden. Ein erfahrener Interviewer passt im Verlauf des Gesprächs seine Fragen individuell an, geht konkret auf einzelne Stakeholder ein, fragt gezielt nach und sorgt somit durch Nachfragen für die Vollständigkeit der Antworten. Größter Nachteil dieser Ermittlungstechnik ist der hohe Zeitaufwand.
- Durch einen Fragebogen mit offenen und/oder geschlossenen Fragen (z.B. Multiple-Choice-Fragen) können ebenfalls Anforderungen der Stakeholder ermittelt werden. Bei vielen zu befragenden Personen kann dies auch online stattfinden. Mit Fragebögen können in kurzer Zeit und mit geringen Kosten sehr viele Informationen eingeholt werden. Sofern im Fragebogen potenzielle Antworten vorformuliert werden, können Stakeholder ihre Wertung auch zu ihnen unbewussten Anforderungen abgeben. Ein Nachteil des Fragebogens ist, dass damit nur abgefragt werden kann, was der Requirements Engineer bereits kennt oder vermutet. Die Erstellung eines guten Fragebogens ist zudem oft aufwendig und erfordert fundiertes Wissen über die Anwendungsdomäne und über die Technik und Psychologie der Fragebogenerstellung. Darüber hinaus gibt es bei Fragebögen im Vergleich zu Interviews keine unmittelbare Rückkopplung zwischen Fragendem und Befragtem, sodass z. B. vergessene oder schlecht formulierte Fragen erst im Zuge der Auswertung des Fragebogens zutage treten. Deswegen müssen Fragebögen immer mehrfach pilotiert werden, um potenzielle Missverständnisse und Fehler vor dem eigentlichen Einsatz bereits auszuräumen.

- *Kollaborationstechniken*

In Kooperation mit verschiedenen Stakeholdern werden gemeinsam Anforderungen erarbeitet. Dies geschieht beispielsweise mittels Workshops.

### **Exkurs: Kollaborationstechniken als Hilfstechniken**

Kollaborationstechniken dienen häufig der Unterstützung anderer Techniken. Häufig werden diese im Rahmen von Workshops genutzt, da diese den Grundrahmen für die Kollaboration legen. In einem Zusammentreffen von Requirements Engineers und Stakeholdern wird das Thema, was das System

leisten soll, intensiv erarbeitet. Zum Beispiel können in einem *Workshop* die benötigten Benutzerschnittstellen des Systems entworfen werden [Gottesdiener 2002]. Unterstützende Techniken dienen bei der Anwendung von Ermittlungstechniken als Ergänzung und ermöglichen es, Schwächen der gewählten Ermittlungstechnik auszugleichen.

- *Mindmapping* ist das Anlegen einer grafischen Darstellung, die Verfeinerungsbeziehungen und Abhängigkeiten zwischen den Begriffen grafisch abbildet. Mindmapping wird häufig als unterstützende Technik für das Brainstorming oder Brainstorming paradox verwendet.
  - *Audio- und Videoaufzeichnungen* sind sehr gut geeignet zur Ermittlung von Anforderungen bei schlechter Verfügbarkeit der Stakeholder, knappem Budget oder einer hohen Systemkritikalität. Insbesondere bei der Feldbeobachtung helfen sie, schnell ablaufende Prozesse für den Requirements Engineer reproduzierbar zu machen. Nachteilig ist, dass durch die Aufzeichnung sich die beteiligten Personen evtl. überwacht fühlen und dadurch verfälschte Aussagen liefern oder im Extremfall die Zusammenarbeit verweigern könnten. Der Einsatz von Video- oder Audioaufzeichnungen muss deshalb vorher motiviert und abgeklärt werden.
  - *Use-Case-Modellierung (Anwendungsfallmodellierung)*  
Mit Use Cases wird zunächst die Außensicht des zu erstellenden Systems dokumentiert. Zwischen dem auslösenden Ereignis und dem vom System erwarteten Ergebnis wird jeweils ein Use Case definiert, der durch das System unterstützt werden muss (siehe Abschnitt 3.4.2).
  - *Prototypen* eignen sich besonders zum Hinterfragen von bereits erarbeiteten Anforderungen und zur Ermittlung von Anforderungen in Situationen, in denen die Stakeholder nur vage Vorstellungen von dem haben, was entwickelt werden soll. Mögliche Konsequenzen von neuen bzw. geänderten Anforderungen können somit leichter erkannt werden. Zum Beispiel werden in der Praxis oft User-Interface-Prototypen eingesetzt, um zusätzliche funktionale Anforderungen zu finden.
- 
- *Beobachtungstechniken*  
Beobachtungstechniken setzen darauf, dass sich die Anforderungen an ein neues System aus der Beobachtung der Bedienung des Alt-systems oder der Beobachtung von Arbeitsprozessen und der Hinterfragung des Gesehenen ermitteln lassen. Somit eignen sich Beobachtungstechniken auch dazu, unbewusste Anforderungen aufzudecken. Beispiele für Beobachtungstechniken sind Feldbeobachtungen und Apprenticing.

## Exkurs: Beobachtungstechniken

In Situationen, in denen Fachspezialisten nicht die Zeit besitzen, das benötigte Wissen an den Requirements Engineer weiterzugeben, eignen sich Beobachtungstechniken. Dabei werden die entsprechenden Stakeholder vom Requirements Engineer bei ihrer Arbeit beobachtet, ihre Arbeitsschritte dokumentiert und daraus die vom System zu unterstützenden Arbeitsabläufe, aber auch potenzielle Fehler, Risiken und offene Fragen ermittelt.

Dies alles sind potenzielle Informationsquellen, um darauf basierend Anforderungen zu formulieren. Die Stakeholder können entweder dem Requirements Engineer ihr Wissen hierbei aktiv in der Anwendung demonstrieren oder nur beobachtet werden. Dabei sollte der Requirements Engineer die beobachteten Abläufe hinterfragen, um die Soll-Situation zu ermitteln, da ansonsten die Gefahr besteht, dass viele veraltete Technologieentscheidungen und suboptimale Prozesse (Ist-Situation) dokumentiert werden. Als externer Beobachter hat der Requirements Engineer gute Chancen, ineffiziente Prozesse zu erkennen und bessere Lösungen vorzuschlagen. Er besitzt mehr Abstand zum gelebten Prozess als die Stakeholder, die häufig aus Gewohnheit ihre Arbeitsschritte durchführen, ohne sie kritisch zu reflektieren. Beobachtungstechniken eignen sich gut dazu, detaillierte Anforderungen und Basisfaktoren zu ermitteln, denn der Requirements Engineer nimmt Basisfaktoren wahr, die viele Stakeholder als bekannt voraussetzen oder nur unbewusst kennen. Außerdem lernt der Requirements Engineer dabei den jeweiligen Fachjargon, was weitere Befragungen erleichtert. Leistungsfaktoren können nur beobachtet werden, wenn sie bereits im gelebten Prozess oder Vorgängersystem umgesetzt sind, deshalb eignet sich diese Technik nicht bei völlig neuen Aspekten der Entwicklung. In der Systementwicklung eignen sich insbesondere die Feldbeobachtung und das »Apprenticing« als Ermittlungstechniken.

- Der Requirements Engineer ist vor Ort bei den Spezialisten bzw. Anwendern des Systems und beobachtet und dokumentiert unmittelbar die stattfindenden Prozesse, Handgriffe und Arbeitsabläufe. Aus diesen Beobachtungen ermittelt er die Anforderungen. Häufig wird dies durch Audio- oder Videoaufzeichnungen unterstützt. Diese Technik ist gut geeignet bei sprachlich schwer vermittelbaren Arbeitsabläufen, jedoch nur bei wirklich beobachtbaren Abläufen.
- Das *Apprenticing* (»in die Lehre gehen«) erfordert vom Requirements Engineer, dass er die Tätigkeiten der Nutzer des Systems konkret erlernen und ausführen muss. Wie ein Lehrling wird der Requirements Engineer aufgefordert, unklare und unverständliche Handlungsschritte

sofort zu hinterfragen, um in dem Bereich Erfahrung zu sammeln. Er kann hierdurch Anforderungen erfahren, die für den Stakeholder so selbstverständlich sind, dass er sie nicht mehr äußert. Ein weiterer Vorteil ist, dass das typische Machtverhältnis zwischen dem fragenden Requirements Engineer und dem antwortenden Spezialisten umgekehrt wird, denn der Stakeholder nimmt nun die Rolle des »Meisters« ein, der Wissen hat, das dem Lehrling noch fehlt.

- *Artefaktbasierte Techniken*

Artefaktbasierte Techniken verwenden bestehende Entwurfs- und Entwicklungsartefakte wieder, um daraus Anforderungen an das neue System zu gewinnen. So können beispielsweise Lösungen und Erfahrungen bestehender Systeme als Ausgangspunkt für die Erhebung weiterer Anforderungen genutzt werden. Zu den artefaktbasierten Techniken zählen u.a. Systemarchäologie, Feedbackanalyse und die Wiederverwendung von Anforderungen aus Katalogen.

#### **4.2.2.2 Entwurfs- und Ideenfindungstechniken**

Entwurfs- und Ideenfindungstechniken zielen größtenteils darauf ab, Begeisterungsfaktoren zu kreieren, indem sie Kreativität fördern. Sie sind Kreativitätstechniken, die neue oder innovative Anforderungen erheben, um innovative Lösungsideen für das vorliegende Problem oder ein herausforderndes Ziel zu finden. Bekannte Techniken sind z. B. Brainstorming [Osborn 1979] und Analogietechniken.

##### **Exkurs: Kreativitätstechniken**

Kreativitätstechniken dienen dazu, innovative Anforderungen zu entwickeln, die erste Vision eines neuen Systems festzulegen und Begeisterungsfaktoren zu ermitteln. Kreativitätstechniken eignen sich allerdings in der Regel nicht dazu, detaillierte Anforderungen an das Systemverhalten herauszuarbeiten. Die folgenden Kreativitätstechniken sind sehr verbreitet [Maiden und Gizikis 2001]:

- Beim *Brainstorming* werden in einer Gruppe von 5 bis 10 Personen in vorgegebener Zeit Ideen gesammelt und zunächst ohne Beurteilung und Kommentierung von einem Moderator notiert. Die Teilnehmer nutzen die Ideen der anderen Teilnehmenden, um neue eigene Ideen zu entwickeln bzw. bestehende Ideen weiterzuentwickeln. Erst anschließend werden die Ideen einer sorgfältigen Analyse unterzogen. Besonders effektiv ist die Technik, wenn viele Personen unterschiedlicher Stakeholder-Gruppen beteiligt sind. Zu den Vorteilen

dieser Technik zählt die Identifikation vieler Ideen in kurzer Zeit, wobei mehrere Personen gemeinsam diese Ideen weiterentwickeln. Die unreflektierte Sammlung der Ideen lässt zudem neue Lösungen entstehen. Weniger effektiv ist Brainstorming bei schwieriger Gruppendynamik oder unterschiedlich dominanten Teilnehmern. Für solche Situationen existieren andere Kreativitätstechniken wie z. B. die 6-3-5-Methode (6 Teilnehmer, je 3 Ideen, 5-mal weiterreichen) [Rohrbach 1969] oder das Brainwriting.

- *Brainstorming paradox* ist eine Modifikation des Brainstormings, bei der Ereignisse gesammelt werden, die nicht erreicht werden sollen. Anschließend werden Maßnahmen entwickelt, wie die gefundenen Ereignisse verhindert werden können. Den Teilnehmerinnen und Teilnehmern wird dabei oft bewusst, welche Aktionen zu negativen Resultaten führen. So können Risiken erkannt und geeignete Gegenmaßnahmen entwickelt werden. Die Vor- und Nachteile dieser Technik sind mit denen des klassischen Brainstormings identisch.
- *Wechsel der Perspektive*  
Bei den Techniken, die mit Perspektivenwechseln arbeiten (der Einnahme unterschiedlicher Extrempositionen), ist die bekannteste Technik das sogenannte Sechs-Hut-Denken [DeBono 2006]. Jeder der sechs Hüte steht für eine andere Sichtweise, die von jedem Teilnehmer einmal eingenommen werden kann. Die resultierenden Lösungsansätze betrachten das Problem dadurch von mehreren Seiten. Auch von ihrer Sicht überzeugte Stakeholder werden so animiert, eine neue Sichtweise einzunehmen. Diese Technik ist hervorragend geeignet, wenn Stakeholder aufgrund einer gewohnten, stark eingeengten Sichtweise ihr Wissen nur sehr einseitig formulieren können, jedoch nicht, wenn die Anforderungen eine tiefe Detaillierungsebene erfordern, da die Technik sonst sehr aufwendig ist.
- *Analogietechniken (Bionik/Bisoziation)*  
In der Bionik werden ähnliche Problemstellungen (und Lösungen) anhand analoger Strukturen in der Natur gesucht, in der Bisoziation sind auch Analogien außerhalb der Natur möglich. Voraussetzung hierfür ist die Fähigkeit der Beteiligten zum Denken in Analogien, viel Zeit und eine tiefe Fachkenntnis des Bereichs, mit dem die Analogie gezogen werden soll.

Entwurfs- und Ideenfindungstechniken beinhalten unter anderem Kreativitätstechniken zur Findung von Ideen, bieten aber zusätzliche Techniken zur Ausarbeitung von Ideen und deren Konsolidierung sowie zur Gewinnung weiterer Erkenntnisse [Kumar 2013]. Zu den gängigen Techniken für diesen Zweck

gehören die Erstellung von Prototypen (z.B. Mock-ups, Wireframes), Szenarien und Storyboards.

### Exkurs: Szenarien

Szenarien stellen ein einfaches und effektives Mittel der Anforderungserhebung dar. Szenarien beschreiben typische Anwendungsfälle auf hohem Detailgrad. Das heißt, es wird nicht ein abstrakter Ablauf beschrieben, sondern eine konkrete Ausprägung. Dies birgt Vorteile in der Kommunikation mit den Stakeholdern, da diese direkt in ihrer Lebens- bzw. Arbeitsrealität abgeholt werden und den Sachverhalt nachvollziehen können. So kann direkt sehr detailliert geklärt werden, ob der beschriebene Ablauf dem vom Stakeholder erdachten Sachverhalt entspricht.

Häufig wird zwischen präskriptiven und deskriptiven Szenarien unterschieden. Deskriptive Szenarien beschreiben einen Sachverhalt wie er aktuell stattfindet. Dies ist ein gängiger Ausgangspunkt in der Ermittlung von Anforderungen: Zunächst wird die Ist-Situation möglichst detailgenau aufgenommen, um anschließend Verbesserungen daraus abzuleiten. Präskriptive Szenarien beschreiben einen Ablauf in einer gewünschten zukünftigen Realität, d.h. unter Nutzung des zu entwickelnden Systems.

Zur Beschreibung von Szenarien eignen sich narrative textuelle Beschreibungen genauso wie modellbasierte Dokumentationsformen. Textuelle Beschreibungen sind häufig in Listenform zu finden, die die einzelnen Schritte des Szenarios ausformulieren. Für modellbasierte Darstellungen wird oftmals auf UML-Sequenzdiagramme zurückgegriffen.

Des Weiteren existieren Frameworks von Techniken, die ebenfalls für die Ermittlung kreativer Ideen geeignet sind. So ist Design Thinking nicht primär eine Entwurfs- und Ideenfindungstechnik, kann aber auch als solche genutzt werden, um neue Anforderungen zu gewinnen. Hierzu existieren verschiedene Ansätze wie z. B. *d.school* [Stanford 2012] und *Designing for Growth* [Liedtka und Oglivie 2011].

#### 4.2.2.3 Auswahl der richtigen Ermittlungstechnik

Ermittlungstechniken müssen es dem Requirements Engineer ermöglichen, alle Arten von Anforderungen (d.h. funktionale Anforderungen, Qualitätsanforderungen und Randbedingungen) zu erheben.

Leider werden in der Realität gerne die Qualitätsanforderungen und Randbedingungen vernachlässigt. Bei deren Erhebung helfen Checklisten oder die Orientierung an einer Norm, wie z. B. der Norm ISO 25010 [ISO/IEC 25010:2011], die auch bei der Quantifizierung der entsprechenden Anforderungen helfen kann, wenn sie als Checkliste genutzt wird.

Um keine Randbedingungen zu vergessen, sollten Sie systematisch alle vorhandenen Einschränkungen ihres Lösungsraums durchdenken und daraus die entsprechenden Anforderungen ableiten. Beachten Sie dabei technische, rechtliche, organisatorische, kulturelle oder umweltbezogene Einschränkungen.

Meist ist eine Kombination unterschiedlicher Ermittlungstechniken von Vorteil, da jede Technik ihre Stärken und Schwächen hat. Die Auswahl der richtigen Ermittlungstechniken hängt von vielen unterschiedlichen Faktoren ab, z. B.:

- *Art des Systems*  
Beispielsweise kann bei einer Weiterentwicklung auf einer bereits bestehenden Anforderungsspezifikation und Dokumentation aufgesetzt werden, während dies bei einer Neuentwicklung nicht möglich ist.
- *Softwareentwicklungs-Lebenszyklusmodell*  
Wird agil entwickelt oder nach einem traditionellen Vorgehensmodell?  
Unterschiedliche Entwicklungsprozesse haben verschiedene Implikationen auf die Anforderungserhebungen.
- *Beteiligte Personen*  
Der Kenntnisstand der Stakeholder und andere Eigenschaften, wie deren Verfügbarkeit, wirken sich unmittelbar auf die Eignung der eingesetzten Ermittlungstechniken aus.
- *Organisatorischer Aufbau*  
Wie wird entwickelt? Ein zentral verfügbares Team bedeutet einen anderen Ansatz zur Anforderungserhebung als eine weltweite Verteilung der Entwickler, da beispielsweise physische Workshops nicht durchgeführt werden können.

#### **Kernfakten 4-4:** Ermittlung von Anforderungen



[www.cpre-buch.de/pk4w4](http://www.cpre-buch.de/pk4w4)

### **4.3 Abstimmung und Konfliktlösung**

Auch bei einer gewissenhaften Ermittlung von Anforderungen kann es vorkommen, dass diese Anforderungen im Widerspruch stehen. Insbesondere wird nicht gewährleistet, dass verschiedene Stakeholder eine Anforderung gleich verstehen oder diese gleich wertschätzen. Besteht bezüglich der Anforderungen unter den Stakeholdern ein Widerspruch, sodass die Anforderungen nicht gemeinsam in einem System umgesetzt werden

*Widersprüchliche  
Anforderungen erzeugen  
Konflikte.*

können, entsteht ein Konflikt zwischen den widersprüchlichen Anforderungen und damit zwischen den Stakeholdern, die diese Anforderungen fordern. Zum Beispiel könnte ein Stakeholder fordern, dass sich das zu erstellende System im Fehlerfall abschaltet, wohingegen ein anderer Stakeholder verlangen könnte, dass das System im Fehlerfall neu startet.

Die Akzeptanz eines geplanten Systems wird durch unaufgelöste Konflikte gefährdet, da sie dazu führen, dass nicht alle Anforderungen aller Stakeholder umgesetzt werden können. Im schlimmsten Fall kann ein unbeachteter Konflikt der Auslöser sein, dass die Entwicklung eines Systems nicht weiter durch die betroffenen Stakeholder unterstützt wird und dadurch die Entwicklung gänzlich scheitert (z.B. [Easterbrook 1994]). Neben den genannten Risiken sind Konflikte allerdings auch eine Chance für das Requirements Engineering, da Konflikte zwischen Stakeholdern eine Lösung erfordern, die unter Umständen auch zur Entwicklung von neuen Ideen beiträgt oder mögliche Optionen in der Entwicklung aufzeigen kann (z.B. [Gause und Weinberg 1989]). Eine offene Behandlung und Auflösung von Konflikten unter Beteiligung der Stakeholder kann die Akzeptanz eines Systems sogar erhöhen.

*Risiken und Chancen von Konflikten*

Um unter den relevanten Stakeholdern ein gemeinsames und übereinstimmendes Verständnis bezüglich der Anforderungen an das zu entwickelnde System herbeizuführen, ist die Abstimmung von Anforderungen eine wichtige Aufgabe des Requirements Engineer.

*Ziel der Abstimmung über Anforderungen*

Die Überprüfung auf Widersprüche und Abstimmung der Anforderungen muss fortlaufend (in unterschiedlicher Intensität) über das gesamte Requirements Engineering hinweg erfolgen. Die Überprüfung und Abstimmung von Anforderungen verursachen dabei zusätzlichen Aufwand und somit zusätzliche Kosten. Der durch die Überprüfung und Abstimmung der Anforderungen erzielte und in den vorangegangenen Abschnitten beschriebene Vorteil (Kostensparnis, Erhöhung der Akzeptanz des Systems, Unterstützung der Definition innovativer Anforderungen) ist in der Regel jedoch wesentlich höher als die durch die Überprüfung und Abstimmung entstehenden Kosten.

*Verringerung der Kosten und Risiken in späteren Phasen*

Zur Abstimmung der Anforderungen an ein zu entwickelndes System ist es notwendig, Konflikte zu identifizieren und die auftretenden Konflikte aufzulösen. Dies geschieht im Rahmen eines systematischen Konfliktmanagements. Das Konfliktmanagement im Requirements Engineering umfasst die folgenden vier Aufgaben:

- Konfliktidentifikation

*Vier Aufgaben des Konfliktmanagements*

- Konfliktanalyse
- Konfliktlösung
- Dokumentation der Konfliktlösung

Diese vier Aufgaben des Konfliktmanagements werden im Folgenden erläutert.

### 4.3.1 Konfliktidentifikation

Konflikte zwischen Stakeholdern können in allen Requirements-Engineering-Aktivitäten auftreten. Zum Beispiel können unterschiedliche Stakeholder im Rahmen der Ermittlung von Anforderungen sich gegenseitig ausschließende Anforderungen äußern. Oft fällt ein solcher Widerspruch bei der Dokumentation der Anforderungen auf, wenn die Anforderungen in ein gemeinsames Kapitel integriert werden sollen.

Konflikte zwischen Anforderungen (bzw. zwischen Stakeholdern) sind manchmal nicht direkt erkennbar. Dafür gibt es verschiedene Gründe. Beispielsweise könnte mehr als ein Requirements Engineer an der Ermittlung von Anforderungen beteiligt sein. Dennoch sollte der Requirements Engineer während des gesamten Requirements Engineering auf potenzielle Konflikte achten, damit diese frühzeitig entdeckt, analysiert und aufgelöst werden können.

*Konflikterkennung in allen Requirements-Engineering-Aktivitäten*

### 4.3.2 Konfliktanalyse

Im Rahmen der Konfliktanalyse wird ein identifizierter Konflikt auf seine Ursachen hin untersucht. Um im späteren Verlauf den Konflikt auflösen zu können, muss zuerst die jeweilige Konfliktursache identifiziert werden. In Anlehnung an [Moore 2014] unterscheiden wir zwischen sechs Konflikttypen. Abbildung 4-3 gibt einen Überblick über diese sechs Konflikttypen.

*Bestimmung des Konflikttyps*



**Abb. 4-3**  
Konflikttypen

Ein *Sachkonflikt* zwischen zwei oder mehr Stakeholdern ist durch einen Mangel an Informationen oder durch Fehlinformation gekennzeichnet. Als Beispiel für einen Sachkonflikt betrachten wir die Anforderung: »R131: Die Antwortzeit des geplanten Systems soll höchstens eine Sekunde betragen.« Ein Sachkonflikt zwischen zwei Stakeholdern bezüglich dieser Anforderung kann dadurch entstehen, dass ein Stakeholder die Antwortzeit von einer Sekunde für zu langsam erachtet und ein anderer Stakeholder die Antwortzeit für nicht realisierbar, d.h. für zu kurz hält.

*Sachkonflikt*

Eine spezielle Ausprägung des Sachkonflikts ist der *Datenkonflikt* (auch Benennungskonflikt). Bei einem Datenkonflikt verstehen die beteiligten Stakeholder unterschiedliche Dinge unter einem Bezeichner. Beispielsweise mögen zwei Stakeholder unterschiedlicher Auffassung bezüglich der Endgültigkeit des Begriffs »löschen« sein. Dies kann beispielsweise als unwiederbringliche Aktion aufgefasst werden oder als einfache rückverfolgbare Änderung eines Sachverhalts.

*Datenkonflikt*

Ein *Interessenkonflikt* zwischen zwei oder mehr Stakeholdern ist durch subjektiv oder objektiv verschiedene Interessen oder Ziele der Stakeholder gekennzeichnet. Ein Interessenkonflikt zwischen zwei oder mehr Stakeholdern kann z. B. dadurch entstehen, dass ein Stakeholder primär geringe Kosten des geplanten Systems anstrebt und ein anderer Stakeholder eine hohe Qualität des geplanten Systems zum Ziel hat. Ein Interessenkonflikt zwischen diesen beiden Stakeholdern tritt dann auf, wenn der erste Stakeholder eine Anforderung aus Kostengründen ablehnt, der zweite Stakeholder die Anforderung aus Qualitätsgründen umsetzen möchte.

*Interessenkonflikt*

Ein *Wertekonflikt* ist durch verschiedene Kriterien (z.B. kulturelle Unterschiede, persönliche Ideale) von Stakeholdern zur Bewertung von Sachverhalten gekennzeichnet. Ein Wertekonflikt besteht z. B., wenn ein Stakeholder Open-Source-Technologie favorisiert und ein anderer Stakeholder jedoch Closed-Source-Technologie bevorzugt.

*Wertekonflikt*

Ein *Beziehungskonflikt* ist durch starke Emotionen, stereotype Beziehungskonzepte, schlechte Kommunikation oder negatives zwischenmenschliches Verhalten von Stakeholdern untereinander (z.B. Missachtung, Beleidigung) gekennzeichnet. Ein Beziehungskonflikt besteht z. B. zwischen zwei gleichrangigen Stakeholdern (z.B. Abteilungsleitern), wenn beide Stakeholder gegenseitig ihre Anforderungen ablehnen, um sich jeweils durch das Einbringen ihrer Anforderungen im Projekt zu profilieren.

*Beziehungskonflikt*

Ein *struktureller Konflikt* ist durch ungleiche Macht- und Autoritätsverhältnisse zwischen Stakeholdern gekennzeichnet. Ein struktureller Konflikt besteht z. B. zwischen einem Mitarbeiter und seiner Vorgesetzten, wenn die Vorgesetzte prinzipiell Anforderungen ablehnt, die ihr Mitarbeiter definiert, da sie ihm die Kompetenz zur Definition von Anforderungen abspricht.

*Struktureller Konflikt*

Aufgetretene Konflikte können häufig nicht eindeutig einem der oben vorgestellten Konflikttypen zugeordnet werden. So kann z. B. ein Konflikt sowohl einen Beziehungs- als auch einen Strukturanteil haben. Ebenso kann beispielsweise ein Interessenkonflikt gleichzeitig auch einen Wertekonflikt beinhalten. Es empfiehlt sich daher, einen identifizierten Konflikt auf alle genannten Konflikttypen hin zu untersuchen, um alle möglichen Ursachen eines Konflikts zu erkennen und somit geeignete Auflösungsstrategien wählen zu können.

*Vermischte Konfliktursachen*

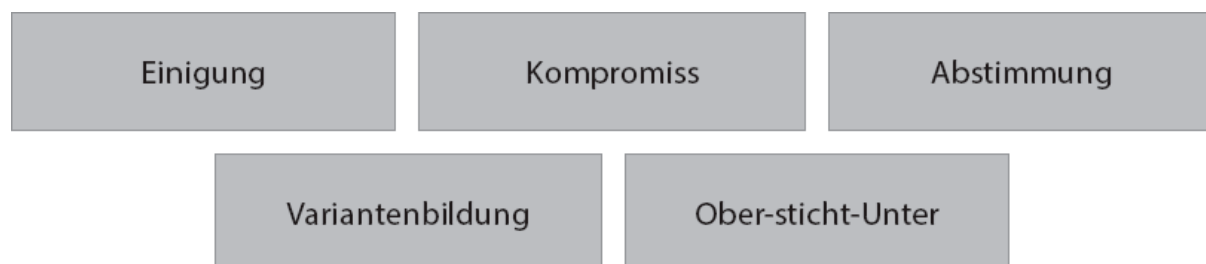
### 4.3.3 Konfliktlösung

Der Konfliktlösung kommt eine große Bedeutung zu, da die Art und Weise der Konfliktlösung großen Einfluss auf die weitere Kooperationsbereitschaft der beteiligten Konfliktparteien (z.B. Kunden, Qualitätssicherer oder Entwickler) haben kann. Zum Beispiel kann eine Konfliktlösung, die von einer Konfliktpartei als unfair empfunden wird, dazu führen, dass die Kooperationsbereitschaft und das Engagement dieser Konfliktpartei für das geplante System sinken. Ebenso kann eine als fair empfundene Konfliktlösung die Kooperationsbereitschaft natürlich auch verbessern, da den Konfliktparteien dadurch signalisiert wird, dass ihre jeweiligen Vorstellungen in Bezug auf das geplante System ernst genommen werden.

*Gute Konfliktlösung ist ein Erfolgsfaktor.*

Unabhängig von der gewählten Konfliktlösungstechnik ist die Einbeziehung aller am Konflikt beteiligten Stakeholder essenziell. Werden nicht alle relevanten Stakeholder einbezogen, so bleiben deren Meinungen und Standpunkte bei der Auflösung unbeachtet. Der Konflikt wird dann nur teilweise und somit unvollständig aufgelöst. Abbildung 4-4 gibt einen Überblick über gängige Techniken zur Konfliktlösung.

*Einbeziehung aller relevanten Stakeholder*



#### **Abb. 4-4**

##### *Konfliktlösungstechniken*

Bei der Konfliktlösungstechnik *Einigung* handeln die Konfliktparteien eine Lösung des Konflikts aus. Die Konfliktparteien tauschen Informationen, Argumente und Meinungen aus und versuchen sich gegenseitig im Dialog von der Richtigkeit des eigenen Standpunkts zu überzeugen und sich so auf eine Lösungsalternative des Konflikts zu einigen.

*Einigung*

Bei der Konfliktlösungstechnik *Kompromiss* versuchen die Konfliktparteien im Rahmen einer Diskussion einen Kompromiss zwischen den verfügbaren Lösungsalternativen zu finden. Im Unterschied zur Einigung besteht ein Kompromiss aus einer Kombination von Teilen der verfügbaren Lösungsalternativen. Ebenso kann ein Kompromiss auch darin bestehen, dass alle Lösungsalternativen verworfen werden und eine vollkommen neue und kreative Lösung entwickelt wird.

*Kompromiss*

Bei der Konfliktlösungstechnik *Abstimmung* wird die Lösung eines Konflikts durch eine Abstimmung erzielt. Die zur Wahl stehenden Alternativen werden den relevanten Stakeholdern zur Abstimmung vorgelegt. Jeder Stakeholder gibt seine Stimme einer der Alternativen. Die Alternative mit den meisten Stimmen wird als Konfliktlösung festgehalten.

*Abstimmung*

Bei der Konfliktlösungstechnik *Variantenbildung* wird das System so gestaltet, dass durch Variantenauswahl oder Parametrierung verschiedene Systemvarianten realisiert oder Auswahlmöglichkeiten bei variablen Systemmerkmalen ermöglicht werden, wodurch das System unterschiedlichen, im Konflikt stehenden Interessen von Stakeholdern genügen kann.

*Variantenbildung*

Bei der Konfliktlösungstechnik *Ober-sticht-Unter* wird ein Konflikt anhand der Hierarchie der Konfliktparteien entschieden, d.h., die Konfliktpartei mit dem höheren organisatorischen Rang gewinnt den Konflikt. Wenn beide Konfliktparteien den gleichen organisatorischen Rang einnehmen, wird der Konflikt durch eine übergeordnete Instanz (z.B. einen Vorgesetzten) entschieden. Diese Konfliktlösungstechnik ist nur dann empfehlenswert, wenn andere Lösungstechniken zu keiner Lösung geführt haben (z.B. kein Kompromiss gefunden werden konnte) oder aus Ressourcengründen nicht anwendbar sind.

*Ober-sticht-Unter*

Neben den Konfliktlösungstechniken gibt es weitere Unterstützungstechniken zur Konfliktlösung. Nachfolgend stellen wir die wichtigsten vor:

- Consider-all-Facts

- Plus-Minus-Interesting
- Entscheidungsmatrix

Bei der Unterstützungstechnik *Consider-all-Facts* (CAF) werden möglichst alle Einflussfaktoren eines Konflikts untersucht, um so viel Informationen wie möglich über den Konflikt zu sammeln und in die Konfliktlösung einzubeziehen. Durch eine Priorisierung der identifizierten Einflussfaktoren wird die Relevanz der Einflussfaktoren bestimmt. Basierend auf den Ergebnissen der Konfliktlösungstechnik *Consider-all-Facts* (CAF) kann die Plus-Minus-Interesting-Konfliktlösungstechnik angewendet werden.

*Consider-all-Facts*

Bei der Unterstützungstechnik *Plus-Minus-Interesting* (PMI) werden alle positiven und negativen Folgen der zur Wahl stehenden Lösungsalternativen untersucht, um positive und negative Auswirkungen der Lösungsalternativen besser beurteilen zu können. Auswirkungen, die als positiv angesehen werden, werden in die Kategorie Plus eingeordnet. Negative Auswirkungen werden der Kategorie Minus zugeordnet. Auswirkungen, die weder als positiv noch als negativ bewertet werden können, werden in die Kategorie Interesting aufgenommen. Diese Einordnung zeigt an, dass die Bedeutung dieser Auswirkungen noch nicht bewertbar ist und ggf. noch weiter untersucht bzw. diskutiert werden muss.

*Plus-Minus-Interesting*

Bei der Unterstützungstechnik *Entscheidungsmatrix* wird eine Tabelle erstellt. In den Spalten der Tabelle werden alle Lösungsalternativen eines Konflikts eingetragen. Die Zeilen der Tabelle enthalten alle relevanten Entscheidungskriterien. Die Entscheidungskriterien können beispielsweise durch die Technik *Consider-all-Facts* identifiziert werden. Für jede Kombination aus Kriterium und Lösungsalternative wird eine Bewertung abgegeben, z. B. durch eine Punktebewertung von irrelevant (0 Punkte) bis sehr relevant (10 Punkte). Tabelle 4-2 zeigt eine Entscheidungsmatrix.

*Entscheidungsmatrix*

**Tab. 4-2**  
*Entscheidungsmatrix*

	<b>Lösungsalternative 1</b>	<b>Lösungsalternative 2</b>	<b>Lösungsalternative 3</b>
<b>Kriterium 1</b>	3	6	2
<b>Kriterium 2</b>	5	4	10
<b>Kriterium 3</b>	10	3	5
<b>Summe</b>	<b>18</b>	<b>13</b>	<b>17</b>

Zur Entscheidungsfindung werden die Spaltensummen der Bewertung gebildet, d.h., die Bewertung der Kriterien einer Lösungsalternative wird aufsummiert. Die Lösungsalternative mit den meisten Punkten wird als Entscheidung festgehalten. Für das Beispiel aus Tabelle 4–2 ist dies die Lösungsalternative 1.

#### 4.3.4 Dokumentation der Konfliktlösung

Die Auflösung eines Konflikts sollte auf jeden Fall nachvollziehbar dokumentiert werden, da ansonsten u.a. die folgenden Gefahren für das Projekt entstehen:

*Risiken einer fehlenden  
Konfliktdokumentation*

- *Wiederholte Behandlung von Konflikten*  
Ein Konflikt kann im Verlauf des Requirements-Engineering-Prozesses erneut entstehen. Ohne eine geeignete Dokumentation einer bereits erfolgten Auflösung des Konflikts muss der Konflikt von Neuem analysiert und aufgelöst werden. Dies erzeugt zusätzlichen Aufwand und kann unter Umständen zu neuen Konflikten führen bzw. bisherige Konfliktlösungen zurückwerfen.
- *Überprüfung von Konfliktlösungen*  
Die Auflösung eines Konflikts kann sich im Verlauf des Requirements-Engineering-Prozesses als falsch bzw. als nicht geeignet herausstellen. In diesem Fall ist der Konflikt erneut zu untersuchen und aufzulösen. Ohne eine geeignete Dokumentation können relevante Informationen, die bei der Analyse und Auflösung des Konflikts in der Vergangenheit berücksichtigt wurden, übersehen werden, und somit kann die Konfliktlösung zu fehlerhaften Ergebnissen führen.

In beiden Fällen unterstützt eine geeignete Dokumentation des Konflikts und seiner Auflösung den Requirements-Engineering-Prozess und gewährleistet, dass bereits bekannte relevante Informationen berücksichtigt werden können. Nachfolgender Weblink veranschaulicht am Beispiel, wie die Dokumentation eines Konflikts mit einer Schablone einfach und nachvollziehbar erfolgen kann.



[www.cpre-buch.de/pk4a1](http://www.cpre-buch.de/pk4a1)





## 4.4 Validieren von Anforderungen

Die Prüfung von Anforderungen im Requirements Engineering soll sicherstellen, dass die dokumentierten Anforderungen festgelegten Qualitätskriterien genügen. Die vorgestellten Prinzipien und Techniken können dabei zur Prüfung und Abstimmung einzelner Anforderungen, aber auch zur Prüfung und Abstimmung von Anforderungsdokumenten eingesetzt werden.

### 4.4.1 Grundlagen der Validierung von Anforderungen

Im Rahmen der Validierung von Anforderungen wird die Entscheidung getroffen, ob eine Anforderung die nötige Qualität aufweist (siehe Abschnitt 3.8) und ob die Anforderung für weitere Entwicklungsaktivitäten (Entwurf, Realisierung und Test) freigegeben werden kann. Diese Entscheidung sollte anhand von vorher festgelegten Prüf- und Abnahmekriterien erfolgen.

*Freigabe von Anforderungen*

#### **Definition 4–1:** Validierung

Der Prozess zur Bestätigung, dass ein Element (ein System, ein Arbeitsprodukt oder ein Teil davon) den Bedürfnissen und Wünschen seiner Stakeholder entspricht.

Übersetzt aus [IREB-Glossar 2020]

Das Ziel der Überprüfung von Anforderungen ist es somit, Fehler in den dokumentierten Anforderungen zu entdecken. Die Anforderungen werden dabei u.a. den Stakeholdern mit dem Ziel vorgelegt, Abweichungen zwischen den definierten Anforderungen und ihren Wünschen bzw. Bedürfnissen zu identifizieren. Typische Beispiele für Fehler in Anforderungen sind Mehrdeutigkeit, Unvollständigkeit und Widersprüche.

*Ziel der Überprüfung*

Die Validierung ist wichtig, da Anforderungsdokumente Referenzdokumente für alle weiteren Entwicklungsaktivitäten sind. Deshalb beeinträchtigen Fehler in Anforderungen natürlich auch alle weiteren Entwicklungsaktivitäten. Ein Anforderungsfehler, der erst im Betrieb des erstellten Systems identifiziert wird, erfordert die Überarbeitung aller Artefakte, die von dem Fehler betroffen sind, wie

*»Fehlerfortpflanzung«*

beispielsweise Quellcode, Testartefakte oder Architekturbeschreibungen. Die Beseitigung solcher Anforderungsfehler verursacht daher oftmals erhebliche Kosten.

Weiterhin existieren rechtliche Risiken, denen mittels einer Validierung entgegengewirkt werden soll. Zum Beispiel wird ein Vertrag zwischen Auftraggeber und Auftragnehmer häufig auf Basis von Anforderungsdokumenten geschlossen. Kritische Fehler in Anforderungen können daher dazu führen, dass durch einen Vertrag getroffene Vereinbarungen nicht eingehalten werden können, wie z. B. der Leistungsumfang, die Qualität oder der Zeitpunkt der Fertigstellung.

*Rechtliche Risiken*

### **Exkurs: Qualitätsaspekte von Anforderungen**

Neben einer Strukturierung anhand der aus Abschnitt 3.8 bekannten Qualitätskriterien eignen sich weitere Strukturierungen, um die Prüfung der Anforderungen zu erleichtern [Pohl 2008]. Beispielsweise erlaubt die Prüfung auf verschiedene Qualitätsaspekte eine Unterscheidung hinsichtlich:

- Inhalt einer Anforderung
- Dokumentationsformat einer Anforderung
- Abgestimmtheit einer Anforderung

Eine Anforderung sollte nur dann für nachfolgende Entwicklungsaktivitäten freigegeben werden, wenn alle drei Qualitätsaspekte geprüft wurden. Die Qualitätsaspekte werden im Folgenden detailliert beschrieben und (ohne Anspruch auf Vollständigkeit) durch verschiedene Qualitätskriterien konkretisiert.

### **Exkurs: Qualitätsaspekt »Inhalt«**

Der Qualitätsaspekt »Inhalt« bezieht sich auf die Überprüfung von Anforderungen auf inhaltliche Fehler. Inhaltliche Fehler in einer Anforderung führen dazu, dass nachfolgende Entwicklungsaktivitäten auf falschen Informationen basieren.

Inhaltliche Fehler sind gegeben, wenn spezifische Qualitätskriterien für Anforderungen oder für Anforderungsdokumente verletzt sind. Die Überprüfung von Anforderungen bezüglich des Qualitätsaspekts Inhalt ist erfolgreich, wenn eine Überprüfung der Anforderungen auf folgende Fehlerarten durchgeführt wurde und keine signifikanten Mängel entdeckt wurden:

- *Vollständigkeit (Menge aller Anforderungen)*  
Sind alle relevanten Anforderungen an das geplante System (für das nächste Systemrelease) erfasst?
- *Vollständigkeit (einzeln Anforderung)*  
Beschreibt jede Anforderung alle für diese Anforderung notwendigen Informationen?
- *Verfolgbarkeit*  
Sind alle relevanten Verfolgbarkeitsbeziehungen definiert (z.B. zu relevanten Anforderungsquellen)?
- *Korrektheit/Adäquatheit*  
Spiegeln die Anforderungen die Bedürfnisse und Wünsche der Stakeholder in angemessener Weise wider?
- *Konsistenz*  
Sind alle definierten Anforderungen an das geplante System gemeinsam erfüllbar bzw. stehen die Anforderungen nicht miteinander in Widerspruch?
- *Keine vorzeitigen Entwurfsentscheidungen*  
Wurden Entwurfsentscheidungen in den Anforderungen vorweggenommen, die nicht durch Randbedingungen induziert sind (z. B. Randbedingung, die die Verwendung einer Client/Server-Architektur vorschreibt)?
- *Überprüfbarkeit*  
Können basierend auf den Anforderungen Abnahme- bzw. Prüfkriterien definiert werden bzw. sind diese bereits definiert?
- *Notwendigkeit*  
Trägt jede Anforderung zur Erfüllung eines definierten Ziels bei?

### **Exkurs: Qualitätsaspekt »Dokumentation«**

Der Qualitätsaspekt »Dokumentation« betrifft die Überprüfung von Anforderungen auf Mängel in der Dokumentation bzw. auf Verstöße gegen geltende Dokumentationsvorschriften, wie z. B. Verständlichkeit der verwendeten Dokumentationsformate oder die Erfüllung organisatorischer oder projektspezifischer Richtlinien bezüglich der Dokumentation von Anforderungen sowie der Struktur von Anforderungsdokumenten.

Die Missachtung von Dokumentationsvorschriften führt u.a. zu folgenden Risiken:

- *Behinderung von Entwicklungsaktivitäten*  
Entwicklungsaktivitäten, die auf einem festgelegten Dokumentationsformat beruhen, können nicht durchgeführt werden.
- *Unverständlichkeit von Anforderungen*  
Die Anforderungen sind für die Adressaten nicht verwendbar bzw. nicht verständlich.
- *Unvollständigkeit*  
Relevante Informationen sind nicht in den Anforderungen dokumentiert.
- *Übersehen von Anforderungen*  
Sind Anforderungen nicht an der vorgeschriebenen Stelle im Anforderungsdokument definiert, so können diese Anforderungen leicht von Stakeholdern der Folgeaktivitäten übersehen werden.

Die Überprüfung von Anforderungen bezüglich des Qualitätsaspekts Dokumentation ist erfolgreich, wenn eine Überprüfung der Anforderungen bzw. des Anforderungsdokuments auf folgende Fehlerarten durchgeführt wurde und keine signifikanten Mängel entdeckt wurden:

- *Konformität zum Dokumentationsformat und zur Dokumentenstruktur*  
Wurden die Anforderungen in dem vorgeschriebenen Dokumentationsformat dokumentiert? Wurde beispielsweise die vorgegebene Schablone oder die vorgegebene Modellierungssprache zur Dokumentation der Anforderungen verwendet? Wurde die Struktur des Anforderungsdokuments eingehalten? Wurden alle Anforderungen an der dafür vorgesehenen Stelle dokumentiert?
- *Verständlichkeit*  
Können die dokumentierten Anforderungen in dem gegebenen Kontext verstanden werden? Wurden die verwendeten Begriffe in einem Glossar definiert (vgl. Abschnitt 3.5)?
- *Eindeutigkeit*  
Lässt die Dokumentation der Anforderungen eine eindeutige Interpretation zu oder sind mehrere unterschiedliche Interpretationen möglich? Ist beispielsweise die textuelle Anforderung sprachlich eindeutig?
- *Konformität mit Dokumentationsregeln*  
Sind die festgelegten Dokumentationsregeln und Dokumentationsrichtlinien beachtet worden? Wurde beispielsweise die Syntax der vorgegebenen Modellierungssprache eingehalten?

### Exkurs: Qualitätsaspekt »Abgestimmtheit«

Der Qualitätsaspekt »Abgestimmtheit« bezieht sich auf die Überprüfung von Anforderungen auf Mängel in der Abstimmung der Anforderungen unter den relevanten Stakeholdern.

Stakeholder erwerben im Verlauf des Requirements-Engineering-Prozesses Wissen über das geplante System. Durch das zusätzliche Wissen kann sich die Meinung von Stakeholdern über bereits abgestimmte Anforderungen ändern. Im Rahmen der Prüfung von Anforderungen haben Stakeholder die Möglichkeit, Änderungswünsche zu äußern, ohne die nachfolgenden Entwicklungsaktivitäten zu beeinträchtigen.

Die Überprüfung von Anforderungen bezüglich des Qualitätsaspekts Abgestimmtheit ist erfolgreich, wenn eine Überprüfung der Anforderungen auf folgende Fehlerarten durchgeführt wurde und keine signifikanten Mängel entdeckt wurden:

- *Abstimmung*  
Wurde jede Anforderung mit allen relevanten Stakeholdern abgestimmt?
- *Abstimmung nach Änderungen*  
Wurde für jede geänderte Anforderung die Zustimmung der Stakeholder nach der Änderung erneut eingeholt?
- *Konflikte aufgelöst*  
Wurden alle bekannten Konflikte bezüglich der Anforderungen aufgelöst?

#### 4.4.2 Wichtige Aspekte der Anforderungvalidierung

Die Beachtung der folgenden Aspekte der Anforderungvalidierung verbessert die Qualität der Überprüfungsergebnisse:

- *Beteiligung der richtigen Stakeholder*  
Die Auswahl von Stakeholdern für die Prüfung von Anforderungen richtet sich nach dem Ziel der Prüfung sowie nach den zu prüfenden Anforderungen.

### Exkurs: Auswahl der Validierer

Bei der Zusammenstellung des Prüfungsteams/der Validierer sollten zumindest die nachfolgend erläuterten zwei Aspekte beachtet werden:

- *Unabhängigkeit des Prüfers*

Im Allgemeinen sollte vermieden werden, dass eine Anforderung durch ihren Ersteller überprüft wird. Der Ersteller einer Anforderung greift beim Lesen bzw. Betrachten einer Anforderung auf sein eigenes Wissen zurück. Dieses Wissen kann die Identifikation von Fehlern negativ beeinflussen, da potenziell fehlerhafte Stellen der Anforderung unbewusst durch eigenes Wissen ergänzt werden und dadurch übersehen werden können.

- *Interne vs. externe Prüfer*

Geeignete Prüfer für Anforderungen können innerhalb und außerhalb der entwickelnden Organisation identifiziert werden. Eine interne Prüfung wird von Stakeholdern durchgeführt, die innerhalb der entwickelnden Organisation stehen, und kann beispielsweise zur Vorprüfung von Zwischenständen der Anforderungen genutzt werden. Eine interne Prüfung ist leichter zu koordinieren und zu organisieren, da Stakeholder innerhalb der entwickelnden Organisation leichter verfügbar sind. Eine externe Prüfung wird von Stakeholdern durchgeführt, die außerhalb der entwickelnden Organisation stehen. Eine externe Prüfung erfordert einen größeren Aufwand, da diese Prüfer identifiziert und (ggf. kostenpflichtig) beauftragt werden müssen. Des Weiteren müssen sich externe Prüfer ggf. in den Kontext des geplanten Systems einarbeiten. Eine externe Prüfung sollte aufgrund des hohen Aufwands nur für Anforderungen mit einem möglichst hohen Qualitätsgrad durchgeführt werden.

- *Trennung von Fehlersuche und Fehlerkorrektur*

Die Trennung zwischen der Suche nach Fehlern und der eigentlichen Korrektur des Fehlers ist ein bewährtes Prinzip bei der Qualitätssicherung von Software. Das gleiche Prinzip kann auf die Prüfung von Anforderungen übertragen werden. Bei der Überprüfung werden die Mängel in den Anforderungen lediglich dokumentiert. Erst im Anschluss an die Überprüfung wird für jeden identifizierten Mangel geprüft, wie dieser behoben werden kann.

### **Exkurs: Konzentration auf Aufdeckung von Fehlern**

Die Trennung von Fehlersuche und Fehlerkorrektur erlaubt die Konzentration auf die Fehleridentifikation. Die Korrekturaktivitäten setzen erst im Anschluss an die Prüfung ein. Dies hat den Vorteil, dass die für die Fehlerkorrektur verfügbaren Ressourcen gezielt eingesetzt werden können und dass die frühzeitige Korrektur von Fehlern keine zusätzlichen Fehler in den

Anforderungen erzeugt bzw. vermeintliche Fehler korrigiert werden, die sich bei näherer Untersuchung nicht als Fehler herausstellen. Zudem wird vermieden, dass man sich bei der Überprüfung auf die Korrektur eines zuerst entdeckten Fehlers konzentriert und hierdurch das Risiko eingeht, eventuell vorhandene signifikante Fehler zu übersehen.

- *Validierung aus unterschiedlichen Sichten*  
Die Validierung von Anforderungen aus unterschiedlichen Sichten hat sich in der Praxis bewährt. Bei diesem Vorgehen werden Anforderungen jeweils aus einem bestimmten Blickwinkel (z.B. von unterschiedlichen Personen) überprüft bzw. abgestimmt.
- *Wiederholte Validierung*  
Die Überprüfungsaktivität findet zu einem bestimmten Zeitpunkt im Entwicklungsprozess statt und bezieht sich auf den Kenntnisstand der Prüfer zu diesem Zeitpunkt. Im Verlauf des Requirements-Engineering-Prozesses gewinnen Stakeholder neues Wissen über das geplante System. Daher ist eine positive Überprüfung von Anforderungen keine Garantie dafür, dass Anforderungen zu einem späteren Zeitpunkt ebenfalls gültig sind.

#### **4.4.3 Reviewtechniken zur Validierung von Anforderungen**

In diesem Abschnitt werden Techniken zur Validierung von Anforderungen vorgestellt. Häufig werden zur Validierung von Anforderungen manuelle Prüftechniken eingesetzt, die unter dem Oberbegriff des Reviews bekannt sind. Bei Reviews werden Walkthroughs und Inspektionen unterschieden.

##### **4.4.3.1 Walkthrough**

Ein Walkthrough zur Überprüfung von Anforderungen ist eine leichtgewichtige Ausprägung eines Reviews.

*Leichtgewichtiges Review*

Ein Walkthrough verläuft weniger strikt als eine Inspektion, und die beteiligten Rollen sind in geringem Maße differenziert. Bei einem Walkthrough sind mindestens die Rollen »Reviewer« (vergleichbar mit dem Inspektor), »Autor« und »Protokollant« und ggf. der Moderator zu besetzen.

Ein Walkthrough von Anforderungen zielt darauf ab, in einem gemeinsamen Prozess Qualitätsmängel in den Anforderungen zu identifizieren und unter den Teilnehmern ein konsolidiertes Verständnis der Anforderungen herzustellen. Zur Vorbereitung des Walkthrough werden die zu prüfenden Anforderungen an die Teilnehmer verteilt und von diesen auf

*Diskussion der identifizierten Qualitätsmängel in einer Gruppensitzung*

Qualitätsmängel hin untersucht. In der Walkthrough-Sitzung besprechen die Teilnehmer, angeleitet durch den Vorleser/den Moderator, schrittweise die zu überprüfenden Anforderungen. Der Autor kann dann zu den eigentlichen Anforderungen noch zusätzliche Informationen an die Gruppe geben (z.B. alternative Anforderungen, Entscheidungen und Begründungen für Entscheidungen). Die von den Reviewern im Vorfeld oder während der Sitzung identifizierten Qualitätsmängel werden im Rahmen der Sitzung diskutiert und durch den Protokollanten festgehalten.

#### **Exkurs: Stellungnahme als Sonderfall**

Die Stellungnahme ist eine weitere Vereinfachung eines Walkthrough. Bei dieser Prüftechnik übergibt der Autor seine Anforderungen an eine dritte Person (z.B. eine Kollegin), um von ihr eine Expertise bezüglich der Qualität der Anforderungen zu erhalten. Diese Person überprüft die Anforderungen dann mit dem Ziel, Qualitätsmängel in den Anforderungen (z.B. Mehrdeutigkeiten oder inhaltliche Fehler) hinsichtlich vorher festgelegter Qualitätskriterien zu identifizieren. Identifizierte Qualitätsmängel in den Anforderungen werden dabei im Dokument markiert und in komprimierter Form erläutert.

#### **4.4.3.2 Inspektion**

Inspektionen haben das Ziel, Entwicklungsartefakte anhand eines strikten Prozessschemas systematisch nach Fehlern zu durchsuchen [Laitenberger und DeBaud 2000].

Eine Inspektion ist typischerweise in die Phasen Planung, Übersicht, Fehlersuche, Fehlersammlung, Fehlerkorrektur, Nachkontrolle und Reflexion unterteilt [Gilb und Graham 1993], wobei für die Überprüfung von Anforderungen die Phasen Planung, Übersicht, Fehlersuche und Fehlersammlung relevant sind. Eine Inspektion muss immer gut vorbereitet werden, denn eine Inspektionssitzung dient in der Regel nur zum Sammeln und Bewerten der Befunde. Gelegentlich wird im Rahmen von Inspektionen sogar völlig auf die Durchführung einer Inspektionssitzung verzichtet.

*Typische Phasen einer Inspektion*

In der *Planungsphase* werden u.a. das Ziel der Inspektion, die zu inspizierenden Arbeitsergebnisse, die Art der Durchführung der Inspektion und die Rollen bzw. Teilnehmenden der Inspektion festgelegt.

*Planung*

In der *Übersichtsphase* erläutert der Autor den anderen Mitgliedern des Inspektionsteams die zu überprüfenden Anforderungen, damit innerhalb des Inspektionsteams ein gemeinsames Verständnis über die Anforderungen entsteht.

*Übersicht*

In der *Phase Fehlersuche* durchsuchen die Inspektoren die Anforderung nach möglichen Fehlern. Die Fehlersuche kann von den Inspektoren individuell oder im Team durchgeführt werden. Die individuelle Fehlersuche hat den Vorteil, dass sich jeder Inspektor auf die Anforderung konzentrieren kann. Die Fehlersuche im Team besitzt dagegen den Vorteil, dass die Kommunikation zwischen Inspektoren Synergieeffekte bei der Fehlersuche erzeugen kann. Im Rahmen der Fehlersuche werden identifizierte Fehler dokumentiert.

### **Exkurs: Rollen bei einer Inspektion**

Für die Durchführung einer Inspektion sind die folgenden Rollen mit geeigneten Personen zu besetzen:

- *Organisator*  
Der Organisator plant und überwacht die Inspektionssitzung und den Inspektionsprozess.
- *Moderator*  
Der Moderator leitet die Sitzung und überwacht die Einhaltung des Prozessschemas. Es empfiehlt sich, einen neutralen Moderator zu wählen, da dieser ggf. gegensätzliche Interessen von Autoren und Inspektoren ausgleichen muss.
- *Autor*  
Der Autor erläutert den Inspektoren die von ihm erstellten Anforderungen in der Übersichtsphase und ist später für die Behebung von aufgedeckten Fehlern zuständig.
- *Vorleser*  
Der Vorleser stellt die zu inspizierenden Anforderungen sukzessive vor und führt die Inspektoren durch die zu untersuchende Anforderung. Die Rolle des Vorlesers sollte durch eine neutrale Person besetzt werden, damit die Inspektoren ihre Aufmerksamkeit auf die Anforderung selbst und nicht auf die Interpretation des Autors richten. Häufig übernimmt der Moderator die Aufgaben des Vorlesers.
- *Inspektoren*  
Die Inspektoren sind für das Auffinden von Fehlern verantwortlich und teilen die gefundenen Fehler den anderen Mitgliedern des Inspektionsteams mit.
- *Protokollant*  
Der Protokollant führt ein Protokoll über die Ergebnisse der Inspektion.

In der Phase *Fehlersammlung* werden die identifizierten Fehler zusammengetragen, konsolidiert und dokumentiert. Im Rahmen der

Fehlersammlung und -  
konsolidierung

Konsolidierung können z. B. doppelte Fehler oder unechte Fehler identifiziert werden. Ein unechter Fehler kann z. B. dadurch entstehen, dass ein Inspektor falsche Annahmen über die Anforderung trifft oder einen Sachverhalt falsch interpretiert hat. Zusammen mit der Konsolidierung werden die gefundenen Fehler und Korrekturmaßnahmen in einer Fehlerliste dokumentiert. Inspektionen werden mitunter auch als »Technische Reviews« bezeichnet.

### Exkurs: Assistenztechniken zur Unterstützung des Reviews

Neben den eigentlichen Reviewtechniken existieren Assistenztechniken zur Unterstützung der Prüfung von Anforderungen. Hier haben sich insbesondere zwei Techniken etabliert:

- Perspektivenbasiertes Lesen
- Einsatz von Checklisten

Das *perspektivenbasierte Lesen* ist eine Technik zur Prüfung von Anforderungen, bei der die zu prüfende Anforderung unter verschiedenen Perspektiven überprüft wird [Basili et al. 1996]. Typischerweise wird das perspektivenbasierte Lesen dabei im Zusammenhang mit den anderen Reviewtechniken eingesetzt (z.B. in Inspektionen oder im Walkthrough). Der Fokus auf einer Perspektive beim Lesen eines Dokuments führt zu nachweislich verbesserten Ergebnissen bei der Prüfung von Anforderungen. Mögliche Perspektiven für die Prüfung ergeben sich z. B. aus den unterschiedlichen Adressaten einer Anforderung [Shull et al. 2000]:

- *Perspektive Kunde/Nutzer*  
In der Perspektive eines Kunden bzw. eines Nutzers wird geprüft, ob die Anforderungen die gewünschte Funktionalität und Qualität beschreiben.
- *Perspektive Softwarearchitekt*  
In der Perspektive eines Softwarearchitekten wird geprüft, ob in der Anforderung alle für den Architekturentwurf benötigten Informationen enthalten sind (z.B. ob alle relevanten Performance-Eigenschaften beschrieben sind).
- *Perspektive Tester*  
In der Perspektive eines Testers wird geprüft, ob die Anforderungen die notwendigen Informationen enthalten, um Testfälle aus den Anforderungen abzuleiten.

Eine weitere Strukturierung der Perspektiven kann anhand von Qualitätsaspekten vorgenommen werden (siehe Exkurse zu Qualitätsaspekten):

- *Perspektive Inhalt*

In der Perspektive »Inhalt« überprüft der Prüfer den Inhalt der Anforderungen und fokussiert somit die inhaltliche Qualität der dokumentierten Anforderungen.

- *Perspektive Dokumentation*

In der Perspektive »Dokumentation« überprüft der Prüfer die Einhaltung von Dokumentationsvorschriften für Anforderungen sowie für Anforderungsdokumente.

- *Perspektive Abgestimmtheit*

In der Perspektive »Abgestimmtheit« fokussiert der Prüfer, ob bezüglich der Anforderungen Einigung besteht, d.h., ob die Anforderungen abgestimmt und Konflikte aufgelöst wurden.

Darüber hinaus sind weitere Perspektiven denkbar, die sich z. B. aus dem individuellen Kontext des Entwicklungsprojekts ergeben.

Bei der perspektivenbasierten Überprüfung wird jedem Prüfer (zu einem Zeitpunkt) eine Perspektive zugeordnet, aus deren Blickwinkel der Prüfer die Anforderung liest und überprüft. Für jede definierte Perspektive sollten detaillierte Anweisungen für die Durchführung der Prüfung definiert werden, da ein Prüfer nicht unbedingt mit allen relevanten Details der jeweiligen Perspektive vertraut ist. Es empfiehlt sich, Prüfanweisungen mit Fragestellungen zu versehen, die durch den Inhalt der Anforderung bzw. nach dem Lesen einer Anforderung durch den Prüfer beantwortet werden müssen. Des Weiteren können Prüfanweisungen mit einer Checkliste der wichtigsten Inhalte ergänzt werden, zu denen eine Anforderung bezüglich der betrachteten Perspektive Informationen liefern sollte.

Im Rahmen der Nachbereitung des perspektivenbasierten Lesens werden die Ergebnisse der gewählten Perspektiven ausgewertet und konsolidiert. Die Ergebnisse des perspektivenbasierten Lesens beinhalten zum einen die Beantwortung der definierten Fragen und zum anderen offene Punkte, die den Prüfern während des Lesens aufgefallen sind. Die Konsolidierung kann vergleichbar mit einem Review im Rahmen einer Gruppensitzung stattfinden.

Das perspektivenbasierte Lesen kann sowohl als eigenständige Technik zur Prüfung von Anforderungen eingesetzt werden als auch als Hilfstechnik für andere Prüfungstechniken, z. B. kann die Inspektion oder das Review eines Anforderungsdokuments unter Verwendung des perspektivenbasierten Lesens durchgeführt werden.

Eine *Checkliste* setzt sich aus einer Menge von Fragen oder Aussagen zusammen, die sich auf einen spezifischen Sachverhalt beziehen. Checklisten können immer dann eingesetzt werden, wenn bei einem komplexen Sachverhalt viele Aspekte berücksichtigt werden müssen und hierbei möglichst kein Aspekt vergessen werden soll. Eine Checkliste für die Prüfung von Anforderungen beinhaltet Fragestellungen, die den Prüfern das Aufdecken von Fehlern in den Anforderungen erleichtern [Boehm 1984]. Der Einsatz von Checklisten bei der Prüfung von Anforderungen ist in der Praxis weit verbreitet. Checklisten können in allen vorgestellten Techniken zur Prüfung von Anforderungen eingesetzt werden.

Vor dem Einsatz einer Checkliste müssen die einzelnen Fragen bzw. Aussagen der Checkliste definiert werden. Die Erstellung von Checklisten zur Unterstützung der Prüfung von Anforderungen kann u.a. auf die folgenden Quellen für Fragen bzw. Aussagen zurückgreifen:

- Die drei Qualitätsaspekte (siehe Exkurs »Qualitätsaspekte von Anforderungen« auf S. 144)
- Prinzipien der Prüfung
- Qualitätskriterien für Anforderungsdokumente
- Qualitätskriterien für die jeweiligen Anforderungen
- Erfahrung der Prüfer aus bisherigen Projekten
- Fehlerstatistiken [Chernak 1996]

Eine Checkliste erhebt keinen Anspruch auf Vollständigkeit. Während der Verwendung einer Checkliste sollte daher stets nach Möglichkeiten gesucht werden, die Checkliste für zukünftige Verwendungen zu verbessern. Wurde z. B. eine Fragestellung vergessen, sollte diese ergänzt werden. Mehrdeutig oder unverständlich formulierte Fragestellungen sollten markiert und überarbeitet werden. Veraltete oder nicht mehr gültige Fragestellungen bzw. Aussagen sollten gelöscht werden.

Eine Checkliste kann die Prüfung von Anforderungen auf verschiedene Weise unterstützen. Die Checkliste kann dem Prüfer als ein Leitfaden dienen. Der Prüfer kann diesem Leitfaden während der Prüfung (z.B. bei einem Review) nach eigenem Ermessen folgen.

Die Checkliste kann für den Prüfer eine unbedingt einzuhaltende Liste von Fragestellungen definieren, die der Prüfer für die zu prüfende Anforderung beantworten muss. In diesem Fall dient die Checkliste als ein Strukturierungsmittel. Zum Beispiel kann durch eine Checkliste das Vorgehen

verschiedener Prüfer vereinheitlicht werden und so eine Vergleichbarkeit der Prüfungsergebnisse erzielt werden.

Mischformen für die Anwendung von Checklisten sind ebenfalls möglich. Zum Beispiel kann eine Checkliste für das perspektivenbasierte Lesen zum Teil verbindliche Fragestellungen beinhalten und zum Teil Hinweise geben, die der Prüfer nach eigenem Ermessen anwenden kann.

Die erfolgreiche Anwendung von Checklisten hängt von der Handhabbarkeit bzw. Komplexität der Checkliste ab. Eine große Anzahl von Fragestellungen erschwert die Anwendung der Checkliste, da der Prüfer nicht alle Fragestellungen ständig präsent haben kann und daher gezwungen ist, die Checkliste häufig zu konsultieren. Es empfiehlt sich daher, dass eine Checkliste nicht mehr als eine Seite Text umfassen sollte [Gilb und Graham 1993]. Des Weiteren erschweren zu generisch formulierte Fragestellungen die Anwendung einer Checkliste. Zum Beispiel führt die Frage »Ist die Anforderung angemessen formuliert?« zu verschiedensten Antworten, abhängig davon, was der Prüfer unter einer angemessenen Formulierung versteht. Die Fragestellungen sind daher möglichst zu präzisieren.

#### **4.4.4 Explorationstechniken**

Neben klassischen Reviews können auch explorative Techniken zur Prüfung von Anforderungen eingesetzt werden. Hierzu zählen unter anderem:

- Die Prüfung durch *Prototypen* oder beispielhafte Systementwicklungen
- Kontrollierte Experimente (bspw. *A/B-Tests* oder *Alpha- und Beta-Testing*)
- *Minimum Viable Product (MVP)*  
in der agilen Entwicklung häufig eingesetzte Technik zur Sammlung von Benutzerfeedback zu einem frühen Entwicklungszeitpunkt.

Nachfolgend werden wir auf die Nutzung von Prototypen und den Einsatz kontrollierter Experimente näher eingehen. Die Erzeugung eines MVP ist keinen konkreten Vorgaben unterlegen und beinhaltet eher die gedankliche Idee, bereits möglichst früh ein erlebbares Produkt zu entwickeln, das dann immer weiter fortentwickelt wird. Wie wir sehen werden, ist diese Idee eng mit Prototypen verwandt. Wir sprechen hier im Folgenden von einem evolutionären Prototyp.

##### **4.4.4.1 Prüfung durch Prototypen**

Die Prüfung von Anforderungen durch Prototypen verfolgt den Ansatz, die Anforderungen für den Prüfer erlebbar und damit ausprobierbar zu gestalten. Mit einem Prototyp kann nicht überprüft werden, ob eine Anforderung eindeutig

dokumentiert/formuliert ist. Das unmittelbare Erleben der Auswirkungen von Anforderungen anhand eines Prototyps ist laut [Jones 1998] die effektivste Methode zur Aufdeckung von Fehlern in Anforderungen. Stakeholder können durch das Ausprobieren eines Prototyps ihre eigene Vorstellung mit einer möglichen Umsetzung des geplanten Systems vergleichen und so Fehler bzw. Abweichungen zwischen ihren Vorstellungen und dem prototypisch umgesetzten System identifizieren. Voraussetzung für den Einsatz von Prototypen zur Validierung von Anforderungen ist natürlich die korrekte Umsetzung der Anforderung im Prototyp.

Abhängig von der weiteren Verwendung eines Prototyps wird zwischen Wegwerfprototypen und evolutionären Prototypen unterschieden

*Evolutionäre vs.  
Wegwerfprototypen*

[Sommerville 2007]. Ein Wegwerfprototyp wird nach der Verwendung nicht weiter gepflegt. Ein evolutionärer Prototyp wird mit dem Ziel entwickelt, später weiterentwickelt und verbessert zu werden. Im Gegensatz zum Wegwerfprototyp besitzt die Realisierung bei evolutionären Prototypen einen größeren Stellenwert, daher geht mit der Verwendung von evolutionären Prototypen in der Regel ein wesentlich höherer Realisierungsaufwand einher.



[www.cpre-buch.de/pk4v2](http://www.cpre-buch.de/pk4v2)

### **Exkurs: Durchführung einer Prüfung mittels Prototyping**

Vor der Realisierung des Prototyps müssen diejenigen Anforderungen ausgewählt werden, die mittels des Prototyps überprüft werden sollen. Die Menge der zu prüfenden Anforderungen wird durch die Ressourcen (Zeit, Geld) beschränkt, die zur Prüfung der Anforderungen bzw. zur Realisierung des Prototyps zur Verfügung stehen. Die Auswahl der Anforderungen kann beispielsweise anhand der Kritikalität der Anforderungen erfolgen.

Für die Prüfung der Anforderungen durch einen Prototyp müssen nach Realisierung des Prototyps die folgenden Vorbereitungen getroffen werden:

- *Handbuch/Schulung*

Die Anwender des Prototyps müssen mit den notwendigen Informationen versorgt werden, um den Prototyp verwenden bzw. bedienen zu können. Dies kann z. B. durch ein Handbuch für den Prototyp erfolgen oder durch eine geeignete Schulung.

- *Prüfszenarien*

Für die Durchführung der Prüfung sollten Prüfszenarien vorbereitet werden, die die Prüfer unter Verwendung des Prototyps durchführen

können. Ein Prüfscenario definiert z. B. relevante Datensätze oder Nutzungsabläufe.

- *Checkliste mit Prüfkriterien*

Für die Prüfung der Anforderungen sollte eine Checkliste mit Kriterien erstellt werden, anhand derer die Prüfer den Prototyp (und damit auch die Anforderungen) überprüfen können.

Im Rahmen der Prüfung sollten die Prüfer den Prototyp möglichst ohne äußere Beeinflussung benutzen, d.h., die Prüfer sollten die Prüfscenarien möglichst eigenständig ausführen. Dies stellt sicher, dass die Prüfungsergebnisse ohne Beeinflussung entstehen.

Im Rahmen der Prüfung können und sollen die Prüfer nach der Abarbeitung der Prüfscenarien auch von den vorgegebenen Szenarien abweichen und den Prototyp experimentell ausprobieren. Auf diesem Weg können z. B. bisher unbeachtete Fehlerfälle identifiziert werden. Für die experimentelle Prüfung des Prototyps müssen die Prüfer die Grenzen des Prototyps kennen, d.h. die Menge der Anforderungen, die bei der Realisierung des Prototyps berücksichtigt wurden. Ohne die Kenntnis der realisierten Anforderungen kann ein Prüfer nicht unterscheiden, ob ein identifizierter Fehler auf eine fehlende Anforderung zurückzuführen ist oder ob die Anforderung bewusst nicht im Prototyp umgesetzt wurde.

Bei der Prüfung von Anforderungen durch Prototypen unterscheiden wir zwischen folgenden Formen der Ergebnisdokumentation:

- *Protokoll des Prüfers*

Der Prüfer protokolliert seine Ergebnisse und Erfahrungen aus der Prüfung des Prototyps z. B. anhand der Prüfscenarien sowie der Checkliste, die ihm zur Prüfung zur Verfügung gestellt wurden.

- *Beobachtungsprotokoll*

Der Prüfer kann durch eine zweite Person beobachtet werden. Die zweite Person fertigt ein sogenanntes Beobachtungsprotokoll an. Dieses Protokoll kann zusätzliche wichtige Hinweise auf Anforderungsfehler enthalten, z. B. kann ein dokumentiertes Zögern des Prüfers bei der Verwendung des Prototyps darauf hinweisen, dass der Prototyp unverständlich ist. Unter gewissen Umständen kann es sich auch anbieten, die Prüfung durch eine Videoaufzeichnung zu dokumentieren, damit Prüfsituationen im Nachgang detailliert untersucht werden können. Zum Beispiel kann durch eine Videoaufzeichnung die Umsetzung von Anforderungen an die Ergonomie oder die intuitive Bedienbarkeit eines Systems genauer untersucht werden.

Die Ergebnisse der Prüfung werden im Anschluss an die Prüfung ausgewertet und in Änderungsvorschläge für die Anforderungen übertragen. Bei gravierenden Änderungen an den Anforderungen kann sich eine Überarbeitung des Prototyps inkl. einer erneuten Prüfung anbieten.

#### 4.4.4.2 Prüfung durch kontrollierte Experimente

Eine weitere Möglichkeit, um Anforderungen zu prüfen, ist das Testen dieser im Rahmen kontrollierter Experimente. Diese Praxis wird häufig auch als A/B-Testen bezeichnet (vgl. u.a. [Kohavi und Thomke 2017]). Unter einem A/B-Test wird nicht das Testen des Systems – also das Prüfen gegen die Anforderungen –, sondern ein statistisch belegbares Überprüfen der Anforderungen gegen eine Menge von Stakeholdern bezeichnet.

*A/B-Testen*

Kontrollierte Experimente eignen sich vor allem in einer Ausführung als Online-Experimente. Hier werden in der Regel einfache Fragebögen erstellt, bei denen der Requirements Engineer Aussagen über die Anforderungen tätigt und diese durch die Stakeholder auf ihren Wahrheitsgehalt überprüfen lässt. Solche kontrollierten Experimente eignen sich vor allem, wenn kein direkter Zugriff auf die Stakeholder im Rahmen von Interviews besteht oder wenn beispielsweise eine große potenzielle Nutzergruppe abgefragt werden soll. Das Ergebnis einer solchen Befragung erlaubt es, einzuschätzen, ob eine vom Requirements Engineer vermutete Anforderung tatsächlich den Wünschen und Bedürfnissen der Stakeholder entspricht. Als solches können kontrollierte Experimente sehr gut verwendet werden, um innovative Anforderungen zu prüfen und weiterzuentwickeln. Besonders die Kombination von geschlossenen und offenen Fragen erlaubt eine Weiterentwicklung der Anforderungen auf Basis des Experimentergebnisses. Mit geschlossenen Fragen lässt sich abprüfen, ob die Stakeholder einer gewissen Aussage zustimmen. Mit einer dazu relationierten offenen Frage – also beispielsweise der Frage nach einer Begründung für oder gegen die Anforderung – lassen sich wertvolle Erkenntnisse sammeln, die für die Weiterentwicklung der Anforderung oder auch für die Entwicklung neuer innovativer Anforderungen genutzt werden können.

*Online-Durchführung für innovative Anforderungsprojekte*

Eine weitere verwandte Technik in diesem Bereich ist laut [IREB-Handbuch 2020] das Alpha- und Beta-Testen. Diese Testtechnik wird typischerweise in der Prüfung finaler Software im Abnahmetest eingesetzt. Beim Alpha-Testen wird das finale Produkt von den Entwicklern in einer simulierten Umgebung untersucht, beim Beta-Testen wird das Endprodukt von den Anwendern in realer Umgebung getestet. Dies ist zunächst einmal ziemlich weit von der Anforderungsvalidierung

entfernt. Bietet aber laut [IREB-Handbuch 2020] wertvolle Rückschlüsse auf die Qualität der Anforderungen.

#### 4.4.4.3 Probe-Entwicklung (Konstruktion von Entwicklungsartefakten)

Die Konstruktion von Entwicklungsartefakten zielt darauf ab, die Qualität von Anforderungen z. B. als Grundlage für die Erstellung von Architektur-/Designbeschreibungen, Testartefakten oder des Benutzerhandbuches zu prüfen. Im Rahmen der Überprüfung werden die Entwicklungsaktivitäten zur Konstruktion der jeweiligen Entwicklungsartefakte beispielhaft durchgeführt. So setzt sich z. B. ein Prüfer bei der Erstellung eines Testfalls mit den entsprechenden Anforderungen intensiv auseinander und kann dadurch Fehler (wie z. B. fehlende Testbarkeit der Anforderung) in den Anforderungen identifizieren. Diese Art der Überprüfung ist allerdings sehr ressourcenaufwendig, da nachfolgende Entwicklungsaktivitäten teilweise ausgeführt werden.

*Eignung der Anforderungen für Entwurf, Test und Handbucherstellung*

Ein in der Praxis häufig angewandter Fall ist die Ableitung von Testfällen. Hier beschäftigt sich der Prüfer mit der Frage, ob die Anforderung ausreichend Informationen umfasst und diese präzise genug definiert sind, um die Anforderungen im späteren Verlauf der Entwicklung testen zu können. An dieser Stelle möchten wir kurz auf die Unterscheidung zwischen dem *Prüfen der Anforderungen* und dem *Prüfen gegen die Anforderungen* eingehen. Im Rahmen des Requirements Engineering beschäftigen wir uns in der Regel mit der Prüfung der Anforderungen, d.h. der Überprüfung, ob eine Anforderung korrekt die Wünsche und Bedürfnisse der Stakeholder reflektiert und alle Qualitätskriterien erfüllt. Im Verlauf der Systementwicklung werden dann andere Arbeitsergebnisse (z.B. Entwurfsdokumente oder der Quellcode) gegen diese Anforderungen geprüft, um sicherzustellen, dass das zu entwickelnde System diese erfüllen wird. Es können natürlich auch bei dieser Prüfung Defekte in den Anforderungen identifiziert werden, jedoch ist dies nicht der wünschenswerte Zustand, da wir im Requirements Engineering bestrebt sein müssen, Defekte in den Anforderungen möglichst frühzeitig aufzudecken und nicht erst beispielsweise bei der Ausführung der Systemtests.

*Testfallableitung*

##### **Exkurs: Wechsel der Dokumentationsform**

Ein Wechsel der Dokumentationsform bei der Überprüfung von Anforderungen nutzt die Stärken einer bestimmten Dokumentationsform, um die Schwächen einer anderen Dokumentationsform auszugleichen. Beispielsweise sind die leichte Verständlichkeit und hohe Ausdruckskraft eine Stärke natürlichsprachiger Texte. Eine Schwäche natürlichsprachiger Texte ist jedoch

die teilweise mangelnde Eindeutigkeit und die Schwierigkeit beim Ausdruck komplexer Sachverhalte. Grafische Modelle können komplexe Sachverhalte gut darstellen, sind aber durch die jeweiligen Modellierungskonstrukte in ihrer Ausdruckskraft eingeschränkt.

Die Überführung einer bereits dokumentierten Anforderung in eine andere Dokumentationsform erleichtert das Auffinden von Fehlern. So können z. B. Mehrdeutigkeiten in natürlichsprachigen Anforderungen durch eine Überführung in eine modellbasierte Darstellung leichter entdeckt werden.

#### **Kernfakten 4–6:** *Validierung von Anforderungen*



**[www.cpre-buch.de/pk4w6](http://www.cpre-buch.de/pk4w6)**

# Index

## A

Abgrenzung

des Kontexts *siehe Kontextabgrenzung*

des Systems *siehe Systemabgrenzung*

Abnahmekriterien

Begriffsdefinition 76

Abstimmung 136

Ad-hoc-Priorisierungstechniken 206

Aggregation (UML) 95

Agile Entwicklung 12

Agile Vorgehensweise 19, 22, 110

AHP *siehe Analytical Hierarchy Processing*

Aktivitätsdiagramm 99

Notationselemente 100

Aktivitätsmodell 97

Alpha- und Beta-Testen 159

Analogietechniken 132, 133

Analytische Priorisierungstechniken 206

Änderungsantrag *siehe Change Request*

Änderungsgremium *siehe Change Control Board*

Änderungshäufigkeit 201

Änderungsmanager (Rolle) 203

Änderungswunsch 201

Anforderung

Abstimmung 136

Änderung 201

- Arten von 3
- Attribuierung 189
- Begriffsdefinition 1
- bewusste 126
- Detaillierungsgrad 55
- Anforderung (Fortsetzung)
  - Dokumentationsstrukturen 109, 111
  - fehlerhafte 7
  - Kategorien 54
  - Klassifikation 10
  - lösungsorientierte 11
  - natürlichsprachige 64
  - nicht validierte 35
  - Praktiken für die Erarbeitung von 117
  - Priorisierung 205
  - Qualitätsaspekte 144
  - Qualitätskriterien 114
  - unbewusste 126
  - unterbewusste 126
  - Validieren von 143, 149, 161
  - Version 185
  - Volatilität der 168
  - Wert einer 18
- Anforderungsänderung 37, 38, 201
  - Dokumentation 203
- Anforderungsattribut 189, 191
- Anforderungsbasis 194
- Anforderungsbasislinie 188
- Anforderungsdokument 9, 109
  - Auswahl 110
  - leichtgewichtiges 110
- Anforderungsdokumentation 49
- Anforderungsermittlung 125
- Anforderungskategorisierung
  - nach dem Kano-Modell 125

- Anforderungskonfiguration 186
  - Dimensionen 187
- Anforderungskontext 43
- Anforderungsmanagement *siehe Requirements Management*
- Anforderungsmodellierung
  - Vor- und Nachteile 82
  - Zweck 83
- Anforderungsquelle 46, 117, 123
- Anforderungssicht 193
  - projektive 193
  - selektive 193, 194
  - verdichtende 193, 194
- Anforderungsspezifikation 51
- Anforderungvalidierung 147
- Anforderungsversion 185
- Anwendungsfallmodellierung *siehe Use-Case-Modellierung*
- Apprenticing 131
- Arbeitsergebnis 49
  - Abstraktionsebenen 54
  - Aspekte von 57
  - Begriffsdefinition 49
  - für die Definition von Anforderungsdokumenten 51
  - Lebensdauer 49
  - Merkmale 50
  - modellbasiertes 79
  - natürlichsprachiges 61
  - Planung von 60
  - Qualitätskriterien 114
  - Repräsentationsform 49, 51
  - Verwendungszweck 49
  - vorlagenbasiertes 69
- Arbeitsprodukt 143, 163, 164, 176, 183, 185, 186, 188, 189
- Artefaktbasierte Techniken 131
- Arten von Ermittlungstechniken 126
- Aspekte von Arbeitsergebnissen 57

Assoziation (UML) 93  
Attribuierung 190  
    schablonenbasierte 189  
Attribuierungsschema 190  
Attribute 189  
Audioaufzeichnung 129  
Auftraggeber 7  
Auftraggeber-Auftragnehmer-Beziehung 165  
Auswirkungsanalyse 196, 200  
A/B-Testen 159

## **B**

Basislinie  
    Begriffsdefinition 188  
Befragungstechniken 127, 128  
Benutzeranforderung 10  
Benutzer-Anforderungsspezifikation 109  
Beobachtungstechniken 130  
Beziehungskonflikt 138  
Bionik 133  
Bisoziation 133  
Blockdefinitionsdiagramm (SysML) 86, 91  
Boxes-and-Lines-Diagramm 86  
BPMN (Business Process Model and Notation) 97  
Brainstorming 132  
Brainstorming paradox 133

## **C**

Certified Professional for Requirements Engineering v  
Change Advisory Board 201, 202  
Change Control Board 201, 202  
    Aufgaben 202  
    Besetzung 202  
Change Request 201, 203  
    Änderungsinformationen 203  
    Verwaltungsinformationen 204  
Checklisten 119, 135, 154

Einsatz von 152

mit Prüfkriterien 157

CMMI 10

Consider-all-Facts (CAF) 140

Constraint *siehe Randbedingung*

CPRE *siehe Certified Professional for Requirements Engineering*

## **D**

Datenfluss (Datenflussdiagramm) 87

Datenflussdiagramm 86, 97

    Kontextmodellierung 86

Datenkonflikt 138

Design Thinking 40, 134

Designing for Growth 40, 134

Detaillierungsgrad von Anforderungen 55

Diagramm, Begriffsdefinition 80

Dokumentation der Konfliktlösung 142

Dokumentation des Systemkontexts 48

Dokumentationspraktik 49

Dokumentationsrichtlinie

    allgemeine 59

    für natürlichsprachige Anforderungen 64

Dokumentationsstrukturen für Anforderungen 109, 111

Dokumentieren (Hauptaktivität) 13

Dokumentvorlage 70, 77

    Begriffsdefinition 77

    standardisierte 77, 78

Dokument, Anforderungsquelle 117

Domänenanforderung 10

Domänenmodell

    statisches 92

## **E**

Einflussfaktoren auf Requirements-Engineering-Prozesse 169

Einsatz von Checklisten 152

Entity-Relationship-Diagramm 92

Entscheidungsmatrix 140

- Entwicklungskontext 165
- Entwicklungsprozess 164
- Entwicklungstyp 165
- Entwurfs- und Ideenfindungstechniken 127, 132
- Epic 50
- Ergebnisdokumentation 158
  - Beobachtungsprotokoll 158
  - Prüfprotokoll 158
- Erhebungstechniken 127
- Ermitteln (Hauptaktivität) 13
- Ermittlung
  - durch Wiederverwendung 131
  - expliziten Wissens 127
  - von Innovationen 132
- Ermittlung von Anforderungen 125
  - Entwurfs- und Ideenfindungstechniken 127
  - Erhebungstechniken 127
- Ermittlungstechniken
  - Arten von 126
  - Auswahl 134
- Evolution (Prinzip 7) 37
- Evolutionärer Prototyp 113, 156
- Explizites gemeinsames Verständnis 22
- Explorationstechniken 156
- Explorativer Prototyp 112
- Explorativer Requirements-Engineering-Prozess 172
- Extend-Beziehung 98
- Externe Schnittstellenbeschreibung 50
- F**
- Feature 50
- Feedbackzyklen 56
- Fehlerfortpflanzung 143
- Fehlerstatistiken 154
- Feldbeobachtung 131
- Formularvorlage 70, 76

Formularvorlage/Formularschablone

    Begriffsdefinition 76

Fragebogen 128

Funktionale Anforderung 106

    Begriffsdefinition 3

    Perspektiven 105

## **G**

Gemeinsames Verständnis 23, 56, 117, 166

    explizites 22

    implizites 22

Gemeinsames Verständnis (Prinzip 3) 22

Generalisierung (UML) 95

Generalisierungsbeziehung 98

Geschäftsanforderung 11

Geschäfts-Anforderungsspezifikation 109

Glossar 51, 106, 107, 210

    Grundlagen 107

    Regeln für den Umgang 107

Glossareintrag 107

Goldplating 114

Goldrandlösung 196

Grauzonen der Systemabgrenzung 28, 45

## **H**

Hardwareschnittstelle 45

Haupttätigkeiten 13

Herstellerbefragung 216

Hyperlink 198

## **I**

Identifikator 203

Impact Analysis *siehe Auswirkungsanalyse*

Implizites gemeinsames Verständnis 22

In die Lehre gehen *siehe Apprenticing*

Include-Beziehung 98

Information Hiding 37

Informationsfluss

- Quellen 45
- Senken 45
- Innovation 39, 132
  - disruptive 40
  - inkrementelle 40
- Innovation (Prinzip 8) 39
- Inspektion 150
  - Rollen 151
- Interaktionsmodell 102
- Interesseneigner *siehe Stakeholder*
- Interessenkonflikt 138
- International Requirements Engineering Board v
- Interview 128
- IREB *siehe International Requirements Engineering Board*
- ISO 26262:2018 168
- ISO/IEC 15504 10
- ISO/IEC 25010:2011 135
- ISO/IEC/IEEE 24765:2017 107
- ISO/IEC/IEEE 29148:2018 13, 70, 77, 78, 115
- Iterativer Requirements-Engineering-Prozess 170, 171
- K**
- Kano-Analyse 207
- Kano-Ansatz *siehe Kano-Modell*
- Kano-Modell 125
  - Basisfaktoren 126
  - Begeisterungsfaktoren 126
  - Leistungsfaktoren 126
- Klasse (UML) 92
- Klassendiagramm (UML) 92
- Kollaborationstechniken 129
- Kommunikationsprobleme 8
- Komplexität 167
- Komposition (UML) 95
- Konfiguration, Begriffsdefinition 186
- Konfigurieren eines Requirements-Engineering-Prozesses 182

- Konflikt
  - Identifikation 137
  - Risiken eines 136
  - Techniken zur Auflösung 139
  - unaufgelöster 136
  - vermischte Ursachen 139
- Konfliktanalyse 137
- Konfliktlösung 136, 139
  - bezüglich Anforderungen 142
  - Dokumentation der 142
- Konfliktlösungstechnik 139
  - Abstimmung 140
  - Einigung 139
  - Kompromiss 140
  - Ober-sticht-Unter 140
  - Variantenbildung 140
- Konfliktmanagement 137
- Konflikttypen 137
- Konfliktursachen 139
- Konstruktion von Entwicklungsartefakten 160
- Kontextabgrenzung 42, 44, 117
- Kontextdiagramm 84
- Kontextgrenze 27, 30
  - bestimmen 44, 47
  - Grauzonen 47
- Kontextmodellierung 48, 84
  - Diagrammtypen zur 86
  - mit Datenflussdiagrammen 86
  - mit SysML-Blockdiagrammen 91
  - mit Use-Case-Diagrammen 88
- Kontrollflussgraph 100
- Kosten-Nutzen-Verhältnis 18, 19
- Kostenschätzung 7
- Kosten-Wert-Analyse 207
- Kreativitätstechniken 40, 132

Kritikalität 167  
Kundenspezifischer Requirements-Engineering-Prozess 173  
Kundenwunsch 7

## **L**

Leichtgewichtige Anforderungsdokumente 110  
Linearer Requirements-Engineering-Prozess 171  
Lösungsorientierte Anforderungen 11  
Lösung, Begriff 33

## **M**

Marktorientierter Requirements-Engineering-Prozess 174  
Methodenwissen 14  
Mindmapping 129, 210  
Mock-ups 112  
Modell  
    Begriffsdefinition 79  
    Eigenschaften 80  
Modellbasiertes Arbeitsergebnis 79  
Modellierung von Funktion und Ablauf 97, 101  
Modellierung von Struktur und Daten 92, 95  
Modellierung von Use Cases 88  
Modellierung von Zustand und Verhalten 102  
Modellierungssprachen 81  
Modellierungswerkzeug 214  
MoSCoW-Priorisierung 207  
Multiplizität (UML) 93

## **N**

Nachvollziehbarkeit *siehe Verfolgbarkeit von Anforderungen*  
Nachweisbarkeit 196  
Nativer Prototyp 112  
Natürliche Sprache 61  
Natürlichsprachige Anforderung, Dokumentationsrichtlinien 64  
Natürlichsprachige Arbeitsergebnisse 61  
Nominalisierung 68  
Nutzer 2, 7, 118  
    existierender Systeme 21

## O

Ober-sticht-Unter 139, 140

one fits all 40, 163

## P

Partizipativer Requirements-Engineering-Prozess 176

Passivformulierung 67

Persona 119, 120

    Begriffsdefinition 120

Perspektivenbasiertes Lesen 152

Perspektivenwechsel 133

Pflichtenheft 109, 178, 179

Pilotprojekt 216

Plus-Minus-Interesting 140

Post-RS-Traceability *siehe Vorwärtsverfolgbarkeit*

Praktiken für die Erarbeitung von Anforderungen 117

Präskriptiver Requirements-Engineering-Prozess 172

Pre-RS-Traceability *siehe Rückwärtsverfolgbarkeit*

Priorisierung von Anforderungen

    Techniken 206

    Vorgehen 205

Priorisierungskriterium 205

Priorisierungstechnik 206, 207

Priorität

    einer Anforderung 205

    eines Änderungsantrags 204

Probe-Entwicklung 160

Problem · Anforderung · Lösung (Prinzip 5) 32

Problem-Anforderung-Lösungs-Kette 33

Problemdomäne 163, 165

Problemkontext 56

Problemverständnis 7

Problem, Begriffsdefinition 32

Product Backlog 51, 110

Produktorientierter Requirements-Engineering-Prozess 179

Projektkontext 56

Projektplanung 7  
Protokollant (Rolle) 152  
Prototyp 111, 130

- evolutionärer 113, 156
- explorativer 112
- Handbuch 157
- im Requirements Engineering 111
- nativer 112
- Wegwerfprototyp 156

Prozess (Datenflussdiagramm) 87  
Prozessmodell 97  
Prüfszenarien 157  
Prüfung durch kontrollierte Experimente 159

- Online-Durchführung 159

Prüfung durch Prototypen 156  
Prüfung mittels Prototyping 157  
Prüfungsteam 148

## **Q**

QFD *siehe Quality Function Deployment*  
Qualitätsanforderung 4

- Begriffsdefinition 3

Qualitätsaspekt

- von Anforderungen 144
  - »Abgestimmtheit« 147
  - »Dokumentation« 145
  - »Inhalt« 144

Qualitätskriterien

- für Anforderungen 114
- für Arbeitsergebnisse 114
- für ein Menge von Anforderungen 115
- für einzelne Anforderungen 114

Quality Function Deployment 207

## **R**

Rahmenbedingung *siehe Randbedingung*  
Randbedingung 6

- Begriffsdefinition 6
  - des Entwicklungsprozesses 6
  - des Systems 6
- Requirements Engineer 5, 118, 134
  - Aufgaben 14
  - Auswahl von Praktiken 41
  - Auswahl von Techniken 41
  - Erfahrungen 169
- Requirements Engineer (Fortsetzung)
  - Persönlichkeitsprofil 15
  - Rechte und Pflichten 122
- Requirements Engineering
  - Begriffsdefinition 3
  - grundlegende Prinzipien 17
  - Hauptaktivitäten 13
  - Konfiguration 163
  - mangelhaftes 7
  - Werkzeuge 209
- Requirements Management 183
  - Basislinien 188
  - Konfigurationen 186
  - Praktiken 183
  - spezialisierte Werkzeuge 212
  - Standard-Büroanwendungen 213
  - Versionskontrolle 184
  - Verwaltung des Lebenszyklus 183
- Requirements-Engineering-Prozess 12, 163
  - Arbeitsstruktur 163
  - bei iterativer Entwicklung 170
  - Einflussfaktoren 163, 169
  - explorativer 172
  - iterativer 171
  - Konfiguration 176, 182
  - kundenspezifischer 173
  - linearer 171

- marktorientierter 173, 174
- one fits all 163
- partizipativer 176
- präskriptiver 172
- produktorientierter 179
- verfügbare Zeit und Budget 168
- vertraglich regulierter 178
- vorgegebene Randbedingungen 168
- Requirements-Engineering-Prozesskonfiguration 169
  - bei phasenorientierter Entwicklung 170
  - Zeitfacette 170
  - Zielfacette 173
  - Zweckfacette 172
- Requirements-Engineering-Werkzeuge 209
- Requirements-Management-Werkzeuge 211
  - Kategorien 212
- Review 149
  - Assistenztechniken zur Unterstützung 152
  - technischer 152
- Reviewtechniken 149
- Rückwärtsverfolgbarkeit 197
- S**
- Sachkonflikt 138
- Satzschablone 69, 71
  - Begriffsdefinition 71
  - für User Stories 75
- Schablone 189
- Schnittstelle
  - Hardware *siehe Hardwareschnittstelle*
  - Software *siehe Softwareschnittstelle*
- Sechs-Hut-Denken 133
- Sichten auf Anforderungen 192
- Sichtenbildung 189
  - auf der Anforderungsbasis 193
  - kombinierte 195

- rollenspezifische 193
- selektive 194
- verdichtende 194
- Sicht, Begriffsdefinition 193
- Softwareentwicklungsprozess 12
- Softwareschnittstelle 45
- Soziotechnisches System 33
- SPICE 10
- Sprachliche Defekte 65
- Sprachliche Effekte 65
  - durch Transformation 62
- Sprint Backlog 51, 110
- Stakeholder 1, 202
  - Anforderungsquelle 2, 117
  - Attribute 121
  - Auswirkung fehlender 118
  - Bedeutung von 118
  - Begriffsdefinition 2
  - Beobachtung 130
  - Bezug zum Requirements Engineer 14
  - Einbeziehung 20
  - Fähigkeiten 166
  - Identifikation 20, 119
  - im Requirements Engineering 7
  - in der Abstimmung 136
  - in der Priorisierung 205
- Stakeholder (Fortsetzung)
  - in der Prüfung 147
  - Konflikte 21
  - Rechte und Pflichten 123
  - Rollen 20
  - Umgang mit 120
  - Verfügbarkeit 166
  - Widersprüche 21
- Stakeholder (Prinzip 2) 20

- Stakeholder-Anforderung 10
- Stakeholder-Anforderungsspezifikation 109
- Stakeholder-Befragungen 39
- Stakeholder-Dokumentation 121
- Stakeholder-Liste 120
- Stakeholder-Relationship-Management 121
- Stakeholder-Vereinbarung 122
- Stakeholder-Zufriedenheit 126
- Standard-Bürosoftware 209
- Standardisierte Dokumentvorlagen 78
- Statisches Domänenmodell 92
- Stellungnahme 150
- Story Map 51, 110
- Story-Landschaft *siehe Story Map*
- Strukturkonflikt 138
- System 29, 42
  - als Anforderungsquelle 118
  - komplexes 167
  - Kontextgrenze 44
  - Scope-Grenze 29
  - sicherheitskritisches 167
  - Systemgrenze 44
  - Umfangsgrenze 29
- Systemabgrenzung 44, 117
  - Grauzonen 28, 45
    - Verschiebung der 46, 47
- Systemanforderung 10
- System-Anforderungsspezifikation 109
- Systemarchäologie 131
- Systematische und disziplinierte Arbeit (Prinzip 9) 40
- Systementwicklung
  - Scope-Grenze 29
  - Umfang (Scope) 29
- Systementwicklungsprozess 12
- Systemgrenze 17, 27, 29, 58, 89

- Begriffsdefinition 27
  - bestimmen 27, 44
  - festlegen 45
  - Veränderung 29
- Systemkontext 17, 42, 58
  - Arten von Aspekten 43
  - Begriffsdefinition 26
  - bestimmen 26
  - Dokumentation 48
  - Objekte im 43
- Systemkontext (Prinzip 4) 25
- Systemnutzer 21
- Systemrelease 204
- Szenarien 11, 134
  - deskriptive 134
  - präskriptive 134

## **T**

- Technischer Review 152
- Terminatoren (Datenflussdiagramm) 87
- Testfall 200
- Textverarbeitung 213
- Theorie endlicher Automaten 102
- Traceability *siehe Verfolgbarkeit von Anforderungen*
- Traditionelle Entwicklung 12
- Trainingsprovider (CPRE) v
- Transformation
  - sprachliche Effekte durch 62
- Transformationseffekte 62
- Twin-Peaks-Modell 34

## **U**

- Übereinstimmung *siehe Abstimmung*
- Umfang (Scope) einer Systementwicklung Begriffsdefinition 29
- Umfang-(Scope-)Abgrenzung 31
- UML-Klassendiagramm 93
  - Basisnotationselemente 92

- UML-Zustandsmaschine
  - Basisnotationselemente 102
- Und-Oder-Baum 85
- Unified Modeling Language (UML) 88
- Universalquantor 68
- Unvollständig spezifiziert
  - Bedingung 66
  - Prozesswort 65
- Use Case 50, 89, 97, 129
  - vorlagenbasierte Spezifikation 76
- Use-Case-Beschreibung 76
- Use-Case-Diagramm 88, 90, 97
  - Kontextmodellierung 88
- Use-Case-Diagramm (UML) 86
- Use-Case-Modell 88, 97
- Use-Case-Modellierung 129
- Use-Case-Spezifikation 88, 97
- User Story 11, 50
- User-Interface-Prototyp 111
- V**
- Validieren von Anforderungen 143, 161
  - durch Artefakterstellung 160
  - durch Prototypen 156
  - interne vs. externe Prüfung 148
  - perspektivenbasierte 149
  - Prinzipien 147
  - Techniken zur 149
  - Wiederholung der 149
- Validieren (Hauptaktivität) 13
- Validierer, Auswahl 148
- Validierung 36
  - Begriffsdefinition 143
- Validierung (Prinzip 6) 35
- Variantenbildung 139, 140
- Verfolgbarkeit

- Repräsentation der 198
  - verwendungsbezogene Definition 198
  - zwischen Anforderungen 197
- Verfolgbarkeit von Anforderungen 196
- Verfolgbarkeitsgraph 199
- Verfolgbarkeitsinformation 196
- Verfolgbarkeitskette 200
- Verfolgbarkeitsmatrix 199
- Versionsnummer 185
- Version, Begriffsdefinition 185
- Vertraglich regulierter Requirements-Engineering-Prozess 178
- Vertragsbedingungen 166
- Vertrauen 166
- Verwalten (Hauptaktivität) 13
- Videoaufzeichnung 129
- Vision 51
- Vision-Dokument 110
- Volatilität der Anforderungen 168
- Vorlagenbasierte Spezifikation von Use Cases 76
- Vorlagenbasiertes Arbeitsergebnis 69
- Vorwärtsverfolgbarkeit 197

## **W**

- Walkthrough 149
- Wegwerfprototyp 156
- Werkzeugeinführung 215
- Werkzeugevaluierung 216
- Werkzeugunterstützung 209
- Wert einer Anforderung, Begriffsdefinition 18
- Wertekonflikt 138
- Wertorientierung (Prinzip 1) 18
- Wiederverwendung 78
- Wiederverwendung von Entwicklungsartefakten 197
- Wireframes 112
- Workshop 129

## **Z**

Zertifizierung (CPRE) v  
Zertifizierungsstelle (CPRE) v  
Zieldekomposition 85  
Ziele 11  
Zielmodellierung im Requirements-Engineering 85  
Zurechenbarkeit von Entwicklungsaufwänden 197  
Zustand (UML) 102  
Zustandsmaschinendiagramm 103, 104  
    Notationselemente 104  
Zustandsmodell 102  
Zustandsübergang (UML) 102