

CPRE

Certified Professional for Requirements Engineering

Lehrplan

Requirements Management

Practitioner | Specialist

Nutzungsbedingungen

1. Einzelpersonen und Seminaranbieter dürfen den Lehrplan als Grundlage für Seminare verwenden, sofern die Inhaber der Urheberrechte als Quelle und Besitzer des Urheberrechts anerkannt und benannt werden. Des Weiteren darf der Lehrplan zu Werbezwecken nur mit Einwilligung des IREB e.V. verwendet werden.
2. Jede Einzelperson oder Gruppe von Einzelpersonen darf den Lehrplan als Grundlage für Artikel, Bücher oder andere abgeleitete Veröffentlichungen verwenden, sofern die Autoren und IREB e.V. als Quelle und Besitzer des Urheberrechts genannt werden.

© IREB e.V.

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Die Verwertung ist – soweit sie nicht ausdrücklich durch das Urheberrechtsgesetz (UrhG) gestattet ist – nur mit Zustimmung der Berechtigten zulässig, dies gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmung, Einspeicherung und Verarbeitung in elektronischen Systemen und öffentliche Zugänglichmachung.

Danksagung

Dieser Lehrplan wurde erstellt von (in alphabetischer Reihenfolge) Stan Bühne, Frank Engel, Sven Eselgrimm, Günter Halmans, Andrea Herrmann, Frank Houdek, Patrick Mäder, Alexander Rachmann, Thomas Schölzl, Amin Soesanto, Frank Stöckel und Malik Tayeh.

Allen sei für das ehrenamtliche Engagement gedankt.

Das Urheberrecht © 2015 – 2024 für diesen Lehrplan besitzen die aufgeführten Autoren. Die Rechte sind übertragen auf das IREB International Requirements Engineering Board e. V.

Zweck des Dokuments

Dieser Lehrplan definiert die Lernziele und eine Zusammenfassung der Lerninhalte für die Requirements Management Practitioner und Specialist Zertifizierungen des International Requirements Engineering Board (IREB). Der Lehrplan dient den Schulungsanbietern als Grundlage für die Erstellung ihrer Kursunterlagen. Die Lernenden können sich anhand des Lehrplans auf die Prüfung vorbereiten.

Inhalt des Lehrplans

Das Modul Requirements Management der Fortgeschrittenenstufe spricht Fachleute der Berufsbilder Requirements Engineering, Geschäftsanalyse/Business Analysis, Business Engineering, Organisationsgestaltung u.ä. an, welche ihre Kenntnisse und Fähigkeiten im Bereich der Verwaltung von Anforderungen vertiefen möchten.

Inhaltsabgrenzung

Beim Practitioner und Specialist werden – wie im Foundation Level – Grundlagen des Requirements Engineering vermittelt, die für alle Bereiche, z. B. eingebettete Systeme, sicherheitskritische Systeme, klassische Informationssysteme, gleichermaßen gültig sind. Dies heißt nicht, dass die Eignung von Ansätzen für die einzelnen Bereiche, unter Beachtung

deren Besonderheiten, in einer Schulung nicht behandelt werden können. Es ist jedoch nicht das Ziel, spezifisches Requirements Engineering einer bestimmten Domäne darzustellen.

Die in diesem Lehrplan behandelten Inhalte sind in gleicher Weise in (Entwicklungs-) Projekten wie auch im Produktmanagement, bei der Evolution bestehender Systeme oder dem kontinuierlichen, projektübergreifenden Management von Anforderungen einsetzbar. Aus Gründen der Vereinfachung wird statt dieser umfangreichen Aufzählung oft nur von Projekten gesprochen.

Es wird kein bestimmtes Vorgehens- und damit verbundenes Prozessmodell zugrunde gelegt, das eine Aussage über die Planung, Steuerung und Reihenfolge der Anwendung der erlernten Konzepte in der Praxis macht.

Es geht nicht darum, einen bestimmten Prozess für Requirements Engineering oder gar das gesamte Software bzw. System Engineering besonders hervorzuheben. Im Rahmen von Kapitel 10 wird allerdings auf Besonderheiten des Requirements Managements in agilen Projekten eingegangen.

Requirements Management ohne den Einsatz von Werkzeugen ist in der Praxis nur schwer möglich. Deshalb wird im Rahmen der einzelnen Kapitel auf die Möglichkeiten und Grenzen einer Werkzeug-Unterstützung eingegangen, ohne jedoch ein spezielles Werkzeug in den Vordergrund zu stellen.

Detailierungsgrad

Der Detailierungsgrad dieses Lehrplans erlaubt international konsistentes Lehren und Prüfen. Um dieses Ziel zu erreichen, beinhaltet dieser Lehrplan Folgendes:

- Allgemeine Lernziele
- Inhalte mit einer Beschreibung der Lernziele und
- Referenzen zu weiterführender Literatur (falls notwendig).

Lernziele/Kognitive Stufen des Wissens

Allen Modulen und Lernzielen in diesem Lehrplan ist eine kognitive Stufe zugeordnet. Die Stufen sind wie folgt klassifiziert:

- **K1: Kennen** (beschreiben, aufzählen, charakterisieren, erkennen, benennen, erinnern, ...) – Der Kandidat kann sich an zuvor gelernten Stoff erinnern oder ihn abrufen.
- **K2: Verstehen** (erklären, interpretieren, vervollständigen, zusammenfassen, begründen, klassifizieren, vergleichen, ...) – Der Kandidat kann die Bedeutung anhand von gegebenem Inhalt oder Situationen begreifen/erfassen.
- **K3: Anwenden** (spezifizieren, schreiben, entwerfen, entwickeln, implementieren, ...) – Der Kandidat kann Wissen und Fähigkeiten in gegebenen Situationen anwenden.
- **K4: Analysieren** (untersuchen, schlussfolgern, Argumente liefern, ...) – Der Kandidat kann ein gegebenes Problem analysieren, argumentieren, was getan werden sollte/kann, das Problem in Teile zerlegen, kritisches Denken anwenden, bezüglich Ursachen und Wirkungen argumentieren.

- **K5: Beurteilen** (kritisieren, beurteilen) — Der Kandidat kann eine gut begründete Kritik an einem gegebenen Artefakt äußern; ein fundiertes Urteil in einem gegebenen Fall abgeben.

Beachten Sie, dass ein Lernziel auf der kognitiven Wissensstufe Kn auch Elemente aller darunterliegenden kognitiven Wissensstufen (K1 bis Kn-1) enthält.

Beispiel: Ein Lernziel der Art „Die RE-Technik xyz anwenden“ ist auf der kognitiven Wissensstufe (K3). Die Fähigkeit zur Anwendung setzt aber voraus, dass die Lernenden die RE Technik xyz kennen (K1) und dass sie verstehen, wozu diese Technik dient (K2).



Alle in diesem Lehrplan verwendeten Begriffe und Begriffe, die im Glossar genannt werden, sind zu kennen (K1), auch wenn sie in den Lernzielen nicht explizit genannt sind.

Das Glossar steht auf der IREB Homepage zum Download zur Verfügung:

<https://www.ireb.org/en/downloads/#cpre-glossary-2-0>

Im Lehrplan sowie im dazugehörigen Handbuch wird die Abkürzung RE für Requirements Engineering verwendet.

Lehrplanaufbau

Der Lehrplan besteht aus 11 Hauptkapiteln. Ein Kapitel umfasst eine Lerneinheit (LE). Jeder Haupttitel eines Kapitels beinhaltet die kognitive Stufe des Kapitels, das ist die höchste Stufe der Teilkapitel. Weiterhin werden die Unterrichtszeiten genannt, welche in einem Kurs mindestens für dieses Kapitel aufgewendet werden sollten. Schulungsunternehmen steht es frei, mehr Zeit für die LEs und Übungen zu investieren. Sie sollten jedoch sicherstellen, dass der Zeitaufwand im Verhältnis zu den übrigen LEs beibehalten wird. Die für ein Kapitel wichtigen Begriffe werden zu Beginn jedes Kapitels aufgelistet.

Beispiel:

Kapitel 3 Attributerung und Sichten bei Anforderungen (K5)

Dauer: 2 Stunden

Begriffe: Attribut, Attributierungsschema, Sicht

Das Beispiel zeigt, dass in Kapitel 3 Lernziele der Stufe K5 enthalten sind und zwei Stunden für das Lehren der Inhalte dieses Kapitels vorgesehen sind.

Jedes Kapitel enthält Unterkapitel. In deren Titel findet sich ebenfalls die kognitive Stufe der betroffenen Teilmaterie.

Vor dem eigentlichen Text sind die Lernziele (LZ) gelistet. Die Nummerierung zeigt die Zugehörigkeit zu Unterkapiteln an.

Beispiel: LZ 3.3.1

Dieses Beispiel verdeutlicht, dass das Lernziel LZ 3.3.1 im Unterkapitel 3.3 beschrieben wird.

Die Prüfung

Dieser Lehrplan umfasst Lerneinheiten und Lernziele für die Zertifizierungsprüfungen zum

- CPRE Requirements Management – Practitioner
- CPRE Requirements Management – Specialist

Die Prüfung zum Erlangen des Requirements Management – Practitioner – Zertifikats besteht aus einer **Multiple-Choice-Prüfung**.

Die Prüfung zum Erlangen des Requirements Management – Specialist – Zertifikats besteht aus einer **schriftlichen Ausarbeitung**.

Beide Prüfungen umfassen Prüfungsfragen zu allen Lerneinheiten und allen Lernzielen des Lehrplans.

Jede Prüfungsfrage kann Stoff aus mehreren Kapiteln des Lehrplans sowie mehreren Lernzielen oder auch von Teilen eines Lernziels beinhalten.

Die **Multiple-Choice-Prüfung** für das **Practitioner** Zertifikat

- prüft alle Lernziele des Lehrplans. Bei den Lernzielen der kognitiven Wissensstufen K4 und K5 beschränken sich die Prüfungsfragen jedoch auf Elemente auf den kognitiven Stufen K1 bis K3.
- kann unmittelbar im Anschluss an einen Kurs aber auch unabhängig davon (z. B. remote oder in einem Prüfzentrum) abgelegt werden.

Die **schriftliche Ausarbeitung** für das **Specialist** Zertifikat

- prüft alle Lernziele des Lehrplans auf den für die jeweiligen Lernziele angegebenen kognitiven Wissensstufen.
- folgt der Aufgabenbeschreibung zum Requirements Management – Specialist –, zu finden unter <https://www.ireb.org/de/downloads/tag:advanced-level-written-assignment#top>.
- erfolgt in Eigenregie und wird bei einer lizenzierten Zertifizierungsstelle eingereicht.

Für die **schriftliche Ausarbeitung** für das **Specialist** Zertifikat gelten zudem die folgenden generischen Lernziele:

- LZ G1: Analysieren und illustrieren von Requirements Management-Problemen in einem Kontext, mit dem die Lernenden vertraut sind, oder der einem solchen Kontext ähnlich ist (K4).
- LZ G2: Evaluieren und reflektieren der Anwendung von Requirements Management-Praktiken, Methoden, Prozessen und Werkzeugen in Projekten, an denen die Lernenden beteiligt waren (K5).

Eine Liste der von IREB lizenzierten Zertifizierungsstellen finden Sie auf der Website <https://www.ireb.org>.

Versions-Historie

Datum	Version	Autor	Beschreibung der Änderung
10. Mai 2018	1.1.0		Überarbeitung des Lehrplans
01. Juli 2022	2.0.0	StS	Informationen zum Advanced Level Exam Split hinzugefügt. Wechsel zu neuen kognitiven Stufen des Wissens (5 Stufen)
05. April 2024	2.1.0	StB	<p>Die Definition der kognitiven Wissensstufen wurde an den IREB-Standard angepasst.</p> <p>LZ 3.7, LZ 7.2.3 Schlüsselwort geändert, um der kognitiven Stufe L4 zu entsprechen.</p> <p>LZ 7.2.2 Schlüsselwort geändert, um der kognitiven Stufe L5 zu entsprechen.</p>

Inhaltsverzeichnis

1	Was ist Requirements Management? (K2)	11
1.1	Definition des Requirements Managements (K1)	11
1.2	Aufgaben im Requirements Management (K1)	12
1.3	Ziele und Nutzen des Requirements Managements (K1)	13
1.4	Requirements-Management-Plan (K2)	14
1.5	Relevante Normen (K1)	14
2	Requirements Information Model (L3)	16
2.1	Grundlagen(L2)	16
2.2	Darstellungsformen (K2)	18
2.3	Erstellung eines Requirements Information Model (K3)	20
3	Attributierung und Sichten bei Anforderungen (K5)	23
3.1	Ziele der Attributierung (K1)	23
3.2	Nutzen eines Attributierungsschemas (K2)	24
3.3	Entwurf eines Attributierungsschemas (K3)	25
3.4	Änderungsmanagement von Attributierungsschemata (K5)	27
3.5	Ziele und Arten von Sichten (K2)	28
3.6	Definition von Sichten und Risiken von Sichten (K1)	29
3.7	Optimierung von Attributierung und Sichtenbildung (K4)	30
4	Bewertung und Priorisierung von Anforderungen (K3)	32
4.1	Grundlagen der Bewertung (K1)	32
4.2	Priorisierung von Anforderungen (K2)	33
4.3	Ad-Hoc Priorisierungstechniken (K3)	34
4.3.1	Zwei-Kriterien-Klassifikation (K3)	34

4.3.2	100-Dollar-Technik (K2)	35
4.4	Analytische Priorisierungstechniken (K2)	36
4.5	Kombination von Priorisierungstechniken (K2)	36
5	Versions- und Änderungsmanagement (K3)	38
5.1	Versionierung von Anforderungen (K2)	38
5.1.1	Versionskontrolle für Anforderungen und Anforderungsdokumente (K1).....	38
5.1.2	Anforderungskonfiguration (K2)	39
5.1.3	Anforderungsbasislinie (K1)	39
5.1.4	Branching von Anforderungen (K1)	40
5.2	Änderungsmanagement für Anforderungen (K1)	40
5.2.1	Ursachen, Quellen und Zeitpunkte von Anforderungsänderungen (K2).....	40
5.2.2	Arten von Änderungen an Anforderungen (K2)	41
5.2.3	Analyse und Dokumentation der Stabilität von Anforderungen (K2).....	41
5.3	Änderungsmanagement-Prozess (K3)	42
6	Verfolgbarkeit von Anforderungen (K3)	44
6.1	Gründe für Verfolgbarkeit von Anforderungen (K1)	44
6.1.1	Was versteht man unter Verfolgbarkeit von Anforderungen? (K1).....	44
6.1.2	Warum Verfolgbarkeit von Anforderungen und Artefakten wichtig ist (K1).....	45
6.2	Unterschiedliche Verfolgbarkeits-Betrachtungen (K2)	46
6.3	Beziehungstypen für Verfolgbarkeitsbeziehungen (K1)	47
6.4	Darstellungsformen für Verfolgbarkeitsbeziehungen (K3)	49
6.4.1	Implizite und explizite Dokumentation von Verfolgbarkeit (K3).....	49
6.4.2	Bidirektionale und unidirektionale Verfolgbarkeitsbeziehungen (K2).....	50
6.4.3	Darstellungsformen für Verfolgbarkeitsbeziehungen (K3).....	50
6.5	Erstellung einer Strategie zur projektspezifischen Verfolgbarkeit (K3).....	54
6.6	Spezifische Verfolgbarkeitsmodelle erstellen und verwenden (K3) ...	55
6.6.1	Prozess zur Definition eines spezifischen Verfolgbarkeitsmodells.....	56
6.6.2	Verwendung eines spezifischen Verfolgbarkeitsmodells	56
6.7	Maße zur Bewertung von umgesetzter Verfolgbarkeit (K2)	57

6.8	Herausforderungen bei der Verfolgbarkeit von nicht-textuellen Artefakten (K1).....	58
7	Variantenmanagement für Anforderungen (K5)	60
7.1	Einsatz von Varianten für Anforderungen (K1)	60
7.2	Formen expliziter Dokumentation von Varianten und deren Bewertung (K5).....	62
7.3	Merkmalsmodellierung (K3)	66
8	Berichtswesen im Requirements Management (K3)	70
8.1	Ziele und Nutzen des Berichtswesens im Requirements Management (K1)	70
8.2	Etablierung eines Berichtswesens im Requirements Management (K2) ..	71
8.2.1	Schnittstellen (K2)	71
8.2.2	Inhalte eines Berichts (K1)	71
8.2.3	Tipps für die Entwicklung und Anwendung des Berichtswesens (K2).....	72
8.2.4	Berichtsdefinitionsprozess (K2)	73
8.3	Kennzahlen im Requirements Engineering (K3)	73
8.3.1	Kennzahlen im Requirements Management (K1)	73
8.3.2	Ableiten von Kennzahlen mittels Goal-Question-Metric-Methode (K3).....	74
8.4	Risiken und Probleme im Berichtswesen (K2)	74
9	Management von Requirements Engineering Prozessen (K3) ..	77
9.1	Requirements Engineering als Prozess (K2)	77
9.2	Parameter des Requirements Engineering Prozesses (K3)	79
9.3	Den Requirements Engineering Prozess dokumentieren (K3)	82
9.4	Den Requirements Engineering Prozess überwachen und steuern (K1) ..	83
9.5	Prozessverbesserung im Requirements Engineering Prozess (K3)	83
10	Anforderungsmanagement in agilen Projekten (K2)	86
10.1	Vorwissen (K1).....	86
10.2	Anforderungsmanagement in agilen Projekten (K1)	87

10.3	Abbildung von Requirements Management-Tätigkeiten auf Scrum-Tätigkeiten (K2).....	88
11	Werkzeugeinsatz im Anforderungsmanagement (K1)	90
11.1	Rolle von Werkzeugen im Anforderungsmanagement (K1)	90
11.2	Prinzipielle Vorgehensweise bei der Werkzeugauswahl (K1)	91
11.3	Datenaustausch zwischen Requirements Management Werkzeugen (K1) ...	91
12	Literaturverzeichnis	93

1 Was ist Requirements Management? (K2)

Dauer: 1,25 Stunden

Begriffe: Requirements Engineering, Requirements Management, Requirements Manager, Requirements Management Plan

Lernziele

- LZ 1.1 Begriffsabgrenzung Requirements Engineering und Requirements Management kennen (K1)
- LZ 1.2 Aufgaben im Requirements Management kennen (K1)
- LZ 1.3 Ziele und Nutzen des Requirements Managements kennen (K2)
- LZ 1.4 Die Notwendigkeit eines Requirements Management Plans kennen (K2)
- LZ 1.5 Anforderung einschlägiger Normen an das Requirements Management kennen (K1)

1.1 Definition des Requirements Managements (K1)

Requirements Management (RM) kann unter zwei Gesichtspunkten betrachtet werden:

1. Das Managen (Verwalten) von Anforderungen und von Anforderungs-Artefakten im Entwicklungsprozess
2. Das Managen der Aktivitäten des Requirements Engineering (d.h. RM als Prozessmanagement)[POHL2010], Kapitel 30.1

RM kann im Kontext eines Entwicklungsprojekts, bei der Evolution eines bestehenden Systems, im Software-Produktmanagement oder im kontinuierlichen, projektübergreifenden Management von Anforderungen genutzt werden.

Das IREB Glossar [IREB Glossary] definiert RM als einen Prozess, um existierende Anforderungen und mit ihnen verbundene Artefakte zu verwalten. Dies beinhaltet die Speicherung, die Veränderungen und die Verfolgung der Anforderungen und der weiteren Artefakte. Darunter fällt unter anderem, Anforderungen zu strukturieren, aufzubereiten sowie konsistent zu ändern und umzusetzen [RuSo2009].

Die Begriffsabgrenzung zwischen Requirements Engineering (RE) und Requirements Management (RM) erfolgt in der Literatur häufig unterschiedlich. Je nach Definition

- ist Requirements Engineering Teil des Requirements Managements (z.B. in [SCHI2001] oder aber
- ist Requirements Management Teil des Requirements Engineerings (z.B. in [IREB2020] oder aber
- sind Requirements Engineering und Requirements Management als nebeneinanderstehende Aspekte definiert (z.B. [CMMI 2011]).

IREB definiert Requirements Management **als einen Teil** des Requirements Engineerings.

Im Rahmen dieses Lehrplans stellt sich die Abgrenzung der Begriffe demnach wie folgt dar:

- Unter dem Begriff **Requirements Engineering** werden die Aufgabenbereiche der Ermittlung, Dokumentation, Prüfung /Abstimmung, und der Verwaltung von Anforderungen zusammengefasst.
- Unter dem Begriff **Requirements Management** hingegen versteht man den Aufgabenteil der Verwaltung von Anforderungen innerhalb des Requirements Engineerings.

1.2 Aufgaben im Requirements Management (K1)

Das Requirements Management ist dafür verantwortlich, Informationen über Anforderungen so bereitzustellen, dass andere Rollen im Projekt effizient damit arbeiten können. Zudem sind Regeln und Methoden bereitzustellen, die die Bereitstellung von Informationen ermöglichen [RuSo2009].

Werden Anforderungen gesammelt, so gelten drei grundsätzliche Annahmen, auf Basis derer sich alle Aufgaben des Requirements Managements ableiten lassen und mit denen sich die nachfolgenden Aufgaben und Methoden begründen lassen (vgl. auch [RuSo2009]):

- Anforderungen müssen von mehreren Personen genutzt werden,
- Anforderungen ändern sich,
- Anforderungen sollen wiederverwendet werden.

Die nachfolgenden Aufgaben und Begrifflichkeiten sind Bestandteil des Requirements Managements [PoRu2011]:

- **Attributierung:** Attribute ermöglichen es, Anforderungen und deren Metainformationen strukturierter zu beschreiben, zu gruppieren oder vergleichbar mit anderen Anforderungen zu machen. Attributierte Anforderungen sind die Grundlage für die Sichtenbildung auf Anforderungen.
- **Bewertung und Priorisierung:** Prioritäten ermöglichen es, mithilfe von Bewertungs- und Priorisierungskriterien zwischen wichtigen und weniger wichtigen, aufwändigen und weniger aufwändigen Anforderungen zu unterscheiden. Diese Unterscheidung wiederum dient als Grundlage von Projektmanagement- und Release-Planungs-Entscheidungen.
- **Verfolgbarkeit:** Verfolgbarkeit ermöglicht es, eine Anforderung über den gesamten Lebenszyklus des Systems hinweg nachvollziehen zu können. Auf Basis dieser Information können beispielsweise bei der Änderung einer Anforderung die davon abhängigen Anforderungen und weitere Entwicklungsartefakte identifiziert werden.
- **Versionierung:** Versionierung ermöglicht es, die Änderung von Anforderungen innerhalb ihres Lebenszyklus nachvollziehen zu können.
- **Berichtswesen:** Das Berichtswesen ist das Sammeln, Auswerten und Darstellen von Informationen über Anforderungen oder den Requirements Engineering Prozess (kurz: RE-Prozess). Die in Berichten enthaltenen Informationen dienen neben dem reinen Informieren auch als Grundlage für Projektentscheidungen sowie zur Steuerung des RE-Prozesses.

- **Management der Prozesse:** Die Verwaltung der Arbeitsvorgänge im Requirements Engineering ermöglicht es, die Prozesse des Teams effizient zu gestalten.

Die Aufgaben des Requirements Managements werden durch den Requirements Engineer oder den Requirements Manager geplant und durchgeführt.

1.3 Ziele und Nutzen des Requirements Managements (K1)

Das Ziel des Requirements Managements ist es, Anforderungen und andere Artefakte mit Bezug zu Anforderungen so zu verwalten, dass die Anforderungen mit vertretbarem Aufwand systematisch durchsucht, gruppiert, bewertet, geändert und verfolgt werden können. Es liefert unter anderem Antworten auf die nachfolgenden Fragestellungen:

- Welche Anforderungen sind Bestandteil des Systems? (Attributierung)
- Welche Anforderungen stammen aus welcher Quelle? (Attributierung und Verfolgbarkeit)
- Welche Anforderungen sind dringend und wichtig? (Bewertung und Priorisierung)
- Welche Anforderung erzeugt zu hohe Kosten bei zu geringem Nutzen? (Bewertung und Priorisierung)
- Welche (Sub-)Systemanforderung gehören zu welchen Stakeholder-Anforderungen? (Verfolgbarkeit)
- Welche Anforderung ist Teil meines Systems / Produkts? (Verfolgbarkeit)
- Welche Version der Anforderung wurde in meinem System umgesetzt? (Versionierung)
- Wer hat die Anforderung zuletzt verändert? (Versionierung)
- Mit welchen Kennzahlen lässt sich mein Requirements Engineering steuern? (Berichtswesen)
- Wie effizient funktioniert mein Requirements Engineering? (Management des Prozesses)

Die Bedeutung des Requirements Managements innerhalb des Entwicklungsprozesses steht im engen Zusammenhang mit den Randbedingungen des Projektes [RuSo2009].

Requirements Management wird umso wichtiger...

- ... je größer die Zahl der Anforderungen ist,
- ... je länger die geschätzte Lebensdauer des Produktes ist,
- ... je mehr Änderungen zu erwarten sind,
- ... je größer die Anzahl der Beteiligten am RE-Prozess ist,
- ... je schlechter die Stakeholder zu erreichen oder einzubeziehen sind,
- ... je höher die Qualitätsansprüche an das System sind,
- ... je mehr Wiederverwendung betrieben werden soll,
- ... je komplexer der Entwicklungsprozess ist,
- ... je inhomogener die Stakeholder-Meinungen sind,
- ... je mehr Releases entwickelt werden,
- ... je wichtiger die Nutzung von Normen ist.

Ein gutes Requirements Management ...[RuSo2009]

- ... erhöht die Qualität von Anforderungen, Produkten und Prozessen,
- ... reduziert Projektkosten und Projektlaufzeit,
- ... vereinfacht die Überwachung komplexer Projekte während aller Phasen,
- ... verbessert die Kommunikation innerhalb der und zwischen den Teams,
- ... erhöht die Kundenzufriedenheit,
- ... reduziert das Projektrisiko.

1.4 Requirements-Management-Plan (K2)

Ähnlich wie ein Projektmanagement-Handbuch beschreibt der Requirements Management Plan (RMP) die Festlegungen zur Umsetzung des Requirements Engineering. Dies umfasst die Planung, welche Arten von Anforderungen dokumentiert werden sollen, wie Anforderungen verwaltet werden und wer welche Verantwortung im RE-Prozess hat.

Der Requirements Management Plan umfasst insgesamt:

- Das Requirements Information Modell (RIM), in dem u.a. die Anforderungslandschaft, d.h. die zu verwaltenden Anforderungs-Arten und deren Detaillierungsebenen, beschrieben sind (siehe 2),
- Attribute und Sichten auf die Anforderungen (siehe 3),
- Priorisierungskriterien und -methoden (siehe 4),
- Vorgaben zur Versionsverwaltung von Anforderungen und Anforderungs-Artefakten sowie den Änderungsprozess (siehe 5),
- Vorgaben zur Verwaltung der Verfolgbarkeit von Anforderungen (siehe 6),
- Vorgaben zur Beschreibung von Anforderungsvarianten (siehe 7),
- Vorgaben zum Berichtswesen über Anforderungen (siehe 8),
- RE-Prozess mit Aktivitäten, Rollen und Verantwortlichkeiten (siehe 9)
- Vorgaben zu den zu nutzenden Werkzeugen (siehe 11).

In der Praxis ist der Requirements Management Plan oft kein eigenständiges Dokument, sondern Bestandteil des Projektmanagementhandbuchs, des Konfigurationsmanagementplans oder anderer Vorgabedokumente zum Entwicklungsprozess.

1.5 Relevante Normen (K1)

Das Thema Requirements Management wird in einer Vielzahl von Normen und Vorgaben in unterschiedlicher Weise aufgegriffen. Nachfolgend sind einige prominente Vertreter aufgeführt.

- Das CMMI (Version 1.3) [CMMI 2011] betrachtet unter anderem die Prozesse „Anforderungsentwicklung (Requirements Development)“ und „Anforderungsmanagement (Requirements Management)“, wobei die zugeordneten Ziele sich teilweise deutlich von den Festlegungen im IREB unterscheiden.

- Die ISO 9000 / ISO 9001 [ISO 9000] ist eine Norm für Qualitätsmanagement in Unternehmen. Die ISO 9001 legt Mindestanforderungen an ein Qualitätsmanagementsystem fest und beschreibt beispielsweise Anforderungen an die Produktrealisierung sowie Messung und Verbesserung und adressiert somit Themen wie Identifizierbarkeit oder Verfolgbarkeit.
- In der ISO/IEC 12207:2008 bzw. 15288 („Systems and software engineering – Software life cycle processes“ bzw. „Systems engineering – Systems life cycle processes“) werden die wichtigsten Prozesse zur Entwicklung von Systemen und Software definiert und damit auch Aufgaben aus dem Bereich Requirements Engineering und Requirements Management beschrieben.
- Die IEC 61508 („Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme“) beschäftigt sich mit der Definition von Anforderungen an die funktionale Sicherheit von Systemen. Insbesondere dem Thema Verfolgbarkeit wird hier eine besondere Bedeutung beigemessen.
- ISO/IEC/IEEE 29148:2011 („Systems and software engineering – Life cycle processes – Requirements engineering“) beschreibt Prozesse speziell für das Requirements Engineering.
- SOX (Sarbanes–Oxley Act) **Fehler! Verweisquelle konnte nicht gefunden werden.** [NEMN2006] ist ein US–Bundesgesetz als Reaktion auf Bilanzskandale, welches die Verlässlichkeit der Berichterstattung von Unternehmen, die am öffentlichen Kapitalmarkt in den USA notiert sind, verbessern soll. Im Kern fordert der Sarbanes–Oxley Act zu wissen wer, zu welchem Zeitpunkt, welche Änderung durchführen ließ und betrifft somit auch Kernaufgaben des Requirements Managements.

Aufgrund äußerer Randbedingungen (z.B. explizite Kundenanforderungen oder regulative Vorgaben) sind oft einzelne oder mehrere dieser Normen oder Vorgaben umzusetzen und damit ist auch die Umsetzung eines Requirements Management verpflichtend.

Die Normen nutzen teilweise unterschiedliche Begriffsdefinitionen und sind somit nur beschränkt kompatibel zueinander.

2 Requirements Information Model L3]

Dauer: 2 Stunden

Begriffe: Anforderungslandschaft, Anforderungsart, Darstellungsformen, Requirements Information Model (RIM)

Lernziele

- LZ 2.1.1 Grundlegende Klassifikationen von Anforderungen kennen (K1)
- LZ 2.1.2 Twin-Peaks-Modell kennen (K1)
- LZ 2.2 Typische Darstellungsformen für Anforderungen kennen (K1)
- LZ 2.3 Requirements Information Model erstellen können (K3)

2.1 Grundlagen(L2)

Anforderungen können nach verschiedenen Gesichtspunkten klassifiziert werden:

- Anforderungsart
- Lösungsunabhängigkeit
- Detaillierungsebene oder Abstraktionsebene der Anforderung

Die Klassifizierung nach Anforderungsart ist aus dem IREB Foundation Level bekannt [IREB2020], Kapitel 1, [POHL2010]:

- Funktionale Anforderungen
- Qualitätsanforderungen
- Randbedingungen

Die Spezifika der verschiedenen Anforderungsarten sowie deren weitergehende Kategorisierung werden ausführlich in [POHL2010], [PoRu2011] behandelt.

In konkreten Projekten werden diese Anforderungsarten oft verfeinert oder es werden auch andere Namen für diese Anforderungsarten verwendet. Detaillierte Klassifikationsschemata für Anforderungen sind z.B. in [YOUN2004], [RuSo2009], [WiBe2013], [RoRo2014] zu finden.

Dazu orthogonal können Anforderungen nach ihrem Grad der Lösungsunabhängigkeit klassifiziert werden [POHL2010]:

Hohe Lösungsunabhängigkeit, z.B. Ziele: Ziele beschreiben die Intention des Systems, ohne dabei auf die Umsetzung einzugehen und beschreiben damit die lösungsunabhängigste Form einer Anforderung.

Mittlere Lösungsunabhängigkeit, z.B. Szenarien: Szenarien beschreiben auf exemplarische Art und Weise mögliche Interaktionsfolgen zur Erfüllung eines oder mehrerer Ziele. Diese sind aus Benutzersicht beschrieben.

Geringe Lösungsunabhängigkeit, z.B. Lösungsorientierte Anforderungen: Lösungsorientierte Anforderungen beschreiben die zur Erfüllung der Ziele und zur Umsetzung der Szenarien notwendigen Daten, Funktionen, Systemverhalten, Zustände sowie die erforderliche Qualität des Systems.

In der dokumentierten Form nennt [POHL2010] diese drei Klassen auch Anforderungs-Artefakt-Typen.

Eine dritte Klassifikation, nach der Anforderungen unterschieden werden können, ist die Detaillierungsebene, auch Abstraktionsebene genannt. Oft werden Anforderungen auf mehreren Detaillierungsebenen beschrieben: In einem geringen Detaillierungsgrad mit wenig Details, um einen Überblick zu bilden, und zusätzlich mit einem höheren Detaillierungsgrad mit mehr Details, um die Anforderungen exakt und vollständig zu spezifizieren.

Twin-Peaks-Modell

Anforderungen bilden die Grundlage für den Systementwurf. Jedoch hat die Praxis gezeigt, dass ein Wasserfallvorgehen, bei dem sequenziell nach der Spezifizierung der Anforderungen eines Systems dessen Entwurf erfolgt, häufig nicht der Realität entspricht [POHL2010], [CHSM2013]. Stattdessen handelt es sich vielmehr um einen iterativen Prozess, bei dem Anforderungen die Architektur eines Systems prägen und umgekehrt. Abbildung 1 verdeutlicht diesen Zusammenhang im sogenannten Twin-Peaks-Modell [NUSE2001]. Die vertikale Achse stellt den Detaillierungsgrad der Systembeschreibung dar, während die horizontale Achse die zunehmende Lösungsabhängigkeit darstellt. Die Abbildung zeigt, dass iterativ eine detaillierter werdende Anforderungsbeschreibung und parallel eine detaillierter werdende Architektur des Systems entstehen.

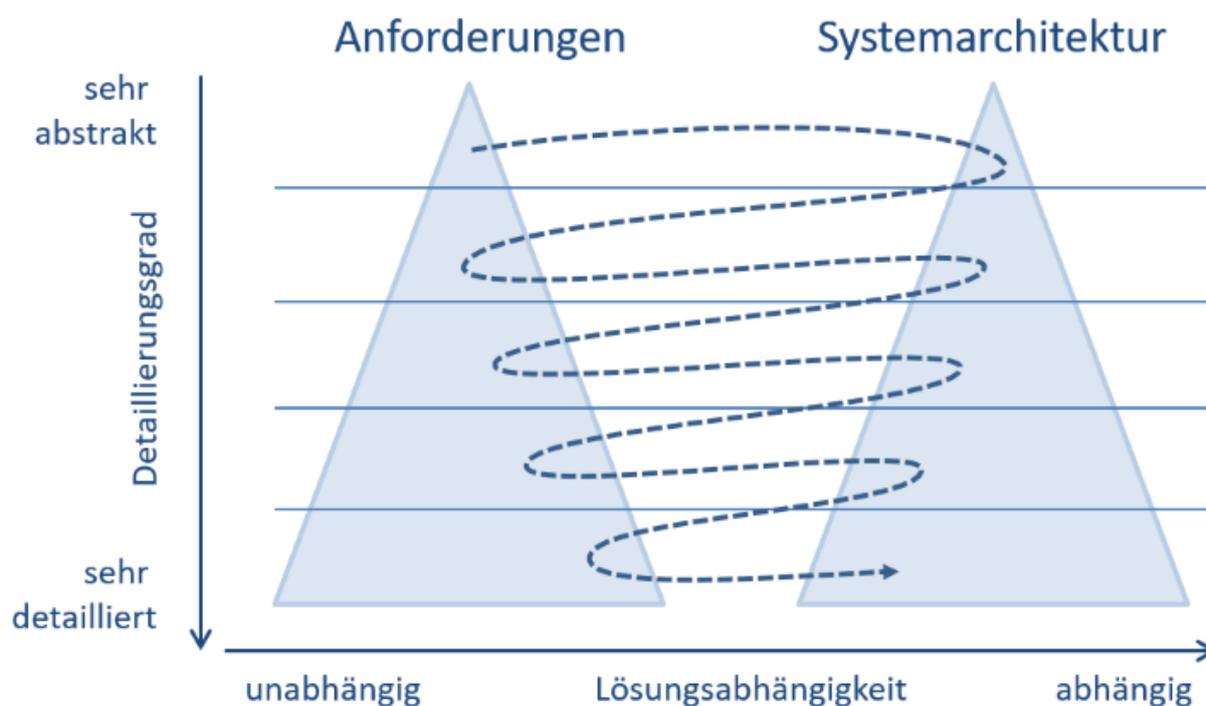


Abbildung 1: Twin-Peaks-Modell

Die Festlegung der benötigten Abstraktionsebenen, d.h. der Ebenen, auf denen die unterschiedlichen Detaillierungsgrade der Anforderungen beschrieben werden, ist projektspezifisch und kann nicht allgemeingültig beschrieben werden. Dennoch wird grundsätzlich empfohlen, Anforderungen so weit zu detaillieren, bis:

- Ein gemeinsames Verständnis aller Stakeholder in Bezug auf die Anforderung erreicht wurde und jedem klar ist, was exakt gefordert wird.

- Die verbleibenden Freiheitsgrade für die Lösung so gering sind, dass eine weitere Präzisierung mehr Kosten als Nutzen verursachen würde (d.h. das Restrisiko, das sich aufgrund verbleibender Freiheitsgrade ergibt, ist akzeptabel).
- Die Anforderungen soweit spezifiziert sind, dass diese gegen die spätere Lösung eindeutig überprüfbar (testbar) sind.

Häufig verwendete Abstraktionsebenen (Detaillierungsebenen) sind beispielsweise „Gesamtsystem“, „Teilsystem“, „Technische Komponente“ oder „Kunden- und Stakeholder-Anforderungen“, „System-, Architektur-, Komponenten-Design Anforderungen“, „Implementierungs-Anforderungen“ oder auch die Aufteilung in „Lastenheft“ und „Pflichtenheft“.

Entscheidend für das Requirements Management sind zwei Aspekte dieses Vorgehens:

- Um verschiedene Sichten auf die Anforderungen zu unterstützen, werden Anforderungen in unterschiedlichen Abstraktionsgraden auf expliziten Detaillierungsebenen spezifiziert und vorgehalten.
- Um diesen inkrementellen und iterativen Prozess später nachvollziehen zu können und sicherzustellen, dass Vollständigkeit und Konsistenz über verschiedene Detaillierungsebenen gewahrt bleiben, ist es unabdingbar, Anforderungen auf verschiedenen Detaillierungsebenen, aber auch Anforderungen und deren realisierende Elemente in der Architektur miteinander zu verknüpfen. Dies wird mit Verfolgbarkeitsbeziehungen (vgl. 6) realisiert.

2.2 Darstellungsformen (K2)

Grundsätzlich ergibt sich mit den drei Anforderungsarten, den drei Arten der Lösungsunabhängigkeit sowie den gewählten Detaillierungsebenen eine Vielzahl potentieller „Kreuzungspunkte“ (siehe Abbildung 2), im Folgenden Klassen von Anforderungen genannt.

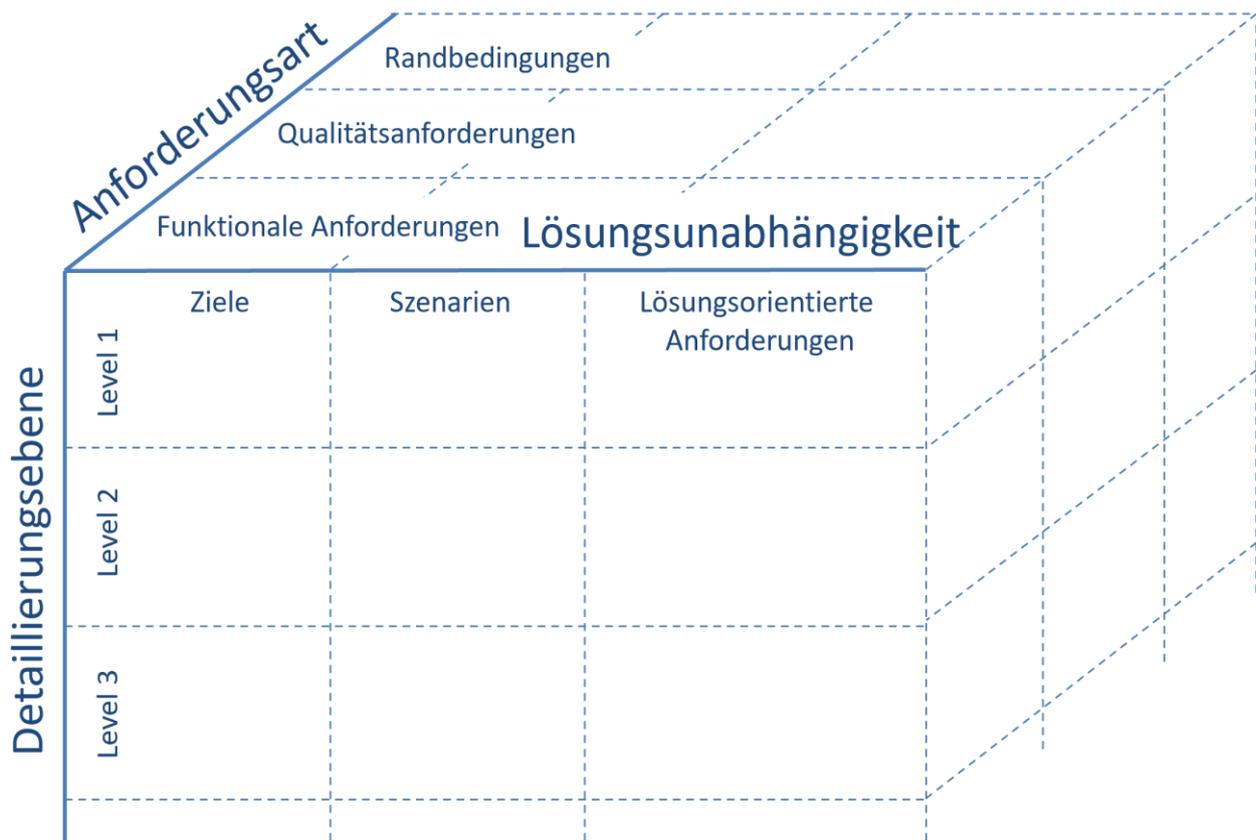


Abbildung 2: Vielzahl potentieller „Kreuzungspunkte“ (Klassen von Anforderungen)

Es muss nun festgelegt werden, (1) welche der „Kreuzungspunkte“ überhaupt berücksichtigt werden und (2) wie diese dokumentiert werden sollen. Eine vollständige Berücksichtigung aller Kreuzungspunkte ist normalerweise wirtschaftlich nicht möglich und oft auch inhaltlich nicht sinnvoll.

Für die Dokumentation von Anforderungen gibt es unterschiedliche Darstellungsformen, Formalitätsgrade und Notationen / Sprachen, z.B.:

- Natürliche Sprache (z.B. Beschreibung in reiner Prosa)
- Strukturierter Text (z.B. Schablonen, Templates / Vorlagen, Satzmuster)
- Modellbasierte, graphische Notationsformen (z.B. UML, Feature-Baum)
- Formale Beschreibungen (z.B. mathematische Funktionen)

Für jeden benötigten „Kreuzungspunkt“ muss festgelegt werden, welche Darstellungsform hierfür benötigt wird. Dabei kann es auch sein, dass für einen „Kreuzungspunkt“ verschiedene Ausprägungen benötigt werden, z.B. in Abhängigkeit von Stakeholder-Gruppen oder Systemteilen. Beispielsweise kann es sinnvoll sein, auf der Detaillierungsebene „Teilsystem“ die Anforderungsart „Funktionale Anforderung“ in der Lösungsunabhängigkeit „Lösungsorientierte Anforderung“ als textuelle Anforderungen (zur Dokumentation von allgemeinen Kundenanforderungen) oder als Message-Sequence-Charts (zur Dokumentation von Schnittstellenanforderungen) zu verwenden.

Zudem muss bereits zu Beginn festgelegt werden, in welcher Landessprache textuelle und modellbasierte Anforderungen zu dokumentieren sind, um später unnötigen Mehraufwand durch Übersetzungsprozesse zu vermeiden.

2.3 Erstellung eines Requirements Information Model (K3)

In realen Projekten wird man immer nur einen Ausschnitt der potentiellen „Kreuzungspunkte“ betrachten, die sich durch die rechnerische Kombination von Anforderungsarten, Lösungsunabhängigkeit und Detaillierungsebenen ergeben.

Die Gesamtheit der in einem Projekt genutzten „Kreuzungspunkte“ nennt man Anforderungslandschaft. Eine Anforderungslandschaft ist eine Festlegung der (a) zu verwendenden Klassifizierung in Bezug auf die Anforderungsarten, (b) zu verwendenden Klassifizierung in Bezug auf die Lösungsunabhängigkeit der Anforderung, (c) notwendigen Detaillierungsebenen je Anforderungs–Artefakt–Typ und (d) zu verwendenden Darstellungsformen je Anforderungs–Artefakt–Typ.

Bisweilen ergeben sich aus zu beachtenden Normen und Vorgaben bereits erste Hinweise, welche „Kreuzungspunkte“ in jedem Fall zu berücksichtigen sind. Beispielsweise fordert die [ISO 26262], dass ausgehend von Sicherheitszielen (d.h. eine besondere Ausprägung von Anforderungen in Form von Zielen), fachliche Sicherheitsanforderungen und daraus folgend technische Sicherheitsanforderungen abgeleitet werden.

[RuSo2009] nennt beispielhaft fünf Detaillierungsebenen, die dort Spezifikationslevel genannt werden:

- **Spezifikationslevel 0:** beschreibt grob das Gesamtvorhaben mit seinen Zielen
- **Spezifikationslevel 1:** beschreibt die Anwendungsfälle und Geschäftsprozesse der Fachbereiche, die der Zielerfüllung dienen (fachliche Spezifikation)
- **Spezifikationslevel 2:** detailliert die Geschäftsprozesse und Geschäftsanforderungen der Fachbereiche auf Spezifikationslevel 1 (fachliche Spezifikation)
- **Spezifikationslevel 3:** beschreibt detaillierte Anwenderanforderungen mit Aufteilung in Teilsysteme und Beschreibung der Schnittstellen (fachliche Spezifikation)
- **Spezifikationslevel 4:** beschreibt die technische Spezifikation mit Trennung in Hardware, Software und sonstige Bestandteile (technische Spezifikation)

Der Spezifikationslevel 0 umfasst also alle Anforderungsarten in der Lösungsunabhängigkeit „Ziele“ mit dem geringsten Detaillierungsgrad.

Aus diesem Beispiel wird ebenfalls deutlich, dass nicht auf allen Detaillierungsebenen alle Ausprägungen von Anforderungen bezüglich Anforderungsart und Lösungsunabhängigkeit dokumentiert werden.

Die getroffenen Festlegungen zur Anforderungslandschaft (d.h. die benötigten „Kreuzungspunkte“ und die gewählten Darstellungsform(en)) sollten als Requirements Information Model (RIM) dokumentiert werden. Dazu bietet sich beispielsweise die Erstellung eines Entity–Relationship Diagramms oder eines Klassendiagramms an. Ein Beispiel hierfür findet sich in Abbildung 3. Das Beispiel lehnt sich dabei an die Spezifikationslevel aus [RuSo2009] an, ohne diese allerdings vollständig umzusetzen.

Neben den bereits getroffenen Festlegungen zu Anforderungsart, Lösungsunabhängigkeit, Detaillierungsebene und Darstellungsform sollte das Requirements Information Model um weitere Aspekte ergänzt werden:

- Welche Attribute werden wo eingesetzt? (see 3)
- Welche Sichtenbildung wird unterstützt? (see 3)
- Welche Bewertungskriterien sind für Anforderungen vorgesehen? (see 4)
- Welchen Rollen obliegt die Pflege und Änderung welcher Informationen? (see 5)
- Welche Verfolgbarkeitsbeziehungen zwischen Anforderungen sowie vor- und nachgelagerten Artefakten werden dokumentiert? (see 6)
- Wie werden Varianten von Anforderungen dokumentiert? (see 7)

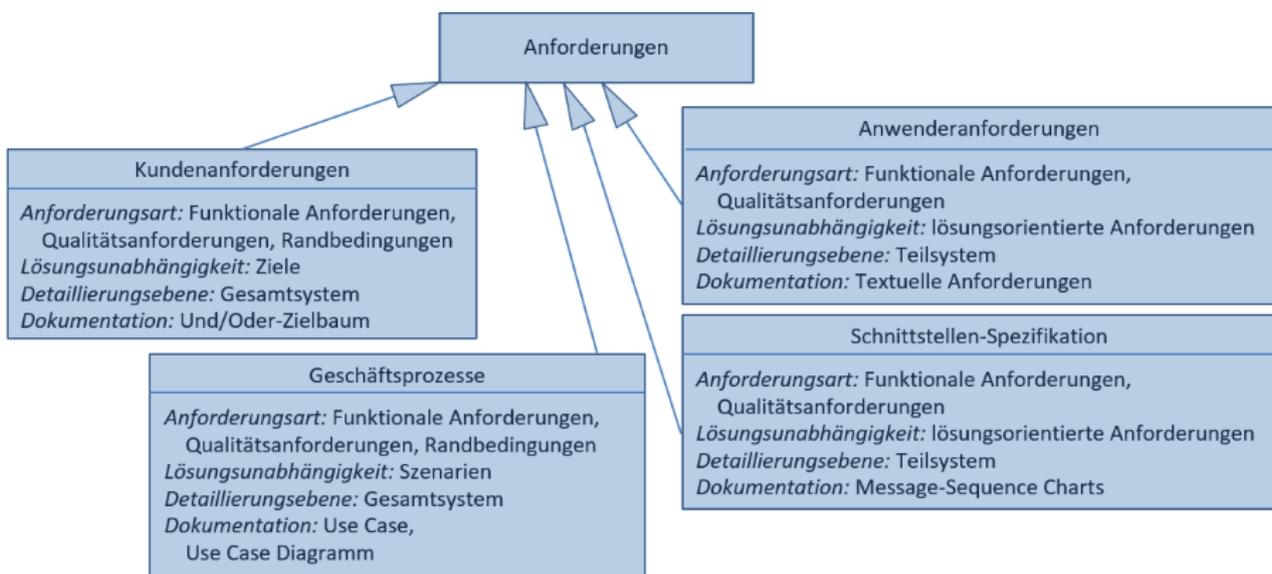


Abbildung 3: Beispiel eines spezifischen Requirements Information Model (RIM)

Das RIM stellt allen Stakeholdern eines Projektes eine zentrale Sicht auf die verwendeten Anforderungsartefakte und deren Verwendung im Requirements Management zur Verfügung und ist damit ein wesentlicher Teil des Requirements Management Plans (RMP).

Bei der Definition des RIM ist immer abzuwägen, zwischen dem Nutzen, den eine umfangreichere Anforderungsdokumentation bringt und den Kosten, die dadurch entstehen [GLIN2008], [DAVI2005].

Bei der Prüfung eines RIM können folgende Kontrollfragen verwendet werden:

- **Formale Vollständigkeit:** Ist für jede Klasse von Anforderungen klar ersichtlich, welche Anforderungsart und Lösungsunabhängigkeit diese hat, welcher Detaillierungsebene diese zugeordnet ist und mit welcher/welchen Darstellungsform(en) diese dokumentiert wird?
- **Inhaltliche Zusammenhänge:** Ist klar ersichtlich, welche Detaillierungsebenen es gibt und wie diese zusammenhängen? Ist klar ersichtlich, wie die Anforderungen in den verschiedenen Anforderungsklassen voneinander abhängen?

- **Angemessenheit:** Ist die Gesamtheit der Anforderungsklassen geeignet, hinreichend detaillierte und vollständige Anforderungen zu liefern, so dass die davon abhängigen Entwicklungsaktivitäten ihrer Aufgabe nachkommen können?

Insbesondere die Frage der Angemessenheit hängt auch stark vom gewählten Vorgehensmodell ab. Agile Ansätze (vgl. 10) fokussieren beispielsweise auf die Lösungsunabhängigkeit „Ziele“ und „Szenarien“ und lassen die (aufwändigeren) lösungsorientierten Anforderungen typischerweise weg. Durch engen Kundenkontakt, Prototyping und kurze Iterationen können benötigte Informationen zeitnah beschafft werden.

3 Attributierung und Sichten bei Anforderungen (K5)

Dauer: 2 Stunden

Begriffe: Attribut, Attributierungsschema, Sicht

Lernziele

- LZ 3.1 Ziel der Attributierung und den Nutzen von Attributierung in Managementtätigkeiten kennen (K1)
- LZ 3.2 Vorteile von Attributierungsschemata kennen (K2)
- LZ 3.3.1 Die Schritte zum Entwurf eines Attributierungsschemas kennen (K1)
- LZ 3.3.2 Ein Attributierungsschema durch die Ausführung vorgegebener Schritte entwerfen können (K3)
- LZ 3.4.1 Arten von Änderungen von Attributierungsschemata kennen (K2)
- LZ 3.4.2 Änderungen an Attributierungsschemata auf der Basis der unterschiedlichen Arten von Änderungen bewerten können (K5)
- LZ 3.5 Ziele der Sichtenbildung auf Anforderungen und Arten von Sichten kennen (K2)
- LZ 3.6 Die Schritte des Prozesses zur Definition von Sichten kennen (K1)
- LZ 3.7 Den Befüllungsgrad eines Attributes analysieren können (K4)

3.1 Ziele der Attributierung (K1)

Das Management von Anforderungen komplexer Produkte über den gesamten Lebenszyklus erfordert die Verwaltung von zahlreichen Informationen zu einer Anforderung. Um beispielsweise in einem Konfliktfall ermitteln zu können, wer eine konfliktäre Anforderung gestellt hat, ist es notwendig, die Quelle dieser Anforderung zu dokumentieren.

Die Attributierung von Anforderungen hat zum Ziel, dass Stakeholder im Rahmen eines RE-Prozesses strukturiert Informationen zu Anforderungen dokumentieren können [POHL2010], Kapitel 18.2], [IREB2020], Kapitel 8.1]. Dazu werden unterschiedliche Attribute definiert, in welchen diese Informationen (z.B. Quelle, Erstelldatum, Autor, etc.) aufgenommen werden.

Die Attributierung von Anforderungen bildet die Basis für eine Vielzahl von Aufgaben im Requirements Management und für andere Managementtätigkeiten, z.B.:

- **Verfolgbarkeit:** Attribute werden im Rahmen des Requirements Managements zur Realisierung von Verfolgbarkeit eingesetzt. Voraussetzung zum Beispiel für die Verfolgbarkeit der Quelle einer Anforderung – um etwa weitere Details erfragen zu können – ist, dass die Quelle in einem entsprechenden Attribut dokumentiert wurde. Darüber hinaus werden über Attribute auch Verknüpfungen zu jeweils anderen Anforderungen ermöglicht, sodass sich eine Verfolgbarkeit zum Beispiel zwischen Zielen und lösungsorientierten Anforderungen aufbauen lässt.
- **Sichten:** Die Implementierung von Sichten auf Anforderungen basiert in der Regel auf Attributen (siehe 3.6).
- **Priorisierung:** Die jeweilige Priorität wird in einem oder mehreren entsprechenden Attributen dokumentiert. Dabei können mehrere Attribute für unterschiedliche Priorisierungsinhalte definiert werden.

- **Variantenmanagement:** Im Rahmen des Variantenmanagements (siehe 6) können Attribute genutzt werden, um Anforderungen bestimmten Varianten zuzuordnen.
- **Berichtswesen:** Attribute bilden eine Basis zur Erzeugung von Berichten, wie zum Beispiel eine Auswertung über die jeweilige Anzahl von Anforderungen in einem bestimmten Status (wie z.B. „in Arbeit“ oder „überprüft“).
- **Projektmanagement:** Wenn im Rahmen des Projektmanagements der Realisierungsaufwand je Anforderung zusammengetragen wird, muss dieser Realisierungsaufwand in einem entsprechenden Attribut erfasst werden können. Darüber hinaus ist auch die Dokumentation des Status einer Anforderung zur Unterstützung des Projektmanagements notwendig.
- **Releasemanagement:** Die Schnittstelle zum Releasemanagement wird durch ein entsprechendes Attribut „Release“ unterstützt. Somit kann dokumentiert werden, welche Anforderungen in welchem Release umgesetzt werden. Vielfach wird hierbei auch zwischen dem gewünschten und dem geplanten Release unterschieden, um den oftmals auftretenden Unterschied zwischen diesen beiden Releases verdeutlichen zu können.

3.2 Nutzen eines Attributierungsschemas (K2)

Die Menge aller definierten Attribute für eine Anforderungsart (z.B. funktionale Anforderungen, Qualitätsanforderungen) wird als das Attributierungsschema bezeichnet ([PoRu2011], Kapitel 8.1.2). Die Bereitstellung eines Attributierungsschemas für Anforderungen führt zu folgenden Vorteilen im Requirements Management [POHL2010], Kapitel 18.2]:

- **Genau und einheitliche Festlegung der benötigten Informationen:** Durch ein vorgegebenes Schema ist definiert, welche Informationen bzw. Attribute zu Anforderungen erfasst werden müssen und welche Werte für diese Informationen erlaubt sind.
- **Erkennung von Lücken:** Es können Lücken in der Bearbeitung von Anforderungen entdeckt werden, wenn bestimmte Attribute nicht ausgefüllt und damit leer sind.
- **Unterstützung der Einarbeitung von Mitarbeitern:** Mitarbeiter, die z.B. in einem vorherigen Projekt bereits mit dem gleichen oder ähnlichen Attributierungsschema gearbeitet haben, finden schnell die notwendigen Informationen bzw. die Stellen, an denen bestimmte Informationen zur Anforderung dokumentiert werden sollen.
- **Vorfinden von gleichen Informationen an jeweils gleicher Stelle:** Da alle Anforderungen im Rahmen eines Projektes auf der Basis desselben Attributierungsschemas dokumentiert werden, ist festgelegt, wo welche Informationen – wie z.B. der Autor – zu einer Anforderung zu finden sind.

Typische Einflussfaktoren für die Definition eines Attributierungsschemas sind [PoRu2011], Kapitel 8.1, [IREB2020], Kapitel 8.1:

- Spezifische Merkmale eines Projektes, z.B. Projektgröße
- Vorgaben seitens des Unternehmens, z.B. Unternehmensstandards

- Eigenschaften und Vorschriften des Anwendungsgebiets, z.B. Referenzmodelle, Normen
- Randbedingungen und Restriktionen des Entwicklungsprozesses, z.B. Prozessstandards, Normen, etc.

Neben den aus dem RE-Prozess motivierten Vorteilen müssen bei der Definition eines Attributierungsschemas auch die Ziele anderer begleitender Prozesse berücksichtigt werden, wie z.B. für das Projektmanagement (siehe auch 3.1). Diese gewünschten Informationen müssen dann über einen jeweiligen Attributtypen zur Anforderung abgebildet werden.

3.3 Entwurf eines Attributierungsschemas (K3)

Die Definition eines Attributierungsschemas sollte vor dem Start der Anforderungsdokumentation erfolgen und mit allen Stakeholdern des RE-Prozesses abgestimmt sein. Nachträgliche Ergänzungen und Änderungen lassen sich meist nur mit hohem Aufwand durchführen [RuSo2009], Kapitel 14.1.

Für den Entwurf eines Attributierungsschemas zur Nutzung in einem konkreten Projekt sind folgende Schritte notwendig:

1. Identifikation von Attribut-Quellen

Zur Auswahl von Attributen ist es zunächst notwendig, die relevanten Quellen für Attribute zu identifizieren. Quellen, die zur Auswahl von Attributen dienen können, sind z.B.:

- Ein Attributierungsschema aus einem ähnlichen Projekt (z.B. ähnlich hinsichtlich des Umfangs, der Anzahl von beteiligten Mitarbeitern etc.)
- ein Referenzschema des Unternehmens oder ein anderer Standard
- Regeln im Unternehmen, die z.B. festlegen, welche Attribute in allen Attributierungsschemata aller Projekte eingesetzt werden müssen
- Stakeholder des RE-Prozesses

2. Auswahl der Attribute

Für die erfolgreiche Nutzung des Attributierungsschemas im Rahmen des Requirements Managements ist es wesentlich, dass die Attribute zielgerichtet und für das Projekt passend ausgewählt werden. Dies gilt für die Nutzung von Referenzschemata in gleicher Weise wie für die Definition eines Attributierungsschemas ohne Nutzung eines Referenzschemas.

Im Folgenden sind einige Schritte zur Auswahl von Attributen aufgelistet:

- Prüfen von im Referenzschema enthaltenden Attributen, gegebenenfalls die Attribute ändern oder ergänzen.
- Zur Auswahl von Attributen für ein Attributierungsschema sowie die Beurteilung eines Referenzschemas auf Vollständigkeit im Rahmen eines Projektes können die in [POHL2010], Kapitel 18.3 vorgestellten sieben Kategorien zur Unterstützung eingesetzt werden: „Identifizierbarkeit“, „Kontextbeziehungen“, „Dokumentationsaspekte“, „Inhaltsaspekte“, „Übereinstimmungsaspekte“, „Validierungsaspekte“ und „Managementaspekte“.

- Beschränkung der Attribute auf eine pragmatisch einsetzbare Menge durch Fokussierung auf die Ziele des Requirements Managements im Projekt [RuSo2009].
- Festlegung des Kontextes für die Eindeutigkeit der Identifikationsnummer (nur im Projekt eindeutig, im Unternehmen eindeutig etc.).

3. Definition von zulässigen Attributwerten und Eigenschaften der Attribute

Nach der Festlegung, welche Attribute in einem Attributierungsschema genutzt werden, müssen die Eigenschaften der Attribute definiert werden, wie z.B.:

- Um welchen Typ handelt es sich bei dem betrachteten Attribut (Text, Zahl, Wahrheitswert, Datum Aufzählung, etc.)?
- Welche Werte für Aufzählungstypen sollen in einem entsprechenden Attribut zur Auswahl angeboten werden?
- Handelt es sich bei dem Attribut um ein Pflichtfeld?
- Kann das Attribut mehrere Werte oder nur einen Wert aufnehmen?
- Werden für die Auswahl von mehreren Werten Benutzungshilfen, wie z.B. „alle Werte“ oder „kein Wert“ ermöglicht und welche Konsequenzen ergeben sich daraus?

4. Definition von Abhängigkeiten zwischen Attributen und ihren Werten

Attribute können in Bezug auf ihre Werte untereinander abhängen. Beispielsweise kann unterbunden werden, dass eine Anforderung mit dem Eintrag „volatil“ im Attribut „Stabilität“ gleichzeitig im Attribut „Status“ den Eintrag „freigegeben“ erhält. Damit kann sichergestellt werden, dass z.B. nur als stabil eingeschätzte Anforderungen zur Entwicklung freigegeben werden.

Im Rahmen der Definition eines Attributierungsschemas sollte auch festgelegt werden, ob bezogen auf zwei Attribute mit z. B. vordefinierten Attributwerten bestimmte Kombinationen nicht erlaubt sind. In diesem Fall kann es sinnvoll sein, diese beiden Attribute in einem Attribut zusammenzufassen und dort nur noch die erlaubten Kombinationen anzubieten. Dies ist vor allem dann sinnvoll, wenn das eingesetzte Werkzeug keine Unterstützung bei der Betrachtung der Attributwert-Abhängigkeiten bietet.

Im Rahmen des Variantenmanagements kann die Zuordnung von Anforderungen zu bestimmten Varianten untersagt sein. Darüber hinaus ist es denkbar, dass bestimmte Variantenkombinationen nicht erlaubt sind.

Abhängigkeiten zwischen Attributen und ihren Werten können auch durch eine Hierarchisierung von Anforderungen entstehen. Wird eine Anforderung A zum Beispiel durch die beiden Anforderungen A.1 und A.2 detailliert, so ist für die Attribute des Attributierungsschemas festzulegen, ob A in einem Attribut hinsichtlich der Auswahl eines Wertes von den Werten aus A.1 und A.2 abhängt, oder nicht.

5. Bereitstellung einer Erfassungsunterstützung

Die Erfassung von zusätzlichen Informationen zur eigentlichen Anforderung stellt für den Requirements Engineer einen zusätzlichen Aufwand dar, der in manchen Fällen nicht ihm selbst, sondern anderen Stakeholdern zugutekommt (z.B. Projektleiter).

Unter anderem aus diesem Grund ist es für den Requirements Engineer sehr hilfreich, wenn er bei der Erfassung dieser Informationen soweit möglich unterstützt wird.

Diese Art der Unterstützung lässt sich vor allem (z.T. ausschließlich) durch die Nutzung eines entsprechenden Werkzeugs zur Erfassung und Verwaltung von Anforderungen realisieren. Ein einfaches Beispiel dafür ist die Festlegung von Standardwerten für Attribute.

6. Dokumentation des Attributierungsschemas

Attributierungsschemata werden abhängig vom Komplexitätsgrad (z.B. bezüglich der Anzahl der Attribute, Abhängigkeiten zwischen Attributen oder deren Ausprägungen) in Tabellenform oder in einem Informationsmodell (siehe 2) definiert. Die Definition in einem Informationsmodell dient der Nutzung durch ein Werkzeug, welches das entsprechende Attributierungsschema abbildet und den Requirements Engineer bei der Dokumentation der zugehörigen Informationen in einem Projekt unterstützt.

3.4 Änderungsmanagement von Attributierungsschemata (K5)

Nachträgliche Änderungen an einem Attributierungsschema während des Projektverlaufs sollten möglichst vermieden werden [RuSo2009], Kapitel 14.1].

Die Konsequenzen einer nachträglichen Änderung an einem Attributierungsschema richten sich nach der Art der Änderung:

- **Hinzufügen, Ändern oder Löschen eines Attributs:**
Beim Hinzufügen eines neuen Attributs muss bewertet werden, wie aufwändig es ist, die bisher dokumentierten Anforderungen hinsichtlich des neuen Attributs nachzupflegen. Soll ein Attribut gelöscht werden, kann dies negative Konsequenzen haben, wenn beispielsweise in einem Werkzeug Module das Attribut abfragen. Statt es zu löschen, kann man seinen Namen ergänzen durch „(nicht mehr verwendet)“.
- **Hinzufügen, Ändern oder Löschen der möglichen Attributwerte (Wertebereich):**
Das Hinzufügen eines Attributwertes ist in der Regel für das zugrundeliegende Werkzeug kein Problem. Fachlich ist zu untersuchen, ob sich durch das Hinzufügen eine neue Bedeutung der bestehenden Werte eines Attributes ergibt, sodass möglicherweise die bisher bewerteten Anforderungen zu diesem Attribut noch einmal analysiert und gegebenenfalls der neue Attributwert gesetzt werden muss. Beim Löschen von Attributwerten aus einem Wertebereich ist darauf zu achten, dass Anforderungen durch den nun leeren Eintrag im Attribut nicht inkonsistent werden. Probleme bereiten dann vor allem Pflichtfelder, da die Anforderung einen Wert in dem betrachteten Attribut besitzen muss. In diesem Fall kann die Lösung darin liegen, den Standardwert in das Feld einzutragen.
Bei Änderungen von Attributwerten ist zu beachten, dass die Änderungen in allen Anforderungen, die den ursprünglichen Wert beinhalten, durchgeführt werden. Generell muss bei der Änderung sowie Löschung von Anforderungswerten entschieden werden, ob diese Änderung auf bereits erfasste Anforderungen durchschlagen oder nur für zukünftige Anforderungen gelten soll.

Bei abhängigen Attributen mit abhängigen Attributwerten ist sicherzustellen, dass durch die Entfernung eines Attributwertes keine inkonsistenten Daten entstehen.

- **Hinzufügen oder Löschen von Beziehungen zwischen Attributen:**

Das Hinzufügen einer Beziehung kann zu Inkonsistenzen innerhalb der bereits erfassten Anforderungsmenge führen. Wird zum Beispiel die Definition hinzugefügt, dass die Auswahl eines Wertes für das Attribut „Stabilität“ nun immer auch zum Ausfüllen des Attributs „Risiko“ führen soll (und damit immer gefüllt ist, wenn auch „Stabilität“ gefüllt ist), so muss im Nachhinein untersucht werden, welche Anforderungen zwar einen Wert für „Stabilität“, aber keinen für „Risiko“ besitzen und entsprechend noch einmal betrachtet werden müssen.

Wenn die Abhängigkeit zwischen Attributen aufgehoben wird, ist darauf zu achten, dass durch die Beziehung eingerichtete Vorbelegungen (im Sinne von: Wenn Attributwert A ausgewählt wurde, dann wird z.B. in einem Werkzeug automatisch Attributwert B vorbelegt) ebenfalls aufgehoben werden.

- **Ändern von Vorbelegungen für Attributtyp:**

Das Ändern von Vorbelegungen sollte sich zunächst nur auf die Eingabe von neuen Anforderungen auswirken. In diesem Zusammenhang sollte aber analysiert werden, ob Anforderungen, welche die bisherige Vorbelegung erhalten haben, noch mit dem richtigen Wert belegt sind oder gegebenenfalls angepasst werden müssen.

- **Ändern der Eigenschaft von „Pflichtfeldern“ und „optionalen Feldern“:**

Die Änderung von einem Pflichtfeld in ein optionales Feld führt in der Regel zu keinem Folgeaufwand. Ist eine Änderung von einem optionalen Attribut zu einem Pflichtattribut geplant, so ist sicherzustellen, dass für alle bereits dokumentierten Anforderungen das Attribut mit einem entsprechenden Wert gefüllt wird. Dabei kann es notwendig sein, dass hierbei nicht ein Standardwert herangezogen werden kann, sondern individuelle Werte vergeben werden müssen.

Generell ist bei Änderungen an Attributierungsschemata zu analysieren, inwieweit zugrundeliegende Werkzeuge davon betroffen sind. Wurden beispielsweise in dem Werkzeug DOORS Skripte erstellt, die ein bestimmtes Attribut abprüfen oder verarbeiten, kann eine entsprechende Änderung an dem Attribut dazu führen, dass die entsprechenden Skripte nicht mehr lauffähig sind.

3.5 Ziele und Arten von Sichten (K2)

Ein Vorteil der Definition von Attributierungsschemata liegt in der Möglichkeit, Sichten für das Informationsmodell zu definieren und im Rahmen einer Werkzeugunterstützung umzusetzen (siehe auch [IREB2020], Kapitel 8.2).

In einem Projekt kann die Komplexität der Informationen zu den Anforderungen sehr hoch sein, insbesondere bei einer Vielzahl von zu dokumentierenden Anforderungen. Zudem sind im Projekt viele Stakeholder in Bezug auf Anforderungen beteiligt. Dies sind neben dem Requirements Engineer zum Beispiel Entwickler, Tester oder auch Projektleiter.

Ziel der Bereitstellung von spezifischen Sichten ist es, den jeweiligen Stakeholdern zielgerichtet und rollenspezifisch Informationen über die Anforderungen zur Verfügung zu stellen und somit die Komplexität aus der Sicht des jeweiligen Stakeholders zu reduzieren. Darüber hinaus lassen sich über die Bildung von Sichten Zugriffe auf Anforderungen rollenspezifisch regeln (siehe auch [POHL2010], Kapitel 18.5).

Sichten lassen sich wie folgt unterscheiden:

- **Selektive Sichten:** Selektive Sichten schränken die Menge der zu betrachtenden Anforderungen durch einen vordefinierten Filter ein.
- **Projektive Sichten:** Projektiven Sichten blenden Attribute aus, welche für die betrachtete Sicht nicht relevant sind.
- **Verdichtende Sichten:** Verdichtende Sichten ermöglichen Auswertungen (z.B. Bildung von Summen) über eine Anforderungsmenge in Bezug auf bestimmte Attribute.
- **Kombination von selektiven, projektiven und verdichtenden Sichten:** In kombinierten Sichten werden bestimmte Attribute ausgeblendet, die Anforderungsmenge in Bezug auf ein oder mehrere Kriterium/Kriterien eingeschränkt und über z.B. ein Attribut gesondert ausgewertet.

Werkzeuge, die Sichten auf Anforderungen bereitstellen, verfügen in der Regel auch über die Möglichkeit, nach verschiedenen Attributen in wiederum bestimmter Reihenfolge zu sortieren [RuSo2009], Kapitel 14.1. Damit hat der Stakeholder in der jeweiligen Rolle die Möglichkeit, den Fokus auf Anforderungen zu ändern (z.B. alle Anforderungen mit einem hohen Realisierungsaufwand am Anfang einer Liste), ohne Anforderungen auszublenden. Darüber hinaus bieten Werkzeuge oftmals auch an, Sichten durch Gruppierung von Anforderungen nach einem Attribut weiter zu strukturieren.

3.6 Definition von Sichten und Risiken von Sichten (K1)

Der Prozess zur Definition von Sichten umfasst u.a. folgende Schritte:

- **Festlegung der Stakeholder, welche eine oder mehrere Sichten benötigen:** Sichten können unterschiedliche Perspektiven auf Anforderungen bereitstellen. Um notwendige Sichten definieren zu können, sind zunächst die Stakeholder festzulegen, die Sichten auf die Anforderungen nutzen sollen. Die Stakeholder dienen dann als Quelle für die Definition einer Sicht.
Wiederverwendung: Darüber hinaus können Sichten aus anderen Projekten oder aus einem Referenzprojekt als Quelle von zu definierenden Sichten herangezogen werden.
Festlegung der Ziele einer Sicht in Bezug auf die unterschiedlichen Stakeholder: Abhängig von den jeweiligen Stakeholdern muss erarbeitet werden, welches Ziel mit der jeweiligen Sicht verfolgt wird. Daraus lässt sich beispielsweise ableiten, ob Verdichtungen notwendig sind oder welche Sortierung initial eingestellt werden soll. In diesem Zusammenhang müssen auch Rechte von Stakeholdern und Sichten festgelegt werden, d.h. welcher Stakeholder soll welche Sicht aktivieren können.

Dabei ist es sinnvoll, wenn eine Sicht von mehreren Stakeholdern genutzt werden kann.

Festlegung notwendiger Attribute und Abgleich mit dem Attributierungsschema:

Um die Ziele einer Sicht erfüllen zu können, muss sichergestellt werden, dass die notwendigen Informationen auch erhoben werden können und somit die entsprechenden Attribute auch zur Verfügung stehen. Der Abgleich mit dem Attributierungsschema führt aber oft auch dazu, dass neue Sichten generiert werden, weil erst bei der genauen Betrachtung des Attributierungsschemas den Stakeholdern klar wird, welche Auswertungen zum Beispiel noch möglich wären. Die Definition von Sichten und Attributierungsschema ist somit ein sehr integrativer Prozess, beide Vorgänge beeinflussen sich gegenseitig.

Implementierung der Sicht: Schließlich müssen die vordefinierten Sichten in dem zugrundeliegenden Werkzeug umgesetzt und getestet werden.

Die Sichtenbildung birgt einige Risiken. In vielen Fällen ist Anwendern eines Requirements-Management-Systems nicht bewusst, dass sich die Vielzahl von Informationen zu einer Anforderung durch Sichten bedarfsgerecht einschränken lässt.

Sie arbeiten dann oft mit einer globalen und allumfassenden Sicht und empfinden das Werkzeug als zu umfangreich sowie gegebenenfalls in ihrer Arbeit hindernd. Darüber hinaus besteht die Gefahr, Sichten zu definieren, bei denen zu viel Kontext verloren geht. Erstellt man z.B. eine Sicht, in der atomare Anforderungen ohne jeglichen Kontext (z.B. Use Cases) in einer Liste aufgeführt sind, wird diese Übersicht nur beschränkt aussagekräftig sein. Um solch wenig effektive Sichten möglichst zu vermeiden, ist die Betrachtung des zugrundeliegenden Zieles einer Sicht von großer Bedeutung.

3.7 Optimierung von Attributierung und Sichtenbildung [K4]

Die Praxis zeigt, dass in manchen Projekten die Attribute nicht im erwarteten Maße mit Werten belegt werden. Daher ist es zweckmäßig, regelmäßig und an definierten Zeitpunkten im Projekt zu überprüfen, ob und wie ein Attribut noch weiter verwendet werden soll [RuSo2009], Kapitel 14.1.

Eine Möglichkeit, den Befüllungsgrad eines Attributs im Rahmen einer Werkzeugnutzung zu gewährleisten, ist die Festlegung dieses Attributs als „Pflichtfeld“. Dies erzwingt eine Eingabe beim Anlegen einer Anforderung. In diesem Zusammenhang ist zu beachten, dass die Definition von zu vielen Pflichtfeldern die Eingabe von Anforderungen sehr behindern kann, weil zum Beispiel zu dem Zeitpunkt, wenn eine Anforderung das erste Mal erfasst wird, Informationen noch nicht vorliegen. Insofern sollten Pflichtfelder nur sparsam und mit Augenmaß als solche deklariert werden.

Voraussetzung für die Überprüfung der Erfassung von Attributen ist eine entsprechende Auswertung, die in der Regel im Rahmen eines genutzten Werkzeugs durchgeführt werden kann.

Bei optionalen Attributen, deren Erfassung nicht zwingend eingefordert wird, sollten folgende Konsequenzen aus den jeweiligen Auswertungsergebnissen gezogen werden:

- **Das Attribut wurde weder in einer Sicht noch in einem Bericht herangezogen:** Dieser Fall deutet darauf hin, dass das betrachtete Attribut keinem spezifischen Ziel folgt, da es in keiner Sicht oder Auswertung sichtbar ist.
- **Das Attribut ist immer nur mit demselben Wert gefüllt, z.B. mit dem Default-Wert gefüllt:** In diesem Fall scheint es keine echte Unterscheidung im Kontext des Attributs für die verschiedenen Anforderungen zu geben, d.h. die vorgeschlagene Auswahlliste von Werten ist nicht passend. Der Requirements Engineer sollte prüfen, ob entweder das Attribut wegfallen kann (weil es kein echtes Ziel dafür gibt) oder die Auswahlliste entsprechend angepasst werden muss. Im letzteren Fall sollten die Hinweise aus 3.4 berücksichtigt werden.
- **Das Attribut ist nie gefüllt:** Wenn das Attribut bewusst nicht gefüllt wurde, ist die Information möglicherweise nicht wichtig. Bestätigt sich diese Annahme, sollte das Attribut entfernt werden. Oftmals liegt der Grund für das Nichtausfüllen von Attributen aber auch darin, dass den Anwendern die Intention des Attributs nicht bekannt ist oder sie direkt keinen Nutzen sehen, weil die Information von einem anderen Stakeholder genutzt wird. Dann sollten die Anwender (nach)geschult werden, um ihnen die Wertschöpfung des jeweiligen Attributs zu verdeutlichen. In dem Fall kann das Attribut weiter eingesetzt werden.
- **Das Attribut ist nur bei wenigen Anforderungen gefüllt:** Hier ist zu hinterfragen, ob das mit dem Attribut verbundene Ziel erreicht werden kann oder überhaupt noch relevant ist. Ist dies nicht der Fall, kann das Attribut entfernt werden. Stellt sich aber heraus, dass das Attribut wichtig ist, kann es gegebenenfalls als Pflichtfeld deklariert werden, was einen Eintrag in der Zukunft erzwingt. In dem Fall muss eine Nacherfassung der bereits dokumentierten Anforderungen, die keinen Wert in dem betrachteten Attribut besitzen, erfolgen (z.B. automatisierte Befüllung mit einem Standardwert).
- **Das Attribut ist in Einzelfällen nicht gefüllt:** Auch für dieses Attribut muss zunächst ermittelt werden, ob es für das Projekt weiterhin relevant ist. Wenn ja, sollte der Requirements Engineer die entsprechenden Anforderungen vervollständigen. Wird das Attribut nicht mehr als wesentlich eingeschätzt, kann es entweder entfernt oder es können alternativ die Lücken toleriert werden.
- **Das Attribut ist immer gefüllt:** In diesem Fall ist keine weitere Aktivität notwendig

Ergeben sich in Bezug auf die definierten und umgesetzten Sichten aus den Rückmeldungen durch die Stakeholder des RE-Prozesses neue Anforderungen an die jeweilige Sicht, so sollten diese aufgenommen, priorisiert und gegebenenfalls umgesetzt werden. Eine solche Anforderung an die Sichten kann eventuell zum Änderungsmanagement des Attributierungsschemas führen, wenn z.B. ein zusätzliches Attribut oder ein zusätzlicher Attributwert gewünscht ist (siehe 3.4).

4 Bewertung und Priorisierung von Anforderungen (K3)

Dauer: 1,75 Stunden

Begriffe: Priorisierung, Bewertung, Priorisierungstechnik, Ad-Hoc Priorisierungstechniken, Analytische Priorisierungstechniken

Lernziele

- LZ 4.1 Grundlegende Eigenschaften jeder Bewertung kennen (K1)
- LZ 4.2.1 Zusammenhang zwischen Bewertung und Priorisierung kennen (K2)
- LZ 4.2.2 Grundlegende Vorgehensweise bei der Priorisierung kennen (K1)
- LZ 4.3 Ad-hoc Priorisierungstechniken anwenden können (K3)
- LZ 4.4 Analytische Priorisierungstechniken anwenden können (K3)
- LZ 4.5 Vorgehen zur Kombination von Priorisierungstechniken kennen (K2)

4.1 Grundlagen der Bewertung (K1)

In allen Tätigkeiten des Requirements Engineerings werden Anforderungen anhand verschiedenster Kriterien bewertet. So werden Anforderungen z.B. während der Ermittlung nach den Kano-Kriterien kategorisiert (vgl. [IREB2020]) während der Dokumentation anhand der rechtlichen Verbindlichkeit unterschieden, oder Anforderungen aufgrund von einzuhaltenden Normen und Standards (wie z.B. der [ISO 26262]) nach ihrer Kritikalität bewertet.

Die Aufgabe des Requirements Managements ist es, diese Bewertungsergebnisse in geeigneter Form zu dokumentieren (vgl. 3) und die Folgen der Bewertung für den RE-Prozess abzusehen (vgl. 9).

So kann die Bewertung einer Anforderung Auswirkung auf den weiteren Umgang mit der bewerteten Anforderung haben. Eine Anforderung, die als besonders sicherheitskritisch bewertet wurde, könnte beispielsweise während der Prüfung einer detaillierteren Qualitätskontrolle unterzogen werden als eine Anforderung, die als unkritisch gilt.

Mögliche Bewertungskriterien sind z.B.:

- Rechtliche Verbindlichkeit einer Anforderung,
- Implementierungsaufwand/Kosten,
- Kritikalität,
- Stabilität,
- Innovationsgrad.

Quellen für Bewertungskriterien sind z.B.:

- Projektmanagement-Standards,
- Normen und Standards,
- Attributierungsschema der Anforderungen,
- Folgedisziplinen der Entwicklung.

Wichtig ist es, bereits bei der Definition der benötigten Bewertungskriterien die folgenden grundlegenden Eigenschaften jeder Bewertung festzulegen:

- Wertebereiche der Bewertung (z.B. „niedrig“, „mittel“ und „hoch“ für das Bewertungskriterium „Kritikalität“),
- Personengruppe, die die Bewertung durchführen darf (z.B. sollte die „Architekturrelevanz“ nur von Architekten bewertet werden),
- Frühester Zeitpunkt der Bewertung (z.B. darf der Implementierungsaufwand erst geschätzt werden, wenn ein Feindesign vorliegt),
- Spätester Zeitpunkt der Bewertung (z.B. muss der Implementierungsaufwand geschätzt sein, bevor eine Anforderung einem Release oder einer Entwicklungsiteration zugeordnet werden darf).

Die Ergebnisse bereits durchgeführte Bewertungen können im Projektverlauf zur Entscheidungsfindung in allen Tätigkeiten des Requirements Engineerings herangezogen werden [POHL2010]. So können Bewertungen als Basis zur Konsolidierung mithilfe von analytischen Konsolidierungstechniken (vgl. [IREB2020]) oder als Grundlage zur Priorisierung von Anforderungen dienen.

4.2 Priorisierung von Anforderungen (K2)

In keinem Projekt sind die zur Verfügung stehenden Ressourcen unbegrenzt. Um sicherzustellen, dass die vorhandenen Ressourcen zielgerichtet eingesetzt werden, ist es daher unabdingbar, die existierenden Anforderungen zu priorisieren. Grundlegend sollte die Priorisierung stets dem folgenden Prozess folgen:

- Festlegung der Ziele der Priorisierung
- Festlegung der Priorisierungskriterien
- Festlegung der an der Priorisierung beteiligten Stakeholder
- Festlegung der zu priorisierenden Anforderungen
- Auswahl der Priorisierungstechnik
- Ggf. Anpassung des Attributierungsschemas
- Durchführen der Priorisierung
- Regelmäßige Überprüfung und ggf. Neupriorisierung der Anforderungen

Die benötigten Priorisierungskriterien hängen von dem mit der Priorisierung verfolgten Ziel ab. Daher muss in einem ersten Schritt festgelegt werden, welche Entscheidungen auf Basis der Priorisierungsergebnisse getroffen werden sollen. So müssen bei der Entscheidung, welche Funktionen des Systems zuerst im Detail spezifiziert werden müssen, andere Priorisierungskriterien betrachten werden als wenn die Reihenfolge der Implementierung festgelegt wird. Aus diesem Grund ist es sinnvoll, bereits zu Projektbeginn festzulegen, welche Priorisierungen im Projektverlauf benötigt werden und welche Bewertungen dementsprechend bereits vor der Priorisierung durchgeführt werden müssen.

Abhängig vom verfolgten Ziel muss festgelegt werden, anhand welches Bewertungs- bzw. Priorisierungskriteriums oder anhand welcher Kombination aus Kriterien die Priorität der Anforderung festgelegt wird.

Bei der Auswahl der zu priorisierenden Anforderungen ist zu beachten, dass sich die Anforderungen auf derselben Detaillierungsebene befinden sollten, um das Ergebnis der Priorisierung nicht zu verfälschen [WiBe2013].

4.3 Ad-Hoc Priorisierungstechniken (K3)

Zur Priorisierung von Anforderungen existiert eine Reihe von Techniken, die sich im Hinblick auf den Aufwand zur Priorisierung und die Eignung für bestimmte Priorisierungssituationen mitunter sehr unterscheiden. „In Projekten sind einfache Ad-hoc-Priorisierungstechniken [...] ein pragmatischer [...] Ansatz zur effektiven Priorisierung von Anforderungen“ [POHL2010]. Die folgenden Ad-hoc-Priorisierungstechniken haben sich in der Praxis bewährt:

- Requirements Triage [DAVI2003]
- Ranking [PoRu2011]
- Top Ten Technik [PoRu2011]
- Ein-Kriterium-Klassifikation [PoRu2011]
- Planungs-Poker
- Zwei-Kriterien-Klassifikation
- 100-Dollar-Technik
- Kano-Klassifikation [PoRu2011]

4.3.1 Zwei-Kriterien-Klassifikation (K3)

Falls das Ergebnis einer Ein-Kriterien-Klassifikation nicht differenziert genug ist, weil bspw. mehrere Priorisierungskriterien betrachtet werden müssen oder weil zu viele Anforderungen innerhalb derselben Ausprägung des gewählten Kriteriums liegen, können die Ausprägungen mehrerer Kriterien ausmultipliziert und die Ergebnisse über ein Ranking priorisiert werden. Jede Anforderung wird nun einer möglichen Kombination von Ausprägungen zugeordnet und erhält damit die Priorisierung, die dieser Kombination zugeordnet wurde. Dieses Vorgehen wird meist durch Matrizen dargestellt. Eine beispielhafte Matrix mit den Priorisierungskriterien *Kundennutzen* und *Kosten* ist in Abbildung 4 dargestellt.

Weit verbreitet ist auch das nach dem ehemaligen US-Präsidenten benannte Eisenhower-Prinzip, das eine Klassifizierung anhand der Kriterien *Wichtigkeit* und *Dringlichkeit* vorschlägt (vgl. Abbildung 5).

		Kosten		
		Hoch	Mittel	Niedrig
Kundennutzen	Hoch	Priorität 5	Priorität 2	Priorität 1
	Mittel	Priorität 6	Priorität 4	Priorität 3
	Niedrig	Priorität 9	Priorität 8	Priorität 7

Abbildung 4: Beispiel Zwei-Kriterien-Klassifikation

		Dringlichkeit	
		Dringend	Nicht dringend
Wichtigkeit	Wichtig	Priorität 1	Priorität 2
	Nicht wichtig	Priorität 3	Priorität 4

Abbildung 5: Zwei-Kriterien-Klassifikation nach Wichtigkeit und Dringlichkeit

4.3.2 100-Dollar-Technik (K2)

Die 100-Dollar-Technik eignet sich gut für die Priorisierung von wenigen Anforderungen. Die Anwendung dieser Technik wird eher für gröbere Anforderungen auf einer höheren Detaillierungsebene empfohlen oder für Anforderungscluster.

Den Stakeholdern werden 100 imaginäre Einheiten (Geld, Zeit, etc.) zugeteilt, die sie auf die Anforderungen verteilen können. Die beteiligten Stakeholder sollten einen definierten Zeitrahmen haben, um über die Verteilung der ihnen zur Verfügung stehenden Ressourcen nachzudenken, bevor sie diese dann auf die Anforderungen verteilen. Je mehr Einheiten eine Anforderung am Ende erhalten hat, desto höher ist die Priorität dieser Anforderung.

Theoretisch funktioniert die Technik auch mit größeren Mengen an Einheiten (1000, 10000, ...).

Dies kostet jedoch viel mehr Zeit und Aufwand. Es empfiehlt sich, hier ein Werkzeug (z.B. eine Software) zu nutzen, um die Summe der vergebenen Einheiten pro Stakeholder zu überprüfen.

Diese Technik sollte nur einmal auf ein bestimmtes Set von Anforderungen angewendet werden, da die Stakeholder durch die Verteilung der anderen Stakeholder beeinflusst werden könnten und so beim nächsten Durchlauf ihre Einheiten möglicherweise anders verteilen.

4.4 Analytische Priorisierungstechniken (K2)

Teilweise wird der Einsatz von Ad-hoc Priorisierungstechniken einem starken Einfluss durch die in die Priorisierung einbezogenen Stakeholder und sind daher bei sehr kritischen Entscheidungen nur bedingt zu empfehlen. Ein neutraleres Priorisierungsergebnis kann mithilfe der folgenden analytischen Priorisierungstechniken erreicht werden.

- **Wiegers'sche Priorisierungsmatrix:** Die von Karl Wiegers vorgeschlagene Matrix zur Priorisierung von Anforderungen stellt den relativen Vorteil und den relativen Nachteil einer jeden Anforderung den relativen Kosten und dem relativen Risiko einer jeden Anforderung gegenüber. Demnach sind Anforderungen mit hohem Kundennutzen und gleichzeitig geringen Kosten und Risiko höher priorisiert als Anforderungen mit vergleichsweise geringem Kundennutzen und gleichen Kosten und Risiko [WiBe2013].
- **Analytical Hierarchy Process (AHP):** Die Grundidee des AHP ist der paarweise Vergleich aller zu priorisierenden Anforderungen untereinander. Dabei wird auf einer Skala von 1 bis 9 festgelegt, um wie viel wichtiger eine Anforderung A gegenüber einer Anforderung B ist [KaRy1997].

4.5 Kombination von Priorisierungstechniken (K2)

Die analytischen Priorisierungstechniken skalieren im Vergleich zu den Ad-Hoc-Priorisierungstechniken sehr schlecht. So wird sowohl für den Analytical Hierarchy Process als auch die Wiegers'sche Priorisierungsmatrix eine maximale Anzahl von 25–30 zu priorisierenden Anforderungen empfohlen, um die Komplexität und den Zeitaufwand der Methoden im beherrschbaren Rahmen zu halten [WiBe2013], [MOIS2002].

Da in vielen Projekten mit einer weit höheren Anzahl an Anforderungen gearbeitet wird, und eine detaillierte Priorisierung im unteren Bereich der Priorisierungsskala wenig Mehrwert bietet, hat sich eine Kombination aus Ad-Hoc und analytischen Priorisierungstechniken bewährt.

Muss beispielsweise in der Planung eines Release entschieden werden, welche Anforderungen in diesem Release umgesetzt werden sollen, werden in einem ersten Schritt alle Anforderungen ad-hoc priorisiert. Weit verbreitet ist hier das Requirements Triage [DAVI2003], eine an die Medizin angelehnte Ein-Kriterien-Klassifikation, oder das Eisenhower-Prinzip.

Sollte die Ad-Hoc-Priorisierungstechnik mehr oder weniger Anforderungen als umzusetzen identifiziert haben als im zu planenden Release umsetzbar sind, werden die fraglichen Anforderungen in der Nähe der Akzeptanzgrenze mithilfe einer analytischen Priorisierungstechnik priorisiert und die am höchsten priorisierten Anforderungen in das Release übernommen oder umgekehrt diejenigen mit der niedrigsten Priorität entfernt.

5 Versions- und Änderungsmanagement (K3)

Dauer: 2 Stunden

Begriffe: Versionskontrolle, Anforderungskonfiguration, Anforderungsbasislinie, Release, Anforderungsänderung, Änderungsmanagement-Prozess

Lernziele

- LZ 5.1.1 Die Tätigkeiten der Versionskontrolle kennen (K1)
- LZ 5.1.2 Eigenschaften von Anforderungskonfigurationen kennen (K2)
- LZ 5.1.3 Entwicklungsaufgaben, die durch Anforderungsbasislinien unterstützt werden, kennen (K1)
- LZ 5.1.4 Notwendigkeit und Nachteile der Anforderungsverzweigung kennen (K2)
- LZ 5.2.1 Wesentliche Gründe für Änderungen an Anforderungen kennen (K2)
- LZ 5.2.2 Arten von Anforderungsänderungen kennen (K2)
- LZ 5.2.3 Heuristik zur Bewertung der Stabilität/Volatilität von Anforderungen kennen (K2)
- LZ 5.3.1 Ziel und Aufgabe eines Change Control Boards (CCB) kennen (K1)
- LZ 5.3.2 Den Änderungsmanagementprozess anwenden können (K3)

5.1 Versionierung von Anforderungen (K2)

5.1.1 Versionskontrolle für Anforderungen und Anforderungsdokumente (K1)

Die Versionierung von Anforderungen ist wesentlicher Bestandteil des Anforderungsmanagements [WiBe2013]. Versionskontrolle von Anforderungen bezeichnet dabei den Vorgang, der es ermöglicht, über den Lebenszyklus eines Systems oder Produktes hinweg spezifische Entwicklungsstände von Anforderungen und Anforderungsdokumenten verfügbar zu halten [IREB2020]. Stakeholder können damit die Historie von Anforderungen und Anforderungsdokumenten nachvollziehen und können sich stets eindeutig auf einen bestimmten Stand einer Anforderung oder eines Anforderungsdokuments beziehen. Diese Möglichkeiten sind bei allen Projekten hilfreich, aber unterstützen insbesondere kollaborative Projekte.

Laut [WiBe2013] besteht die Versionskontrolle aus drei wesentlichen Tätigkeiten:

1. Definieren eines Schemas zur Kennzeichnung von Versionen
2. Identifizieren der Versionen einzelner Anforderungen
3. Identifizieren der Versionen von Anforderungs-Artefakten

Im gleichen Projekt können Anforderungen sowohl auf der Ebene einzelner Anforderungen (vgl. Tätigkeit 2) als auch auf der Ebene von Anforderungs-Artefakten (vgl. Tätigkeit 3) versioniert sein. Die Versionskontrolle auf Dokumentenebene erlaubt nur einen groben Änderungsnachweis, während auf der Anforderungsebene jede Änderung einer Anforderung genau nachvollziehbar ist. Dafür ist die Versionskontrolle auf Anforderungsebene deutlich aufwändiger als diejenige auf Dokumentenebene.

Versionen einer Anforderung oder eines Anforderungsdokuments müssen eindeutig gekennzeichnet sein. Hierzu muss in jedem Projekt ein entsprechendes Kennzeichnungsschema definiert und durchgängig verwendet werden.

Jeder Projektbeteiligte muss auf die aktuelle Version der Anforderungen zugreifen können. Änderungen müssen deutlich dokumentiert und an die Betroffenen kommuniziert werden [WiBe2013].

Beim Erzeugen einer neuen Version einer Anforderung oder eines Anforderungsdokuments sollten der verantwortliche Projektteilnehmer, der Zeitpunkt der Änderung und der Grund der Änderung erfasst werden [RuSo2009].

5.1.2 Anforderungskonfiguration (K2)

Eine Anforderungskonfiguration fasst eine konsistente Menge logisch zusammengehöriger Anforderungen oder Anforderungs-Artefakte zusammen [IREB Glossary], wobei jede Anforderung und jedes Artefakt maximal in einer Version in der Konfiguration enthalten ist. Eine Konfiguration besitzt laut [POHL2010] folgende Eigenschaften:

- **Konsistenz** – die zusammengefassten Anforderungen und Anforderungsdokumente sind widerspruchsfrei und gehören sachlogisch zusammen.¹
- **Eindeutigkeit** – eine Konfiguration verfügt über einen Bezeichner, der sie eindeutig identifiziert.
- **Unveränderlichkeit** – Änderungen an einzelnen Anforderungen oder Anforderungsdokumenten einer Konfiguration führen zu einer neuen Version einer Konfiguration.

5.1.3 Anforderungsbasislinie (K1)

Eine Anforderungsbasislinie (engl. requirements baseline) ist eine inhaltlich stabile und konsolidierte Anforderungskonfiguration [IREB Glossary], [POHL2010].

Die Basislinie sollte daher nur Anforderungen enthalten, welche für eine bestimmte Version des Produkts (z.B. Release) geplant sind und nicht solche, die vorgeschlagen oder noch in Bearbeitung sind [WiBe2013].

Dadurch entsteht für alle Projektbeteiligten eine eindeutige Kommunikationsgrundlage. Die Versionskontrolle ermöglicht es, den jeweils aktuellen Stand der ausgewählten Anforderungen mit einer Basislinie zu verbinden und später als solche eindeutig zu rekonstruieren.

¹ In der Praxis werden oft auch Konfigurationen gebildet, die nicht inhaltlich konsistent sind. Diese Konfigurationen werden aus der Notwendigkeit heraus gebildet, den aktuellen Arbeitsstand einzufrieren, um später gegebenenfalls zurückgreifen zu können. Beispielsweise kann eine Konfiguration erzeugt werden, die den Startpunkt von Review-Aktivitäten dokumentiert.

Laut [RuSo2009] und [POHL2010] unterstützen Anforderungsbasislinien drei wesentliche Tätigkeiten im Entwicklungsprozess:

- Sie liefern die Grundlage **für das Planen von Auslieferungsständen (Releases)**. Als für den Kunden sichtbare Konfiguration von stabilen Anforderungen eines Produkts dienen sie als Diskussionsgrundlage für die Definition eines Auslieferungszustands.
- Sie dienen der **Abschätzung des Realisierungsaufwands** eines bestimmten Auslieferungszustands.
- Sie ermöglichen den **Vergleich mit Konkurrenzprodukten** am Markt.

Ein geeigneter Zeitpunkt zum Erstellen einer Anforderungsbasislinie kann das Erreichen eines Meilensteins, der Abschluss der Spezifikation eines Teilbereichs oder ein System–Release sein [RuSo2009].

5.1.4 Branching von Anforderungen (K1)

Ein Anforderungszweig (engl. requirements branch) bezeichnet eine Menge von Anforderungen, die zu einem bestimmten Zeitpunkt aus der aktuellen Anforderungskonfiguration kopiert und seither unabhängig vom Original verändert worden sind. Im Gegensatz zu Versionen sind die Anforderungen in verschiedenen Zweigen gleichzeitig gültig [RuSo2009]. Anforderungszweige werden beispielsweise gebildet, wenn kundenspezifische Varianten einer Anforderungsspezifikation für ein Produkt erzeugt werden müssen.

Die Verwaltung von Anforderungszweigen ist kompliziert und stiftet bei undisziplinierter Verwendung mehr Schaden als Nutzen [RuSo2009]. Insbesondere treten folgende Probleme auf:

- Die eindeutige Identifizierung von Anforderungen wird erschwert.
- Neben Versionen und Verfeinerungen bilden Verzweigungen eine dritte Dimension der Anforderungsentwicklung und erhöhen damit die Komplexität des Requirements Managements.
- Verzweigungen erzeugen redundante Anforderungsinformationen, welche parallel gepflegt und langfristig wieder zusammengeführt werden müssen.

5.2 Änderungsmanagement für Anforderungen (K1)

5.2.1 Ursachen, Quellen und Zeitpunkte von Anforderungsänderungen (K2)

Anforderungen an ein (Software–)System unterliegen im Rahmen des Lebenszyklus dieses (Software–)Systems Änderungen. Diese Änderungen können aus unterschiedlichen Gründen notwendig sein:

- Neue oder veränderte Bedürfnisse von Stakeholdern [POHL2010], Kapitel 37.1, **Fehler! Verweisquelle konnte nicht gefunden werden.**, Seite 222

- Änderungen im Systemkontext (z.B. Gesetzesänderungen) [POHL2010], Kapitel 37.1, **Fehler! Verweisquelle konnte nicht gefunden werden.**, Seite 222
- Fehler in bestehenden Anforderungen
- Auswirkungen der Änderung einer Anforderung auf andere, abhängige Anforderungen
- Architektur- und Implementierungsentscheidungen, die Rückwirkungen auf Anforderungen haben

Änderungen an Anforderungen können während der gesamten Entwicklungs- und Nutzungsdauer eines Systems auftreten.

Die sich aus den verschiedenen Zeitpunkten des Auftretens von Änderungen ergebende Komplexität erfordert sowohl einen wohl definierten Prozess als auch geeignete Methoden und Werkzeuge.

5.2.2 Arten von Änderungen an Anforderungen (K2)

Änderungen an Anforderungen werden wie folgt unterschieden:

- Integration einer neuen Anforderung
- Löschen einer bestehenden Anforderung
- Änderung einer Anforderung

Änderungen umfassen dabei nicht nur direkt die Beschreibung der Anforderung, sondern auch weitere Attribute (z.B. Einschätzung der Stabilität) oder Beziehungen zu anderen Artefakten (beispielsweise zu Use Cases oder zu anderen Anforderungen).

5.2.3 Analyse und Dokumentation der Stabilität von Anforderungen (K2)

Anforderungen können hinsichtlich ihrer Stabilität und damit hinsichtlich der Wahrscheinlichkeit, dass die aktuelle Version der Anforderung noch verändert wird, klassifiziert werden. Die instabilen Anforderungen werden eher von Änderungen betroffen sein, als die stabilen Anforderungen. Eine solche Klassifikation kann durch die Bewertung in einem dafür bereitgestellten Attribut durchgeführt werden (vgl. 3). Diese Bewertung ist beispielsweise relevant für die Einschätzung des Risikos, eine spezifizierte Anforderung umzusetzen. Eine bestimmte Bewertung kann auch eine Einschätzung des noch notwendigen Aufwands zur stabilen Dokumentation der Anforderung ausdrücken. Darüber hinaus kann die Bewertung dazu dienen, bei größeren Änderungen an einem System auch Anforderungen gesondert zu betrachten, die prinzipiell volatil sind, weil sie auf Entscheidungen zwischen verschiedenen Alternativen beruhen.

Folgende heuristische Regeln dienen dazu, wahrscheinliche Änderungen bzw. noch instabile Anforderungen zu entdecken und den Fokus darauf zu erhalten (vgl. auch [VANL2009], Seite 224).

- Anforderungsgruppen, die demselben Ziel dienen und grundsätzlich eine hohe Stabilität (gemessen an der Änderungshäufigkeit) aufweisen, haben eine geringere Änderungswahrscheinlichkeit als einzelne Anforderungen.
- Ziele und konzeptuelle Aspekte sind stabiler als lösungsorientierte Anforderungen.
- Funktionale Anforderungen, welche die Kernziele erfüllen, sind stabiler als Qualitätsanforderungen.
- Funktionale Anforderungen, die in der Menge der Anforderungen wiederkehrend (als Verschmelzung, Erweiterung oder Variante) auftauchen, sind in der Regel als stabile Anforderungen zu betrachten.
- Anforderungen, die alternative Auswahlmöglichkeiten beschreiben, sind unter besonderer Vorsicht zu betrachten und in der Regel weniger stabil als oben genannte, da Entscheidungen oft auf unvollständigem Wissen und Annahmen beruhen.
- Anforderungen, welche zwingend einer Variante oder Erweiterung des Systems zugeordnet sind, sind stabiler als noch nicht zugeordnete Anforderungen.
- Anforderungen, die bis vor kurzem häufig geändert wurden, sind voraussichtlich noch nicht als stabile Anforderung anzusehen.
- Anforderungen, die zu einer Gruppe mehrheitlich stabiler Anforderungen gehören, sind stabiler als solche, die zu einer Gruppe mehrheitlich volatiler Anforderungen gehören.

5.3 Änderungsmanagement-Prozess (K3)

Ziel des Änderungsmanagements ist es, jede Änderung an einer oder mehreren Anforderungen vorab zu prüfen, um das damit verbundene Risiko zu kontrollieren, und jede Änderung später nachvollziehen zu können. Dies erreicht der Änderungsmanagementprozess durch die Definition von Aktivitäten, Verantwortlichkeiten und notwendigen Artefakten, die eine klare Vorgehensweise für die Abwicklung von Änderungswünschen an Anforderungen beschreibt.

Eine zentrale Rolle im Änderungsmanagementprozess kommt dem Change Control Board (CCB) zu. Um dessen Aufgabe, nämlich die Bewertung von Änderungsanträgen, bestmöglich ausführen zu können, ist es sinnvoll, das Board mit Rollen aus verschiedenen Bereichen zu besetzen, z.B. [PoRu2011], Kapitel 8.5:

- Kunde
- Architekt
- Requirements Engineer
- Entwickler
- Tester

Dabei ist es essentiell, dass mindestens je ein Vertreter der Kunden- und der Lieferantenseite beteiligt ist.

Die wesentlichen Aktivitäten eines Änderungsmanagementprozesses lassen sich wie folgt beschreiben. Input eines Änderungsmanagementprozesses ist dabei ein Änderungsantrag (engl. change request), welcher in einer für den Prozess vordefinierten Form (Template) beschrieben wird.

Schritt 1: Erstellen des Änderungsantrages

Schritt 2: formale Prüfung des Änderungsantrages

Schritt 3: Im Rahmen der Klassifikation von Änderungsanträgen wird festgestellt, ob es sich um eine korrektive, eine adaptive oder eine Ausnahmeänderung handelt. Dabei ist der Requirements Engineer in die Bewertung eingebunden, um z.B. auch zu entscheiden, welche Ursache eine Änderung hat. Diese Bewertung ist z.B. wichtig, um festzustellen, ob eine Änderung durch den Lieferanten des (Software-)Systems oder durch den Auftraggeber ausgelöst worden ist.

Schritt 4: Die Auswirkungsanalyse hat zum Ziel, die Auswirkungen der Änderungen abzuschätzen und zu dokumentieren. Diese Auswirkungen müssen nicht nur für andere Anforderungen, sondern auch für weitere Artefakte (Architektur, Quell-Code, Testfälle, Schulungsunterlagen) bewertet werden.

Schritt 5: Die Ergebnisse der Auswirkungsanalyse werden vom Change Control Board genutzt um festzulegen, ob ein Änderungsantrag angenommen oder abgelehnt wird. Es ist nicht in jedem Fall sinnvoll, einen Änderungsantrag zu akzeptieren und umzusetzen. Gründe für eine mögliche Ablehnung eines Änderungsantrages sind z.B.:

- Die Änderung ist zu aufwändig und steht in keinem gerechtfertigten Verhältnis zu dem notwendigen Aufwand für die Realisierung oder erwarteten Nutzen.
- Die gewünschte Änderung steht im Widerspruch zu anderen Anforderungen.
- Die Umsetzung der Änderung würde zu einem zu hohen Risiko bezüglich der Stabilität des betrachteten (Software-)Systems führen.
- Die Änderung ist nicht durch einen Vertrag abgedeckt.

Aus Gründen der Nachvollziehbarkeit und zur Abstimmung mit den beteiligten Stakeholdern ist es zwingend notwendig, die Entscheidungen im Rahmen eines Change-Control Boards zu dokumentieren.

Schritt 6: Die angenommenen Änderungsanträge werden durch das Change-Control Board priorisiert.

Schritt 7: Die angenommenen Änderungsanträge werden zur Umsetzung eingeplant und umgesetzt.

6 Verfolgbarkeit von Anforderungen (K3)

Dauer: 2,5 Stunden

Begriffe: Artefakt, Anforderungsartefakt, Verfolgbarkeit, Verfolgbarkeitsmodell

Lernziele

- LZ 6.1 Gründe für Verfolgbarkeit von Anforderungen kennen (K1)
- LZ 6.2 Unterschiedliche Verfolgbarkeits-Betrachtungen kennen (K2)
- LZ 6.3 Beziehungstypen für Verfolgbarkeitsbeziehungen kennen (K1)
- LZ 6.4 Darstellungsformen für Verfolgbarkeitsbeziehungen kennen (K3)
- LZ 6.5 Eine spezifische Strategie für die Verfolgbarkeit erstellen können (K3)
- LZ 6.6 Ein spezifisches Verfolgbarkeitsmodell erstellen und verwenden können (K3)
- LZ 6.7 Maße zur Bewertung von umgesetzter Verfolgbarkeit kennen (K2)
- LZ 6.8 Herausforderungen bei der Verfolgbarkeit von nicht-textuellen Artefakten kennen (K1)

6.1 Gründe für Verfolgbarkeit von Anforderungen (K1)

Die Verfolgbarkeit von Anforderungen hat eine wesentliche Bedeutung für das Requirements Management. Im Kontext des Requirements Managements bezeichnet die Umsetzung von Verfolgbarkeit im Wesentlichen die Pflege von Beziehungen zwischen unterschiedlichen Anforderungen und sonstigen Entwicklungs- bzw. Qualitätssicherungsartefakten.

Die gezielte Pflege von Verfolgbarkeitsbeziehungen ermöglicht es, vorhandene Abhängigkeiten zwischen Artefakten zu kennen, um beispielsweise einen Nachweis der Anforderungsumsetzung zu erbringen oder um zu identifizieren, welche Änderungen die Anpassung einer bestimmten Anforderung nach sich zieht.

In den folgenden Kapiteln dieser Lehreinheit greifen wir in der Wiederholung noch einmal den Begriff der Verfolgbarkeit auf, bevor wir den Nutzen der Verfolgbarkeit erläutern.

In einem ersten Schritt werden die unterschiedlichen Begriffe für die Verfolgbarkeit von Anforderungen erläutert. In der Literatur findet man verschiedene Begrifflichkeiten, welche im Wesentlichen das gleiche meinen: Nachverfolgbarkeit, Verfolgbarkeit, Traceability, Anforderungsverfolgbarkeit, etc. In dieser Lehreinheit werden wir den Begriff **Verfolgbarkeit** verwenden, wenn wir nicht auf eine spezifische Referenz in der Literatur eingehen.

6.1.1 Was versteht man unter Verfolgbarkeit von Anforderungen? (K1)

Unter Verfolgbarkeit von Anforderungen verstehen wir die Fähigkeit, die Abhängigkeiten von Anforderungen zu anderen Artefakten über ihren Entwicklungs- oder Lebenszyklus nachzuvollziehen. Welche Information für die Verfolgbarkeit zu dokumentieren sind, wird durch das Ziel definiert, welches mit der Verfolgbarkeit erreicht werden soll.

Soll beispielsweise durch die Verfolgbarkeit sichergestellt werden, dass alle Businessanforderungen in einem Projekt durch Systemanforderungen abgedeckt wurden, bzw. dass eine Systemanforderung mindestens einer Businessanforderung dient, dann ist unter Umständen bereits eine einfache bidirektionale Referenzierung zwischen diesen Artefakten ausreichend.

6.1.2 Warum Verfolgbarkeit von Anforderungen und Artefakten wichtig ist (K1)

Die Verfolgbarkeit von Anforderungen und anderen Artefakten ist in der Regel kein Projektziel, sondern vielmehr ein Mittel zum Zweck, beispielsweise um einen Nachweis zu erbringen, ob und wie eine Anforderung umgesetzt und getestet wurde. In der Literatur finden sich eine Reihe von Gründen, die die Verfolgbarkeit zwischen Artefakten motivieren, vgl. [HuJD2011], [PoRu2011], [WiBe2013]:

- Belegbarkeit, wie Ziele und Anforderungen erreicht werden sollen.
- Nachweisbarkeit, warum, ob und wie eine Anforderung umgesetzt wurde
- Identifikation von unnötigen Anforderungen und Eigenschaften im System (Goldrandlösungen)
- Identifikation von fehlenden Artefakten (z.B. fehlende Testfälle)
- Vereinfachung der Zurechenbarkeit von Entwicklungsaufwänden zu Anforderungen
- Unterstützung bei der Wiederverwendung von Artefakten
- Unterstützung in Wartung, Pflege und Weiterentwicklung eines Systems

Die Verfolgbarkeit von Anforderungen hilft ferner bei der Beantwortung wichtiger Fragen, beispielsweise nach den Auswirkungen einer Änderung oder warum eine Anforderung überhaupt existiert. Insbesondere die folgenden Analysen werden durch das Vorhandensein von Verfolgbarkeitsbeziehungen erheblich vereinfacht vgl. [HuJD2011], Seite 11ff., [EBER2012], Seite 305 ff., [PMI2021]:

- **Auswirkungsanalyse:** Analyse, welche Artefakte durch eine Änderung (Reduktion oder Erweiterung des Umfangs) beeinflusst werden (vgl. 5.3 Änderungsmanagement-Prozess)
- **Herkunftsanalyse:** Analyse, warum ein bestimmtes Artefakt (z.B. Anforderung) existiert, um bspw. unnötige Anforderungen zu erkennen und zu vermeiden
- **Abdeckungsanalyse:** Analyse, ob bei der Erfassung der Anforderungen bzw. nachfolgender Entwicklungsartefakte alles berücksichtigt wurde, so dass das gewünschte Produkt vollständig erfasst, entwickelt und getestet werden kann
- **Leistungswertanalyse (Earned Value Analyse):** Analyse zur Ermittlung des Arbeitsfortschritts (sog. Leistungswert), um diesen gegen den ursprünglichen Projektplan zu vergleichen und bei Bedarf geeignete Maßnahmen einzuleiten

Darüber hinaus ist die Verfolgbarkeit zwischen Anforderungen und anderen Artefakten notwendig, um bestimmte Reifegrade für Referenzmodelle (z.B. CMMI) oder gesetzliche Randbedingungen (z.B. ISO12207) zu erfüllen.

6.2 Unterschiedliche Verfolgbarkeits-Betrachtungen (K2)

Die Verfolgbarkeit von Anforderungen kann im Wesentlichen durch folgende Dimensionen differenziert werden:

- **Verfolgbarkeit zwischen Anforderung auf derselben Detaillierungsebene.** Diese Art der Verfolgbarkeit beschreibt beispielsweise inhaltliche Abhängigkeiten zwischen funktionalen Anforderungen.
- **Verfolgbarkeit von Anforderungen auf unterschiedlichen Detaillierungsebenen.** Diese Art der Verfolgbarkeit beschreibt beispielsweise Detaillierungen von gesetzlichen Anforderungen zu Systemanforderungen.
- **Verfolgbarkeit zwischen Versionen von Anforderungen:** Diese Art beschreibt die Verfolgbarkeit der Evolution einer Anforderung über die Zeit. Speziell bei dieser Betrachtung ist, dass es zu jedem Zeitpunkt nur eine gültige Version gibt.
- **Verfolgbarkeit zwischen Anforderungen zu nachgelagerten Entwicklungsartefakten.** Diese Art der Verfolgbarkeit beschreibt beispielsweise Abhängigkeiten, welche die Umsetzung / Realisierung der Anforderung bis hin zur System-Komponente oder zum Testfall dokumentieren.
- **Verfolgbarkeit zwischen Anforderungen und vorgelagerten Artefakten.** Diese Art der Verfolgbarkeit beschreibt die Begründung bzw. den Ursprung einer Anforderung.

Die letzten beiden Dimensionen finden sich in der Fachliteratur häufig unter dem Begriff der Pre- und Post Requirements-Specification Traceability [GoFi1994] bzw. der erweiterten Pre- und Post RS Traceability [POHL2010], [PoRu2011].

Pre- und Post Requirements-Specification Traceability

- **Pre-Requirements-Specification Traceability** ist die Verfolgbarkeit von Anforderungen zu ihrem Ursprung also bspw. zu den vorgelagerten Zielen und Visionen oder sonstigen Anforderungsquellen aus dem Systemkontext, z.B. vorhandene Dokumente und Stakeholder.
- **Post-Requirements-Specification Traceability** ist die Verfolgbarkeit von Anforderungen zu nachfolgenden Entwicklungsartefakten, z.B. Architekturentwurf, Implementierung, Test.

Erweiterte Pre- und Post Requirements-Specification Traceability

Zusätzlich zu der Differenzierung in Pre- und Post Requirements-Specification Traceability wird die Verfolgbarkeit zwischen Anforderungsartefakten untereinander betrachtet. Dazu gehören Verfeinerungen und die Verfolgbarkeit zu abhängigen Funktionalitäten, Qualitätsanforderungen, etc.

Abbildung 6 stellt das oben beschriebene Konzept der erweiterten Pre- und Post-Requirements-Specification Traceability aus Sicht der Anforderungen zu vor- und nachgelagerten Artefakten sowie die Verfolgbarkeit zwischen Anforderungen, auf einer und zwischen Abstraktionsebenen, graphisch dar.

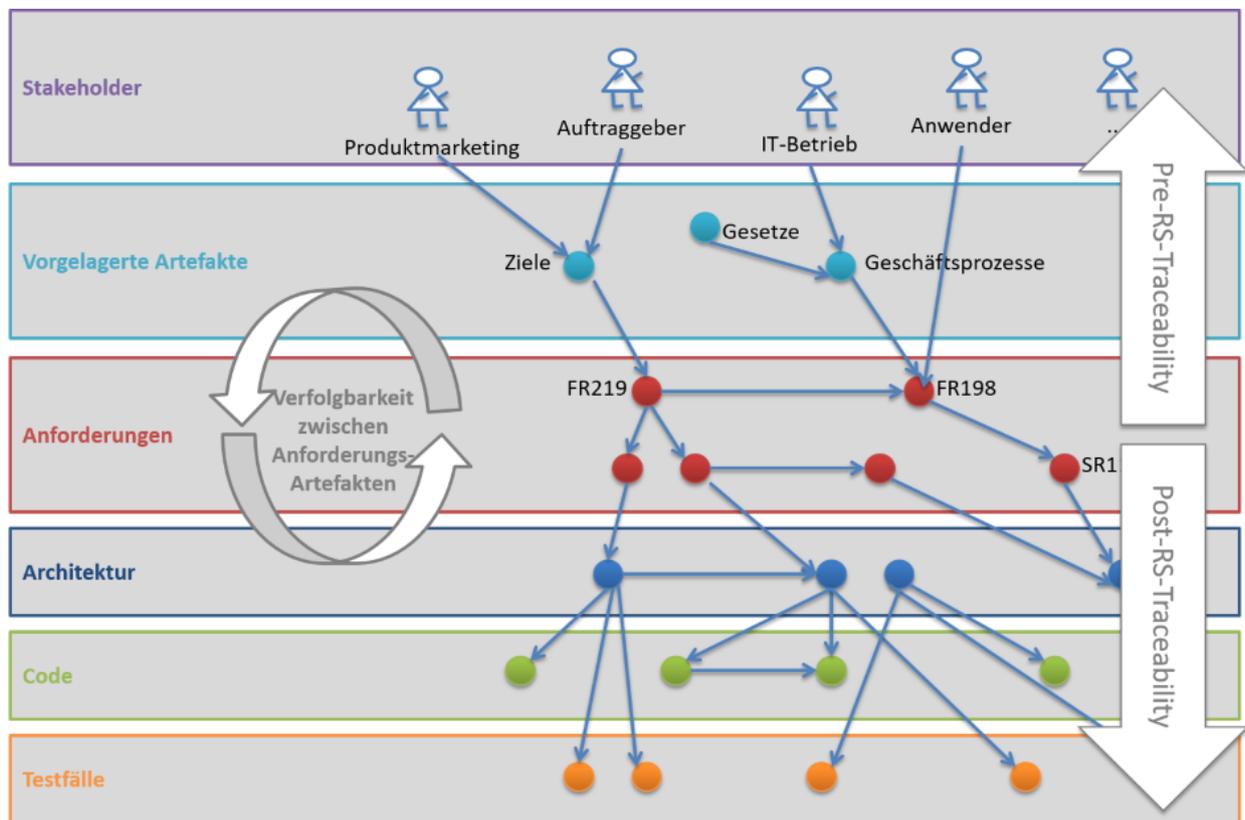


Abbildung 6: Differenzierung von Verfolgbarkeitsbetrachtungen

Wesentlich hierbei ist die Erkenntnis, dass die Verfolgbarkeit nicht bei Anforderungen beginnt oder endet, sondern dass die Verfolgbarkeit einer Anforderung „ganzheitlich“, also über alle Abstraktionsebenen und Phasen – also vom Ursprung (Pre-RS-Traceability) bis zur Umsetzung bzw. Abnahme (Post-RS-Traceability) – erfolgen sollte.

Die Granularität der Verfolgbarkeit kann dabei sowohl auf der Ebene von Einzelanforderungen, aber auch auf der Ebene von Anforderungsgruppen oder Anforderungsartefakten liegen.

6.3 Beziehungstypen für Verfolgbarkeitsbeziehungen (K1)

Es gibt keine allgemein anerkannte Festlegung von Typen für Verfolgbarkeitsbeziehungen. Wesentlich ist, dass Verfolgbarkeitsbeziehungen entsprechend dem Dokumentationsziel eingesetzt werden und dass im Team Einigkeit darüber existiert, welche Beziehungstypen verwendet werden, was diese bedeuten und zwischen welchen Artefakten diese eingesetzt werden.

Theoretisch könnten Verfolgbarkeitsbeziehungen über einen einzigen Beziehungstyp (z.B. steht in Beziehung zu) dargestellt werden. Diese Beziehung sagt dann allerdings nichts darüber aus, in welcher Beziehung die Artefakte zueinander stehen. So kann ein Artefakt eine Detaillierung, eine Realisierung, ein Testfall, eine Variante oder gar ein Widerspruch zu einem anderen Artefakt sein.

In der Literatur findet man aus diesem Grund unterschiedliche Typen von Verfolgbarkeitsbeziehungen, die für eine spezifische Dokumentation von Verfolgbarkeit verwendet werden können.

Pohl unterteilt Verfolgbarkeitsbeziehungen in verschiedene Klassen [POHL2010], denen jeweils unterschiedliche Beziehungstypen zugeordnet werden können:

- **Bedingung:** Unter der Klasse *Bedingung* werden Verfolgbarkeitsbeziehungen zusammengefasst, die inhaltliche Abhängigkeiten zwischen zwei Artefakten beschreiben (Einschränkung, Vorbedingung, etc.).

Beispiel: **Anforderung 1** (ist_Bedingung_für) **Anforderung 12**

- **Inhalt:** Unter der Klasse *Inhalt* werden Verfolgbarkeitsbeziehungen zusammengefasst, die inhaltliche Vergleiche zwischen zwei Artefakten anstellen (Gleichheit, Widerspruch, Konflikt, etc.).

Beispiel: **Anforderung 6** (steht_in_Widerspruch_zu) **Anforderung 10**

- **Dokumentation:** Unter der Klasse *Dokumentation* werden Verfolgbarkeitsbeziehungen zusammengefasst, die weitere Informationen zu einem Artefakt geben (Begründung, Beispiel, Kommentar, Testfall, etc.).

Beispiel: **Test-Artefakt 99** (ist_Testfall_fuer) **Anforderung 3**

- **Abstraktion:** Unter der Klasse *Abstraktion* werden Verfolgbarkeitsbeziehungen zusammengefasst, die Abstraktionsbeziehungen zwischen zwei Artefakten beschreiben (Klassifikation, Aggregation, Generalisierung, etc.).

Beispiel: **Anforderung 43** (generalisiert) **Anforderung 84**

- **Evolution:** Unter der Klasse *Evolution* werden Verfolgbarkeitsbeziehungen zusammengefasst, welche die Art der Weiterentwicklung einer Anforderung beschreiben (erfüllt, verfeinert, ersetzt, erweitert, etc.).

Beispiel: **Anforderung 73, Version 1.2** (ersetzt) **Anforderung 73, Version 1.1**

Welche Beziehungstypen /-Klassen für das jeweilige Projekt tatsächlich zum Tragen kommen, kann nicht pauschal beantwortet werden.

Wichtig ist, sich vor dem Start der Dokumentation (entweder als Unternehmensrichtlinie oder projektspezifisch) Gedanken über das Ziel und die zu verwendenden Verfolgbarkeitsbeziehungen zu machen und diese für alle Beteiligten festzulegen (siehe auch [MaGP2009], [MJZC2013]).

Beispiel: Um sicherzustellen, dass im Rahmen eines Projektes alle Anforderungen begründet sind und dass jede Anforderung getestet wird, sind beispielsweise folgende Typen von Verfolgbarkeitsbeziehungen sinnvoll:

- Beziehung vom Typ „erfüllt“, um sicherstellen, dass keine Anforderung existiert, die keinem Geschäftsziel zugeordnet werden kann
- Beziehung vom Typ „getestet“, um sicherzustellen, dass zu jeder Anforderung ein Testfall existiert
- Beziehung vom Typ „begründet“ um sicherzustellen, dass Entscheidungen zu Anforderungsänderungen dokumentiert wurden

Damit die Festlegung und Verwendung der Verfolgbarkeitsbeziehungen bewusst erfolgt, sollten zum einen die relevanten Artefakte und zum anderen die Beziehungstypen zwischen diesen Artefakten in einem Verfolgbarkeitsmodell (vgl. 6.6) dokumentiert werden. Siehe dazu auch 2.3.

6.4 Darstellungsformen für Verfolgbarkeitsbeziehungen [K3]

6.4.1 Implizite und explizite Dokumentation von Verfolgbarkeit [K3]

Eine Verfolgbarkeitsbeziehung kann entweder implizit oder explizit dokumentiert werden. Auch wenn sich diese Lehreinheit im Wesentlichen auf die explizite Dokumentation von Verfolgbarkeit konzentriert, soll an dieser Stelle auf diesen Unterschied eingegangen werden.

Explizite Dokumentation von Verfolgbarkeit: Die explizite Verfolgbarkeit wird über definierte und bewusst gesetzte Beziehungen zwischen Artefakten erreicht (vgl. Kapitel 6.4.3).

Implizite Dokumentation von Verfolgbarkeit: Eine implizite Verfolgbarkeit kann beispielsweise über Namenskonventionen oder Dokumentenstrukturen erreicht werden.

6.4.2 Bidirektionale und unidirektionale Verfolgbarkeitsbeziehungen (K2)

Verfolgbarkeitsbeziehungen können je nachdem, welches Ziel mit der Verfolgbarkeit erreicht werden soll, unidirektional (gerichtet) oder bidirektional (nicht gerichtet) dokumentiert werden.

- **Unidirektionale Verfolgbarkeit:** erlaubt die Verfolgbarkeit von einem Artefakt zu einem anderen, aber nicht umgekehrt. So erlaubt die Referenz von einer Testanforderung zu einer Systemanforderung beispielweise zu überprüfen, warum die Testanforderung existiert bzw. wovon diese abhängt. Es kann allerdings von der Systemanforderung kein eindeutiger Verweis auf eine Testanforderung gefunden werden. Diese Art der Beziehungen ist häufig in dokumentenbasierten Techniken zu finden, bei denen Beziehungen manuell bspw. durch textuelle Referenzen gepflegt werden und entweder auf das Vorgänger- oder Nachfolge-Artefakt verweisen. Bei der Dokumentationsrichtung ist wichtig zu beachten, dass auf das Artefakt verwiesen wird, zu dem eine Abhängigkeit besteht.
- **Bidirektionale Verfolgbarkeit:** erlaubt die Verfolgbarkeit von einem Artefakt zu einem anderen und umgekehrt. Anders als bei der unidirektionalen Beziehung ist es hier möglich, zwischen den Artefakten zu navigieren, z.B. von der Anforderung aus zum Testfall (z.B. durch eine textuelle Referenz auf einen Testfall) und vom Testfall zur entsprechenden Anforderung, die mit diesem Testfall überprüft werden soll. Diese Art der Beziehungen erlaubt die Betrachtung des Vorgänger und Nachfolge-Artefaktes (Pre- und Post- Requirements-Specification Traceability). In Anforderungsmanagement-Werkzeugen werden diese Beziehungen in der Regel automatisch erzeugt, sodass eine Navigation oder Auswirkungsanalyse in beiden Richtungen durch das Werkzeug unterstützt wird. Bei rein textuellen Referenzen ist hingegen die explizite Pflege je beteiligtem Artefakt notwendig.

6.4.3 Darstellungsformen für Verfolgbarkeitsbeziehungen (K3)

Zur expliziten Dokumentation von Verfolgbarkeit können unterschiedliche Darstellungsformen gewählt werden. Siehe [POHL2010], [PoRu2011]:

- **Textuelle Referenzen:** Die Dokumentation und Repräsentation von textuellen Referenzen ist die einfachste Möglichkeit, um Verfolgbarkeitsbeziehungen zwischen Artefakten zu implementieren (vgl. Abbildung 7). Hierbei beschreibt die Beziehung den Beziehungstyp sowie eine eindeutige Kennung des Artefakts, auf das die Beziehung verweist (z.B. [TEST_CASE_FOR → ReqID 1189]). Diese Art der Repräsentation hat den Vorteil, dass diese unabhängig von einem Anforderungsmanagement-Werkzeug genutzt werden kann und leicht verständlich ist. Sie wird in der Regel direkt beim Artefakt dokumentiert, also bspw. beim Testfall der Verweis auf die Anforderung.

Testfall	Referenz auf Anforderung	Testfallbeschreibung	Priorität
TC_0021	FR_0012 FR_0013 FR_0016	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea ...	Hoch
...			
TC_0150	FR_0020	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed ...	Hoch

Abbildung 7: Textuelle Referenzen

- **Hyperlinks:** Im Gegensatz zu textuellen Referenzen erlauben Hyperlinks die direkte Navigation zum Zielartefakt. Hyperlinks werden dabei immer vom Ausgangsartefakt zum Zielartefakt erstellt. Bidirektionale Beziehungen lassen sich durch entsprechende Kreuzreferenzen erzeugen. Gegenüber einfachen textuellen Referenzen hat der Einsatz von Hyperlinks den Vorteil, dass man direkt zu den referenzierten Artefakten „springen“ kann. Dies ist in der Regel allerdings nur innerhalb eines Werkzeuges möglich.
- **Verfolgbarkeitsmatrizen:** In einer Verfolgbarkeitsmatrix werden Verfolgbarkeitsbeziehungen durch Referenzen in den Zellen einer Matrix dargestellt (vgl. Abbildung 8). Die daraus resultierende Matrix dokumentiert damit die Beziehung vom Ausgangsartefakt zum Zielartefakt. Diese Art der Darstellung erlaubt eine abstrakte Darstellung von Abhängigkeiten zwischen Artefakten, z.B. BR_0010 wird detailliert durch UC_10; BR_0020 wird detailliert durch UC_30 und UC_40; UC_40 wiederum detailliert (rückwärts gerichtete Leserichtung) BR_0020 und BR_0030.
- Verfolgbarkeitsmatrizen können gut dazu verwendet werden, um jeweils genau einen Beziehungstyp zwischen zwei Artefakten zu repräsentieren. Aber auch die Darstellung verschiedener Beziehungstypen ist möglich. Anforderungsmanagement-Werkzeuge wie DOORS erstellen diese Darstellungen automatisiert auf Basis der Artefakte und Beziehungen, und ermöglichen so eine Übersicht der Beziehungen zwischen Artefakten. In der Praxis werden diese Matrizen allerdings schnell sehr groß und nur schwer überschaubar.

		Zielartefakte			
		UC_10	UC_20	UC_30	UC_40
Ausgangsartefakte	BR_0010	wird detailliert durch			
	BR_0011		wird detailliert durch		
	BR_0020			wird detailliert durch	wird detailliert durch
	BR_0030				wird detailliert durch

Abbildung 8: Verfolgbarkeitsmatrix (BR = Business Requirement, UC = Use Case)

- Verfolgbarkeitstabellen:** Im Gegensatz zu Verfolgbarkeitsmatrizen bieten Verfolgbarkeitstabellen die Möglichkeit, Verfolgbarkeitsbeziehungen zwischen allen Artefakten auf unterschiedlichen Detaillierungsebenen zu beschreiben (vgl. Abbildung 9). Sie bieten damit ein mächtiges Hilfsmittel, um die Verfolgbarkeit von Zielen über Use Cases und funktionale Anforderungen bis hin zu Testfällen zu dokumentieren. Dieses Hilfsmittel kann unabhängig von einem speziellen Requirements Management-Werkzeug dazu genutzt werden, um die Verfolgbarkeit zwischen Artefakten zu dokumentieren, die in unterschiedlichen Werkzeugen (Word, Excel, Rational Rose, Visual Paradigm, Quality Center, etc.) dokumentiert sind.

Business Anforderung	Use Case	Funktionale Anforderung	System Anforderung	GUI Element	Testfall
BR_0010	UC_10	FR_0012 FR_0013 FR_0016	CRM_0011 CRM_0020 DWH_0010 Billing_0020	GUI_0081	TC_0021 TC_0022 TC_0025
BR_0011	UC_20	FR_0020	CRM_0011 CRM_0020		TC_0060 TC_0150

Abbildung 9: Verfolgbarkeitstabelle (BR = Business Requirement, UC = Use Case, FR = Functional Requirement, CRM = Customer Relationship Management, DWH = Data Warehouse, GUI = Graphical User Interface / grafische Benutzeroberfläche, TC = test case / Testfall)

- Verfolgbarkeitsgraphen:** In einem Verfolgbarkeitsgraphen stellen die Knoten Artefakte dar und die Kanten repräsentieren die Beziehungen zwischen den Artefakten (vgl. Abbildung 10). Um die unterschiedlichen Entwicklungsartefakte (z.B. Szenario, Anforderung, Testfall) und Beziehungen (z.B. refines, implements, testcase_for) auf einen Blick voneinander unterscheiden zu können, empfiehlt es sich, eine entsprechende Notationsform festzulegen. Der Einsatz ist hier allerdings nur zu empfehlen, wenn diese Graphen werkzeuggestützt auf Basis der Artefakte und Beziehungen automatisiert erstellt werden können. Ein manueller Nachbau solcher Graphen und deren Pflege wird in der Praxis zu aufwändig sein. Grundsätzlich bieten Verfügbarkeitsgraphen allerdings eine gut verständliche Art, die Abhängigkeiten zu überprüfen und zwischen den unterschiedlichen Artefakten zu navigieren.

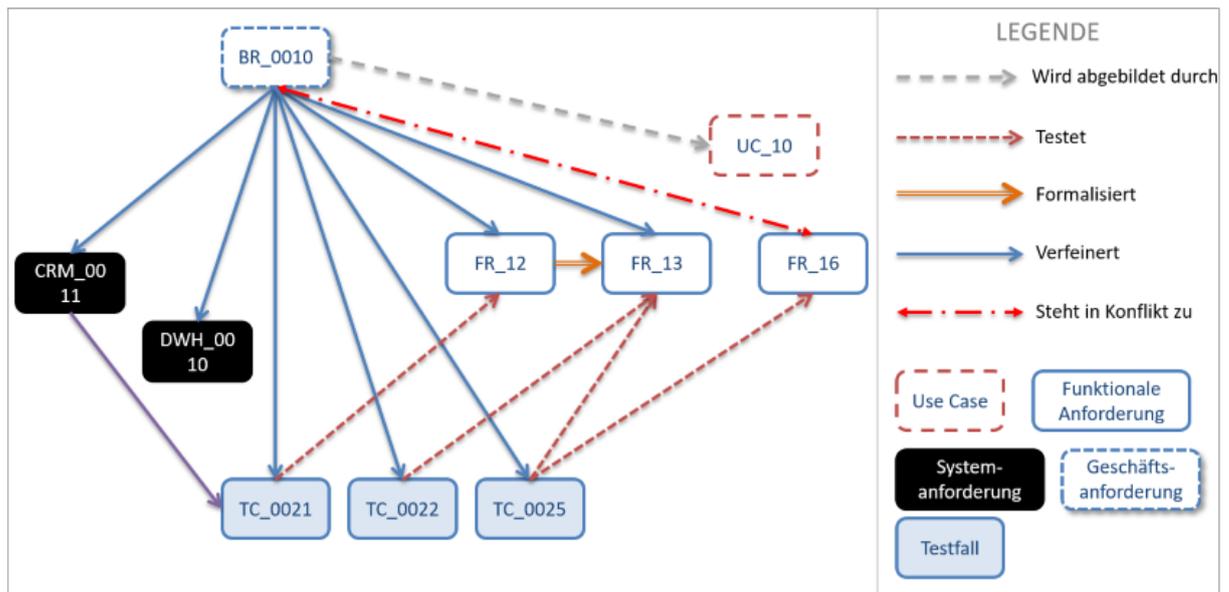


Abbildung 10: Verfolgbarkeits-Graph

Tabelle 1 fasst die beschriebenen Darstellungsformen von Verfolgbarkeitsbeziehungen zusammen und bewertet sie.

Darstellungsform	Positiv	Negativ	Gut geeignet für
------------------	---------	---------	------------------

Inline Dokumentation von Verfolgbarkeit

Textuelle Referenzen	Werkzeugunabhängig und übergreifend umsetzbar Beziehung ist beim Artefakt als Klartext erkennbar.	Verfolgbarkeitsanalysen sind sehr aufwändig.	Zur Repräsentation von Verfolgbarkeit in papierbasierten textuellen Spezifikationen.
-----------------------------	--	--	--

Hyperlinks	Beziehung ist beim Artefakt als Klartext erkennbar. Einfache Navigierbarkeit zwischen Artefakten um direkte Abhängigkeiten zu erkennen.	Verfolgbarkeit zwischen verschiedenen Werkzeugen nicht ohne weiteres möglich.	Zur Repräsentation von Verfolgbarkeit in elektronischen Spezifikationen.
-------------------	--	---	--

Orthogonale Dokumentation von Verfolgbarkeit

Darstellungsform	Positiv	Negativ	Gut geeignet für
Verfolgbarkeitsmatrizen	Abhängigkeit zwischen zwei Artefakten ist schnell und einfach erkennbar.	Manuelle Erstellung von Verfolgbarkeitsmatrizen ist aufwändig und führt zu großen, nur schwach gefüllten Matrizen.	Repräsentation eines einzigen Beziehungstypen zwischen zwei bestimmten Artefakt-Typen (z.B. Use Cases und Anforderungen)
Verfolgbarkeitstabellen	Ermöglicht übersichtliche Darstellung der erweiterten Pre- & Post RS Verfolgbarkeit. Erlaubt vielfältige Verfolgbarkeitsanalysen.	Hohe Komplexität bei der Erstellung.	Repräsentation von Verfolgbarkeit zwischen textuellen und modellbasierten Artefakten in unterschiedlichen Dokumenten / Werkzeugen
Verfolgbarkeitsgraphen	Graphische Darstellung von Verfolgbarkeit; ermöglicht „abstrakte“ Darstellung von Verfolgbarkeitsbeziehungen zwischen Artefakten.	Einsatz nur mit entsprechender Werkzeugunterstützung realisierbar.	Repräsentation komplexer Verfolgbarkeit zwischen Artefakten in einem Requirements Management-Werkzeug.

Tabelle 1: Darstellungsformen von Verfolgbarkeitsbeziehungen

6.5 Erstellung einer Strategie zur projektspezifischen Verfolgbarkeit (K3)

Wie bereits erwähnt muss die Erstellung und Verwendung von Verfolgbarkeit in einem Projekt geplant werden. Es ist in der Regel nicht zielführend, jegliche Beziehung zwischen Artefakten zu dokumentieren. Vielmehr sollte man sich zu Projektbeginn Gedanken darüber machen, wozu Verfolgbarkeit in diesem Projekt notwendig ist und an welchen Stellen welche Art von Verfolgbarkeit benötigt wird.

Neben der Festlegung der relevanten Artefakte und Beziehungstypen ist es für eine spezifische Verfolgbarkeitsstrategie notwendig, auch die Systematik zur Aufzeichnung und Verwendung dieser Informationen festzulegen.

Eine spezifische Verfolgbarkeitsstrategie betrachtet folgende Aspekte:

- **Verfolgbarkeitsziel:** Festlegung aus welchem Grund (also warum) Verfolgbarkeit im Rahmen des jeweiligen Projektes erforderlich ist oder was damit erreicht werden soll (also wozu) (vgl. 6.1.2).

- **Nutzungsstrategie:** Festlegung von Strategien zur Verwendung von Verfolgbarkeitsinformationen durch das Entwicklungsteam. Eine Nutzungsstrategie kann beispielsweise die Auswirkungsanalyse bei Änderungen sein, bei der die Verfolgbarkeitsbeziehungen dazu verwendet werden, um herauszufinden, auf welche Artefakte sich eine Änderung auswirkt.
- **Aufzeichnungsstrategie:** Festlegung von Strategien zur Aufzeichnung von Verfolgbarkeitsinformationen durch das Team. Die Verantwortung für die Dokumentation von Verfolgbarkeitsbeziehungen muss explizit zugewiesen werden, damit diese wahrgenommen wird. Hierbei sollte die Verantwortlichkeit je Beziehungstyp festgelegt werden (z.B. Use Case zu funktionaler Anforderung durch den Business Analyst; funktionale Anforderung zu Testfall durch Quality-Manager). Eine Aufzeichnungsstrategie kann beispielsweise die von [HuJD2011] oder [WiBe2013] vorgeschlagene chronologische Dokumentation von Verfolgbarkeitsbeziehungen sein. Hierbei wird die Beziehung zwischen zwei Artefakten erstellt, sobald das neue Artefakt (z.B. Anforderungsverfeinerung oder Testfall) erzeugt wird. Dies hat den Vorteil, dass eine eindeutige Verantwortung für das Setzen von Verfolgbarkeitsbeziehungen existiert.
- **Projektspezifisches Verfolgbarkeitsmodell:** Festlegung der aufzuzeichnenden Verfolgbarkeitsinformationen sowie der Darstellungsform. Ein Verfolgbarkeitsmodell beschreibt, zwischen welchen Artefakten (z.B. Anforderungen und Testfällen) welche Beziehungstypen (z.B. wird_verfeinert_durch; wird_getestet_durch) in welcher Form (z.B. als textuelle Referenz des jeweils anderen Artefakts an beiden Artefakten) dokumentiert werden sollen (siehe 6.6).

Bei der Festlegung einer Verfolgbarkeitsstrategie ist darauf zu achten, dass:

- Dem gesamten Team die Notwendigkeit der Anforderungsverfolgung bewusst ist,
- das Verfolgbarkeitsmodell verständlich und von allen Beteiligten akzeptiert ist,
- die Teammitglieder die ihnen zugewiesenen Verantwortlichkeiten für die Dokumentation von Nachverfolgungsbeziehungen kennen und akzeptieren,
- die notwendigen Rahmenbedingungen geschaffen werden, um Verfolgbarkeit korrekt zu dokumentieren.

6.6 Spezifische Verfolgbarkeitsmodelle erstellen und verwenden (K3)

Für die Erstellung eines projektspezifischen Verfolgbarkeitsmodells sollte man sich zu Beginn überlegen, zwischen welchen Artefakten Verfolgbarkeit hergestellt werden soll und welche Verfolgbarkeitsbeziehungen zwischen diesen Artefakten notwendig oder erlaubt sind. Diese Festlegungen sollten durch ein projektspezifisches Verfolgbarkeitsmodell (vgl. [MJZC2013] und [POHL1996]) beschrieben und im Projekt kommuniziert werden.

In einem spezifischen Verfolgbarkeitsmodell kann für alle Projektbeteiligten eindeutig dargestellt werden, welche Artefakte existieren, welche Beziehungstypen zu pflegen sind und wer diese wie zu pflegen hat (vgl. [POHL1996], [POHL2010], [MaGP2009]).

6.6.1 Prozess zur Definition eines spezifischen Verfolgbarkeitsmodells

Im Folgenden wird ein exemplarischer Prozess zur Definition eines spezifischen Verfolgbarkeitsmodells beschrieben.

1. Auswahl eines Referenzschemas
Im ersten Schritt sollte geprüft werden, ob man ein bereits vorhandenes Verfolgbarkeitsmodell wiederverwenden und anpassen kann. Eine effektive Möglichkeit, ein spezifisches Verfolgbarkeitsmodell zu definieren, liegt in der Wiederverwendung eines bestehenden Verfolgbarkeitsmodells aus einem ähnlichen Projekt oder eines unternehmensweiten Verfolgbarkeitsmodells. Ein solches Verfolgbarkeitsmodell kann als Basis für die Definition des spezifischen Verfolgbarkeitsmodells dienen und wird in der Regel eine Vielzahl der zu definierenden Artefakte und Abhängigkeiten bereits enthalten.
2. Auswahl der Artefakte
In diesem Schritt wird festgelegt, zwischen welchen Artefakten grundsätzlich eine Verfolgbarkeit gewährleistet werden soll, um das im Rahmen der Verfolgbarkeitsstrategie festgelegte Ziel und die Nutzungsszenarien zu unterstützen, z. B. Verfolgbarkeit zwischen Use Case und funktionaler Anforderung sowie zwischen Anforderung und Testfall.
3. Definition von zulässigen Beziehungstypen zwischen den Artefakt-Typen
Hier ist festzulegen, welche Beziehungstypen zur Abbildung der Verfolgbarkeit (vgl. 6.3) zwischen zwei Artefakt-Typen erlaubt sind, z.B. gültige Beziehung zwischen Anforderung und Testfall: „wird validiert durch“.
4. Festlegung der Anzahl der Verfolgbarkeitsbeziehungen (auf Instanz-Ebene)
Hier wird festgelegt, wie viele Beziehungen zwischen den realen Artefakten (auf Instanz-Ebene des Verfolgbarkeitsmodells) mindestens erwartet werden, z. B. jede Anforderung benötigt eine Verfolgbarkeitsbeziehung zu einem Testfall.
5. Definition der Abhängigkeit zwischen Artefakten
Hier ist festzulegen, welches Artefakt von einem anderen Artefakt abhängig ist, z.B. der Testfall ist abhängig vom Inhalt der Anforderung. Bei der Verwendung von unidirektionalen Beziehungen ist dabei auf die Referenzierung zu achten (vgl. 0)

6.6.2 Verwendung eines spezifischen Verfolgbarkeitsmodells

Neben der Festlegung von Artefakten und Verfolgbarkeitsbeziehungen, z.B. dokumentiert in einem Informationsmodell, sind für die Umsetzung und Verwendung eines spezifischen Verfolgbarkeitsmodells weitere Aspekte zu beachten:

1. Festlegung der Darstellungsform
Nachdem festgelegt wurde, welche Beziehungen zwischen welchen Artefakten dokumentiert werden sollen, ist zu klären, in welcher Art und Darstellungsform Verfolgbarkeitsbeziehungen dokumentiert werden sollen. Die Auswahl der Darstellungsform für Verfolgbarkeitsbeziehungen wird hier in der Regel von der Art der Repräsentation der Artefakte beeinflusst (siehe 6.4.3).

2. Bereitstellung einer Erfassungsunterstützung
Die Erfassung von Verfolgbarkeitsbeziehungen zwischen Artefakten stellt einen zusätzlichen Aufwand dar (siehe **Fehler! Verweisquelle konnte nicht gefunden werden.**), der in der Regel anderen Stakeholdern zugutekommt (z. B. Projektleiter). Aus diesem Grund ist es sehr hilfreich, wenn die Dokumentation von Verfolgbarkeitsbeziehungen soweit möglich unterstützt wird. Diese Unterstützung kann zum einen durch Requirements Management-Werkzeuge – oder aber auch durch bestimmte Eigenprogrammierungen in bspw. Wordmakros unterstützt werden.
3. Erstellen eines Abgleichs von Werkzeug- zu Projektartefakten
Bei der Verwendung eines RM-Werkzeugs benötigt man in der Regel eine Übersetzung in die vorhandene Terminologie des Werkzeugs. In diesem Schritt werden die Bezeichner der im Modell definierten Artefakte und Beziehungstypen zu denen durch das Werkzeug angebotenen Bezeichner in Beziehung gebracht und eindeutig referenziert. Falls das Werkzeug beispielsweise nur einen Artefakttyp „Anforderung“ anbietet, das Verfolgbarkeitsmodell allerdings in „Benutzeranforderung“ und „Systemanforderung“ unterscheidet, dann ist hier eine entsprechende Zuordnung und ggf. eine zusätzliche Attributierung notwendig, die eine spätere Unterscheidung ermöglicht.

6.7 Maße zur Bewertung von umgesetzter Verfolgbarkeit (K2)

Im Rahmen der Verfolgbarkeit stellt sich unweigerlich die Frage, wie gut und vollständig Verfolgbarkeitsinformationen zwischen den Artefakten (Anforderungen, Entscheidungen, Code-Fragmenten, Testfällen, etc.) tatsächlich dokumentiert sind und ob die umgesetzte Verfolgbarkeit ihr eigentliches Ziel (siehe 6.1) erfüllt.

Eine Überprüfung der Verfolgbarkeits-Informationen gibt Aufschluss über die Qualität der aktuellen Dokumentation. Des Weiteren sind diese Ergebnisse zusätzlich hilfreich, um festgefahrene Prozesse oder „untaugliche“ Verfolgbarkeitsmodelle zu identifizieren.

Folgende beispielhafte Maße können dabei helfen, die Vollständigkeit und Qualität der Verfolgbarkeitsbeziehungen zu prüfen:

- Verhältnis der Anzahl von korrekten Verfolgbarkeitsbeziehungen zur Gesamtzahl von Verfolgbarkeitsbeziehungen (Korrektheit)
- Verhältnis der Anzahl der vorhandenen Verfolgbarkeitsbeziehungen zur Gesamtanzahl der benötigten Verfolgbarkeitsbeziehungen (Vollständigkeit)
- Verhältnis der Anzahl von Anforderungen mit Verfolgbarkeitsbeziehung zur Gesamtzahl von Anforderungen (Dichte).

Ein zu geringer Anteil von Beziehungen zwischen Artefakten lässt vermuten, dass die Beziehungen nicht konsequent gepflegt wurden. Ein geringes Verhältnis korrekter Beziehungen hingegen lässt entweder vermuten, dass Beziehungen nachlässig gepflegt wurden oder dass etwaige Änderungen nicht konsequent an allen betroffenen Artefakten durchgeführt wurden. Jede Abweichung vom festgelegten Ziel kann unterschiedliche Gründe haben, die es zu erörtern gilt. Die entsprechenden Schwellwerte, bei deren Unterschreiten Aktionen zu erfolgen haben, sind spezifisch zu definieren.

Neben der Erkenntnis, dass die festgelegte Verfolgbarkeitsstrategie nicht gelebt wird, stellt sich die Frage, warum die Verfolgbarkeit nicht bzw. nur fehlerhaft umgesetzt wird.

Mögliche Gründe für die fehlende bzw. fehlerhafte Dokumentation von Verfolgbarkeit sind:

- Notwendigkeit der Verfolgbarkeit ist nicht bewusst
- Fehlende Verfolgbarkeitsstrategie (Wer dokumentiert was warum)
- Zeitliche Rahmenbedingungen erlauben keine Dokumentation von Verfolgbarkeit
- Es liegt kein abgestimmtes Verfolgbarkeitsmodell vor
- Unzureichende Werkzeugunterstützung bei der Erfassung von Verfolgbarkeitsbeziehungen

6.8 Herausforderungen bei der Verfolgbarkeit von nicht-textuellen Artefakten (K1)

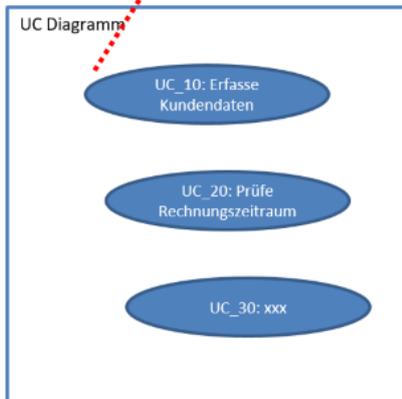
Die Verfolgbarkeit zwischen textuellen Artefakten (z.B. funktionale Anforderungen) und modellbasierten Artefakten (z.B. Aktivitäten in UML-Aktivitätsdiagrammen) bzw. zwischen modellbasierten Artefakten untereinander ist nur mit hohem Aufwand zu erreichen.

Unabhängig davon, wie die Verfolgbarkeitsbeziehungen erstellt wurden, ist die Nutzung dieser Verfolgbarkeitsinformation für Auswertungen und Analysen (z.B. welche Auswirkung hat die Änderung einer Geschäftsanforderung auf die bestehende Software) immer noch aufwändig, da in der Regel kein direktes Öffnen des entsprechenden Artefakts (wie bei einem Hyperlink) möglich ist. Letztendlich macht diese Pflege eine Auswirkungsanalyse aber überhaupt möglich.

Eine werkzeugunabhängige Möglichkeit, Referenzen zwischen textuellen Artefakten und Modellelementen zu dokumentieren, sind bspw. Verfolgbarkeitstabellen (vgl. 6.4.3), die sowohl die textuellen Artefakte als auch die Modellelemente eindeutig referenzieren können. Hierzu ist es allerdings notwendig, die Elemente im Modell mit eindeutigen IDs (manuell oder systemisch) zu versehen. Über eine textuelle Referenz oder über eine Verfolgbarkeitstabelle hat man so bspw. die Möglichkeit, Modellelemente wie Use Cases zur entsprechenden textuellen Anforderung eindeutig zu referenzieren.

Abbildung 11 zeigt ein einfaches Beispiel für eine Referenzierung von textuellen Anforderungen und Use Cases in einem Use Case Diagramm über eine Verfolgbarkeitstabelle.

Business Anforderung	Use Case	Funktionale Anforderung	System Anforderung	GUI Element	Testfall
BR_0010	UC_10	FR_0012 FR_0013 FR_0016	CRM_0011 CRM_0020 DWH_0010 Billing_0020	GUI_0081	TC_0021 TC_0022 TC_0025
BR_0011	UC_20	FR_0020	CRM_0011 CRM_0020		TC_0060 TC_0150



FR_ID	Funktionale Anforderung	Priorität
FR_0012	Der Name des Kunden muss vor der Erfassung gegen die Bestandskundendaten geprüft werden.	Hoch
FR_0013	Die Bankverbindung des Kunden muss gegen den SEPA HUB validiert werden	Hoch
...

Abbildung 11: Verfolgbarkeit zwischen textuellen und modellbasierten Artefakten

7 Variantenmanagement für Anforderungen (K5)

Dauer: 3 Stunden

Begriffe: Variabilität, Wiederverwendung, Produktfamilie, Produktlinie, Merkmal (Feature), Merkmalsmodell (Featuremodell), Variationspunkt, Variante, Bindezeitpunkt

Lernziele

- LZ 7.1.1 Gründe für den Einsatz von Varianten für Anforderungen kennen (K1)
- LZ 7.1.2 Zentrale Begriffe im Bereich Variantenmanagement für Anforderungen kennen (K1)
- LZ 7.1.3 Vorteile der expliziten Dokumentation von Variabilität kennen (K2)
- LZ 7.2.1 Gängige Darstellungsformen der Variantenbildung von Anforderungen kennen (K3)
- LZ 7.2.2 Eine gegebene Darstellungsform von Variabilität anhand vorgegebener Kriterien beurteilen können (K5)
- LZ 7.2.3 Eine gegebene Darstellungsform von Variabilität im Hinblick auf eine Einsatzsituation in einem Kontext analysieren können (K4)
- LZ 7.3.1 Das Konzept der Merkmalsmodellierung kennen (K1)
- LZ 7.3.2 Ein Merkmalsmodell erstellen können (K3)

7.1 Einsatz von Varianten für Anforderungen (K1)

Häufig wird im Rahmen einer Produktentwicklung nicht nur ein solitäres Produkt erstellt, sondern es entstehen – parallel oder sequentiell – eine Reihe ähnlicher Produkte. In diesem Fall ist es zumeist sinnvoll, die Produktentwicklungen nicht unabhängig voneinander zu betreiben, sondern Entwicklungsartefakte wie Anforderungen, Architekturen, Programmcode oder Testfälle geeignet wiederzuverwenden [ISO 26550].

Erfolgt die Entwicklung ähnlicher Produkte, indem Entwicklungsartefakte systematisch und geplant wiederverwendet werden, spricht man von einer Produktlinie oder Produktfamilie [CINo2007].

Die Definition des Begriffs Produktlinie ist wie folgt: *“A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way.”* [aus [C1No2007]].

Zu den Kern-Entwicklungsobjekten (core assets) gehören typischerweise auch Anforderungen. Die Anforderungen an eine Menge von Produkten werden nicht unabhängig voneinander verwaltet, sondern die Anforderungen liegen in geeigneter Weise in einem gemeinsamen „Anforderungspool“ und werden den einzelnen Produkten zugeordnet.

Der Produktlinien-Gedanke kann auch nur auf der Ebene von Anforderungen beschränkt werden. Auch in diesem Fall werden Anforderungen an verschiedene Produkte in einem „Anforderungspool“ verwaltet und aus diesem werden dann die Anforderungen an spezielle Produkte abgeleitet, auch wenn die Produktentwicklung der verschiedenen Produkte dann unabhängig voneinander erfolgt. In diesem Fall spricht man von einer Anforderungs-Produktlinie.

Ein Anforderungspool ist in beiden Fälle eine Menge von Anforderungen, die mehr beinhaltet als die Menge der Anforderungen an ein konkretes Produkt. Ein Anforderungspool kann auch Anforderungen beinhalten, die aktuell in keinem Produkt Berücksichtigung finden.

Bei der Produktlinienentwicklung ist es üblich, zwei Prozesse zu unterscheiden [WeLa1999], [ISO 26550]:

- Domänen-Engineering: Im Domänen-Engineering wird ein „Grund-Produkt“ entwickelt
- Applikations-Engineering: Hier wird das „Grund-Produkt“ produktspezifisch angepasst.

Ein zentrales Konzept bei der Wiederverwendung von Anforderungen ist die **Variabilität** [POHL2010], [PoBL2005]. Variabilität ermöglicht die Definition und Realisierung unterschiedlicher Produkte durch Auswahl von konkreten Varianten aus einer vorgegebenen Menge von möglichen Varianten.

Jede Variante bezieht sich immer auf einen Variationspunkt. Variationspunkte beschreiben, wo innerhalb einer Produktlinie etwas variiert, Varianten beschreiben mögliche (zulässige) Ausprägungen (von Anforderungen) an einem Variationspunkt. In der Regel finden sich in einem Anforderungsdokument eine Vielzahl an Variationspunkten mit typischerweise mehreren Varianten. Allerdings ist nicht jede Kombination einzelner Varianten zulässig. Es gibt typischerweise Variantenabhängigkeiten. Die Variantenabhängigkeiten können dabei optional oder obligatorisch sein. Optional bedeutet, dass eine Variante gewählt werden kann, aber nicht muss. Obligatorisch bedeutet, dass eine Variante (z.B. aus mehreren Alternativen) gewählt werden muss.

Durch die Auswahl konkreter Varianten für jeden Variationspunkt wird ein spezifisches Produkt „konfiguriert“. Werden dabei alle Variantenabhängigkeiten berücksichtigt, so entsteht ein zulässiges, spezifisches Produkt.

In der Praxis werden nicht alle Variationspunkte während der Anforderungserhebung mit konkreten Varianten belegt, d.h. eine von mehreren möglichen Varianten gewählt. Einige Variationspunkte bleiben offen, d.h. sie werden noch nicht an eine konkrete Variante „gebunden“. Der Bindezeitpunkt (vgl. [CoHW1998]) kann dabei bis zur Auslieferung des Systems oder auch bis in den Systembetrieb offen bleiben.

Mögliche Bindezeitpunkte sind vor der Entwicklung, „während der Entwicklung“ (implementation time), „während des Zusammenbaus“ (build time), „während der Inbetriebnahme“ (installation time), „während des Systemstarts“ (startup time) sowie „während der Laufzeit“ (runtime) [ATKI2002].

Variabilität kann implizit oder explizit dokumentiert werden. Im Falle einer impliziten Dokumentation muss der Leser aus der Formulierung der Anforderung erkennen, dass hier verschiedene Produktvarianten (Produktausprägungen) möglich sind.

Bei einer **impliziten** Dokumentation wird durch das Wort „oder“ oft signalisiert, dass verschiedene Produktausprägungen möglich sind. Allerdings ist das Wort „oder“ kein zuverlässiger Indikator für einen Variationspunkt, da dieses ebenfalls sehr häufig in logischen Bedingungen verwendet wird. Sprachlich sind darüber hinaus auch andere Schlüsselbegriffe möglich, die ebenfalls verschiedene Produktausprägungen signalisieren (z.B. „sowohl ... als auch“). Auch diese sind hierfür i.d.R. nicht eindeutig genug.

Die **explizite** Dokumentation von Variabilität kann integriert in die Dokumentation der Anforderung erfolgen oder orthogonal, d.h. in einem separaten Modell. Im Falle textueller Anforderungen finden sich bei einer integrierten Dokumentation sowohl die Variationspunkte als auch die möglichen Varianten im Anforderungstext explizit ausgewiesen wieder.

Im Falle einer orthogonalen Dokumentation bleibt die textuelle Anforderung unangetastet. Die Dokumentation von Variationspunkten und Varianten erfolgt in einem separaten Modell.

In der Produktlinien-Entwicklung im Allgemeinen und im Requirements Management im Speziellen hat die explizite Dokumentation von Variabilität folgende Vorteile [POHL2010]:

- **Kommunikation:** Die explizite Dokumentation von Variationspunkten und möglichen Varianten unterstützt die Kommunikation mit den betroffenen Stakeholdern, da hier einfach ersichtlich wird, an welchen Stellen welche Varianten ausgewählt werden können.
- **Entscheidungsunterstützung:** Die explizite Dokumentation führt zu bewussteren Entscheidungen, (1) an welchen Stellen Variabilität vorgesehen wird und (2) welche konkrete Variante für ein gegebenes Produkt ausgewählt wurde.
- **Verfolgbarkeit:** Durch eine explizite Dokumentation von Variantenabhängigkeiten können Abhängigkeiten analysiert und im Falle von Anforderungsänderungen zur Analyse etwaiger Folgeänderungen herangezogen werden.

7.2 Formen expliziter Dokumentation von Varianten und deren Bewertung (K5)

In der Praxis finden sich viele verschiedene Formen der Dokumentation von Varianten in Anforderungsdokumenten. Diese Formen setzen die in Abschnitt 7.1 eingeführten Konzepte wie Variationspunkt, Variante, Produktzuordnung und Dokumentation von Bindungszeitpunkten sehr unterschiedlich um.

Im Folgenden werden zuerst gängige Darstellungsformen [BOUT2011] skizzenhaft vorgestellt und anhand eines kurzen Beispiels aus der Automobilindustrie illustriert. Dann werden diese Darstellungsformen bezüglich der in 7.1 eingeführten Konzepte analysiert. Anschließend werden Kriterien zur Bewertung von Stärken und Schwächen der Darstellungsformen identifiziert und die vorgestellten Formen bewertet.

Eine weitere Darstellungsform stellen Merkmalsmodelle dar. Diese werden in Abschnitt 7.3 näher betrachtet.

Form 1: Textuelle Zuordnung von Anforderungen zu konkreten Produkten

In diesem Fall werden in den einzelnen Anforderungen die betroffenen Produkte explizit benannt (Abbildung 12).

ID	Anforderung
R32	Die Sonnenblende der A-Klasse soll kunststoffbeschichtet sein.
R33	Die Sonnenblende der E-Klasse soll mit Leder bezogen sein.
R34	Die Sonnenblende in allen Produkten soll einen beleuchteten Kosmetikspiegel beinhalten.

Abbildung 12: Textuelle Zuordnung von Anforderungen zu konkreten Produkten

Form 2: Explizite Zuordnung von Anforderungen zu konkreten Produkten

In diesem Fall werden die einzelnen Anforderungen direkt den betroffenen Produkten (Produktvarianten) zugeordnet (Abbildung 13).

ID	Anforderung	A-Klasse	E-Klasse
R32	Die Sonnenblende soll kunststoffbeschichtet sein.	X	
R33	Die Sonnenblende soll mit Leder bezogen sein.		X
R34	Die Sonnenblende soll einen beleuchteten Kosmetikspiegel beinhalten.	X	X

Abbildung 13: Explizite Zuordnung von Anforderungen zu konkreten Produkten

Bezüglich der konkreten Ausprägung der direkten Zuordnung gibt es wiederum verschiedene Unterausprägungen:

- Explizite Zuordnung zu separaten Produktspalten
- Mehrfachauswahl in einer Produktspalte (siehe auch erste Produktspalte in Abbildung 14)

Form 3: Explizite Zuordnung von Anforderungen zu konkreten Produktmerkmalen

In diesem Fall werden einzelne Anforderungen mehreren Produktmerkmalen direkt zugeordnet (Abbildung 14). Ein konkretes Produkt wird durch mehrere Produktmerkmale definiert, die jeweils verschiedene Ausprägungen haben können. Eine Anforderung gehört zu einem Produkt, wenn die Anforderung den zum Produkt gehörenden Produktmerkmalen zugeordnet ist. Zum Produkt „E-Klasse in den USA“ gehören beispielsweise die Anforderungen R33 und R34.

ID	Anforderung	Baureihe	Markt
R32	Die Sonnenblende soll kunststoffbeschichtet sein.	A-Klasse	USA Europa
R33	Die Sonnenblende soll mit Leder bezogen sein.	E-Klasse	USA
R34	Die Sonnenblende soll einen beleuchteten Kosmetikspiegel beinhalten.	A-Klasse E-Klasse	USA Europa

Abbildung 14: Explizite Zuordnung von Anforderungen zu konkreten Produktmerkmalen

Hinweis: Diese Darstellungsform birgt die Gefahr, dass ungültige Konfigurationen aufgrund der Kombinatorik plötzlich gültig werden. Beispiel: Im Fall von R34 soll die Anforderung nur für die A-Klasse USA sowie die E-Klasse USA und E-Klasse Europa gelten. Der Ausschluss von A-Klasse Europa lässt sich mit der gewählten Klassifikationsmethodik nicht darstellen.

Form 4: Indirekte Zuordnung von Anforderungen zu Produkten über Merkmale

In diesem Fall werden die einzelnen Anforderungen Produktmerkmalen zugeordnet. In einer separaten Produktkonfiguration wird dann festgelegt, welche Produktmerkmale in einem spezifischen Produkt (d.h. einer spezifischen Produktvariante) enthalten sind, wodurch indirekt die relevanten Anforderungen ausgewählt werden (Abbildung 15).

ID	Anforderung	Feature
R32	Die Sonnenblende soll kunststoffbeschichtet sein.	Kunststoffoberfläche
R33	Die Sonnenblende soll mit Leder bezogen sein.	Lederoberfläche
R34	Die Sonnenblende soll einen beleuchteten Kosmetikspiegel beinhalten.	

Produkt	Merkmale
A-Klasse	Kunststoffoberfläche und ...
E-Klasse	Lederoberfläche und ...

Abbildung 15: Indirekte Zuordnung von Anforderungen zu Merkmalen und Produktkonfiguration

Anforderungen, die keinem Merkmal zugeordnet sind, gelten hier für alle Produkte.

Analyse der Darstellungsformen

Bei der Analyse von Darstellungsformen sind die folgenden, in 7.1 vorgestellten Aspekte zu betrachten:

- **Beschreibung von Bindezeitpunkten**

Alle vorgestellten Darstellungsformen beschränken sich nur auf einen Bindezeitpunkt. In diesen Fall wird nur der Bindezeitpunkt „während der Entwicklung“ (implementation time) betrachtet.

- **Variationspunkte und Varianten**

Variationspunkte werden bei diesen Darstellungsformen nur indirekt erkennbar, indem es mehrere Anforderungen gibt, die augenscheinlich im Widerspruch zueinander stehen und erst durch die Zuordnung der Varianten zu konkreten Produkten der Widerspruch aufgelöst wird (vgl. Anforderungen R32 und R33). Die konkreten Varianten sind direkt im Anforderungstext beschrieben.

- **Variantenabhängigkeiten und Prüfmöglichkeit**

Variantenabhängigkeiten werden in den vorgestellten Fällen nicht (Form 1, 2, 3) oder nur indirekt (Form 4) dokumentiert und können folglich auch nicht überprüft werden. Fehler in der Konfiguration können im besten Fall durch einen menschlichen Prüfer identifiziert werden, wenn beispielsweise widersprüchliche Anforderungen für dasselbe Produkt ausgewählt werden (wenn beispielsweise R32 und R33 für die A-Klasse gewählt worden wären). In Form 4 kann indirekt auf eine Variantenabhängigkeit geschlossen werden, wenn es mehrere Anforderungen gibt, die sich auf dieselben Merkmale beziehen.

Stärken- und Schwächen-Analyse

Bei der Bewertung einer konkret eingesetzten Darstellungsform für Variabilität sind folgende Kriterien für die praktische Anwendung von Relevanz [BOUT2011]:

- **Vermittelbarkeit:** Wie einfach kann die gewählte Darstellungsform fachfremdem Personal geschult werden?
- **Skalierbarkeit:** Wie leicht kann die gewählte Darstellungsform auch für eine größere Anzahl von Produkten sinnvoll eingesetzt werden?
- **Erweiterbarkeit:** Wie viel Aufwand ist notwendig, um ein neues Produkt zu konfigurieren?
- **Migrierbarkeit:** Inwieweit können bestehende Anforderungs-Dokumentationen ohne explizite Variabilitätsinformation in Richtung der gewählten Darstellungsform weiterentwickelt werden?
- **Überprüfbarkeit:** Inwieweit können Fehlkonfigurationen in der gewählten Darstellungsform automatisiert identifiziert werden?
- **Vergleichbarkeit:** Inwieweit können die Anforderungen verschiedener Produkte einfach miteinander verglichen werden?
- **Änderbarkeit:** Wie einfach können bestehende Anforderungen für ein einzelnes Produkt geändert werden ohne Auswirkung auf andere Produkte der Produktfamilie?

7.3 Merkmalsmodellierung (K3)

Merkmalsmodellierung (feature modeling) ist eine gängige Technik zur Dokumentation von Variabilität. Der bekannteste Vertreter der Merkmalsmodellierung ist FODA – Feature Oriented Domain Analysis [KCeA1990]. Mittlerweile ist Merkmalsmodellierung vor allem in der Produktlinienentwicklung weit verbreitet und es gibt eine Vielzahl an Erweiterungen und Ergänzungen zum ursprünglichen FODA-Ansatz [ScHT2006].

Ein Merkmal (feature) ist dabei definiert als „*prominent or distinctive user-visible aspect, quality, or characteristic of a software system or system*“ [KCeA1990]. Ein Merkmalsmodell beschreibt Merkmale sowie deren Abhängigkeiten untereinander. Eine Produktvariante besteht aus einer Menge von Merkmalen, die das Produkt beschreiben.

Eine (gültige) Produktkonfiguration und die darin enthaltenen Merkmale werden definiert durch die Randbedingungen, die das Merkmalsmodell vorgibt. Merkmalsmodelle werden häufig graphisch in Form eines Merkmalsdiagramms dargestellt.

Die Beschreibungselemente eines Merkmalsmodells lassen sich in die folgenden drei Kategorien einteilen:

- Basiselemente
- Erweiterte Elemente
- Kardinalitätsbasierte Elemente

Mit den Basiselementen eines Merkmalsmodells werden Eltern-Merkmale sowie deren Kinder beschrieben und deren Verhältnis zueinander ausgedrückt. Kinder-Merkmale können zu Eltern-Merkmalen folgende Beziehungen aufweisen:

- Verpflichtend – Das Kind-Merkmal ist zwingend notwendig
- Optional – Das Kind-Merkmal kann verwendet werden
- Oder – Mindestens eines der Kind-Merkmale muss ausgewählt werden
- Alternative – Genau eines der Kind-Merkmale muss ausgewählt werden

Mit den erweiterten Elementen können ergänzend Abhängigkeiten zwischen Merkmalen festgelegt werden. Die bekanntesten Abhängigkeiten sind dabei

- A benötigt B – Die Auswahl von Merkmal A impliziert die Auswahl von Merkmal B.
- A schließt B aus – Die Merkmale A und B können nicht im gleichen Produkt enthalten sein.

Mit kardinalitätsbasierten Elementen können die erlaubten Beziehungen der Basiselemente weiter präzisiert werden, indem beispielsweise Notationen wie [min, max] an den Eltern-Kind Beziehungen annotiert werden.

Die im Folgenden verwendete Notation entstammt [CzEi2000].

Abbildung 16 zeigt ein exemplarisches Merkmalsmodell. Das Merkmal F besitzt zwei verpflichtende Merkmale f1 und f4, wobei f1 seinerseits wieder zwei verpflichtende Merkmale f2 und f3 besitzt, während f4 nur ein optionales Merkmal f5 hat.

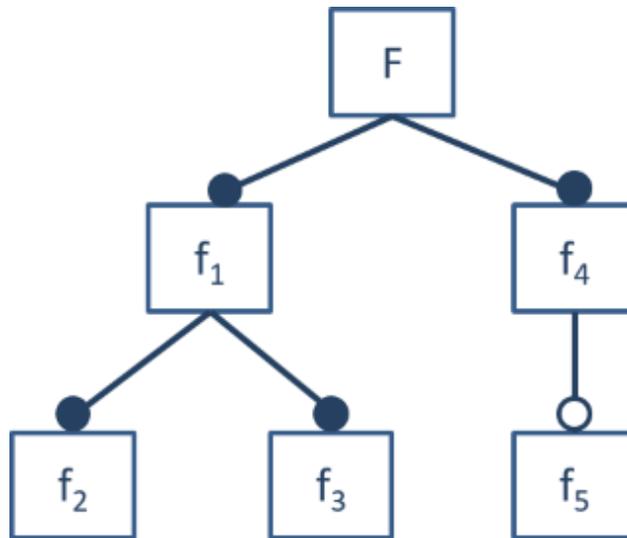


Abbildung 16: Beispiel für ein einfaches Merkmalsmodell

Abbildung 17 zeigt die Darstellung von Alternativ- (leerer Bogen) und Oder-Beziehungen (ausgefüllter Bogen) in Merkmalsmodellen. Auf der rechten Seite sind alle möglichen Produktkonfigurationen dargestellt, die sich aus diesem Modell ergeben.

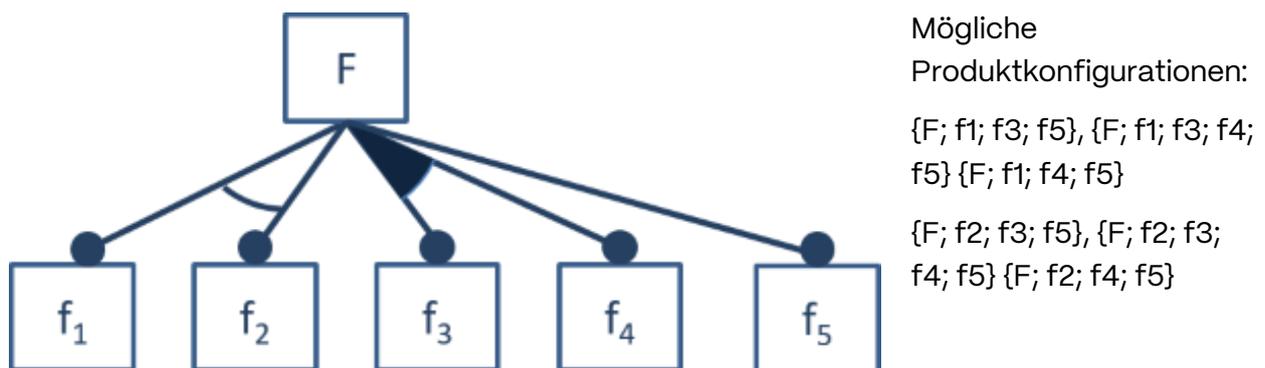


Abbildung 17: Darstellung von Alternativ- und Oder-Beziehungen (links) und daraus resultierenden möglichen Produktkonfigurationen (rechts)

In Abbildung 18 findet sich nun ein Beispiel, in dem zusätzlich Abhängigkeiten zwischen Merkmalen dargestellt sind.

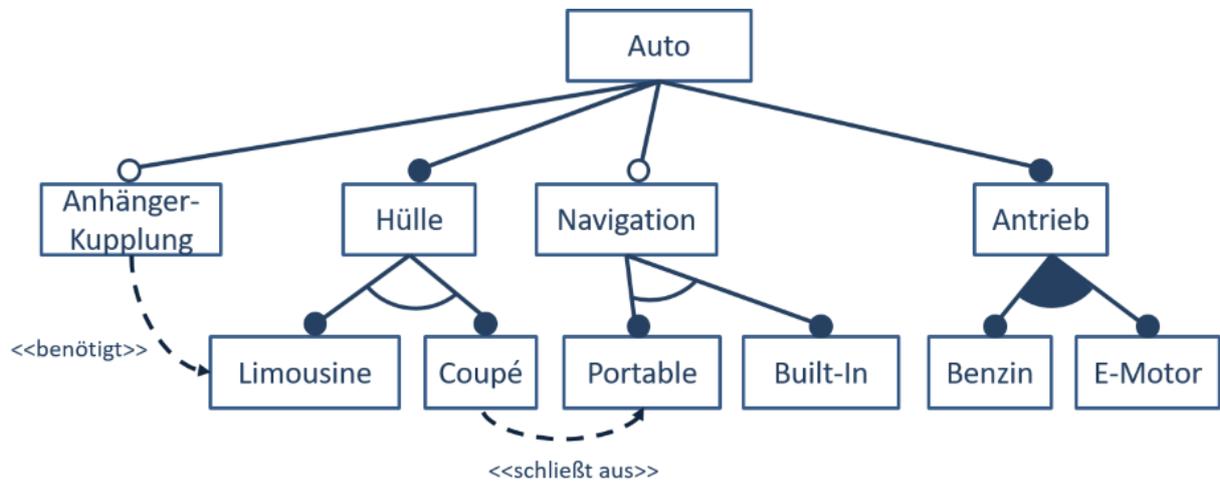


Abbildung 18: Merkmalsmodell für ein Auto mit Abhängigkeiten

In diesem Beispiel kann eine Anhängerkupplung beispielsweise nur verbaut werden, wenn es sich um eine Limousine handelt.

Ein Coupé schließt den Verbau eines tragbaren (portablen) Navigationsgerätes aus. Der Antrieb kann rein verbrennungsmotorisch (Benzin), rein elektrisch (E-Motor) oder auch aus einer Kombination (d.h. Hybrid-Antrieb) sein.

In der Merkmalsmodellierung werden Varianten durch Blatt-Elemente im Merkmalsmodell repräsentiert. Variationspunkte sind Nicht-Blatt-Elemente.

Die Gruppen-Verfeinerungs-Beziehungen „Oder“ und „Alternativ“ haben eine höhere Wertigkeit als die Einfach-Verfeinerungs-Beziehungen „Verpflichtend“ und „Optional“.

Bei der Verwendung von „Oder“ sowie „Alternative“ kann die Verwendung der Kennzeichnung „Optional“ oder „Verpflichtend“ bei den betroffenen Kind-Merkmalen darum entfallen.

In Abbildung 19 findet sich ein Beispiel für kardinalitätsbasierte Elemente. Das Produkt F muss hierbei zwei oder drei der Merkmale f1 bis f4 haben.

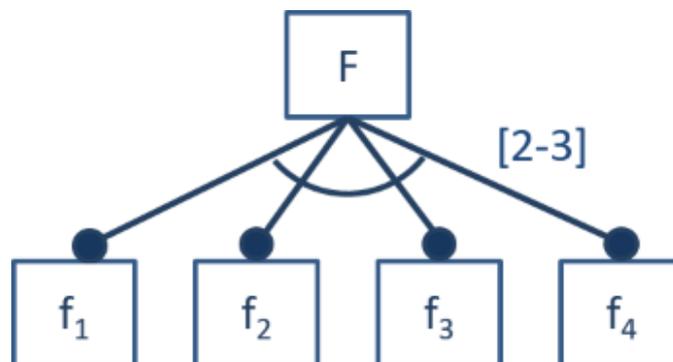


Abbildung 19: Merkmalsmodell mit kardinalitätsbasierten Elementen

Identifikation von Merkmalen

Soll ein merkmalsbasiertes Variantenmanagement eingeführt werden, so stellt sich die Frage, welche Merkmale künftig zu verwenden sind. Dabei ist es in der Regel sinnvoll, nicht völlig neue Merkmale zu erfinden, sondern bei der Definition von Merkmalen auf existierende Anforderungsdokumente zurückzugreifen [BoHo2011].

Ein guter Ausgangspunkt bei der Identifikation von Merkmalen ist es, die Substantive eines Anforderungstextes als potentielle Merkmale zu betrachten. Wichtig ist hierbei, allgemeine Substantive, die mit dem Produkt als solches nichts zu tun haben (z.B. in Abschnitten, die sich mit vertraglichen Aspekten oder dem Entwicklungsprozess beschäftigen; solche Substantive werden auch als Stopp-Wörter bezeichnet) zu ignorieren. Basierend auf solchen Substantiv-Listen kann ein Experte dann in der Regel schnell potentielle Merkmale identifizieren. Ein Nachteil dieser Vorgehensweise ist allerdings, dass insbesondere Variationspunkte, die textuell nicht explizit erwähnt sind, nicht identifiziert werden können.

Variationspunkte können oft identifiziert werden, wenn der Anforderungsanalytiker bei unterschiedlichen Varianten beharrlich nach dem „Warum“ fragt.

Werkzeugunterstützung

Soll Variabilität explizit dokumentiert werden, ist dies ohne den Einsatz von speziellen Werkzeugen in der Regel nicht möglich. Auf dem Markt sind Werkzeugen verfügbar, die es erlauben

- Merkmalsmodelle zu erstellen,
- Produktkonfigurationen zu bilden,
- Produktkonfigurationen auf ihre Zulässigkeit zu prüfen.

Typischerweise haben solche Werkzeuge ausgeprägte Schnittstellen zu anderen Werkzeugen, in denen die eigentlichen Entwicklungs-Artefakte (z.B. Anforderungen oder Testfälle) liegen und die es so ermöglichen, die Entwicklungs-Artefakte mit den Merkmalen in Beziehung zu setzen.

8 Berichtswesen im Requirements Management (K3)

Dauer: 1,75 Stunden

Begriffe: Berichtswesen, Bericht, Kennzahl, Goal-Question-Metric-Methode

Lernziele

- LZ 8.1 Ziele und Nutzen des Berichtswesens im RM kennen (K1)
- LZ 8.2 Schnittstellen, Inhalte und den Definitionsprozess eines Berichts kennen (K2)
- LZ 8.3.1 Typische Kennzahlen im Requirements Management kennen (K1)
- LZ 8.3.2 Requirements Management-Kennzahlen mittels der Goal-Question-Metric Methode ableiten können (K3)
- LZ 8.4 Risiken und Probleme bei der Anwendung des Berichtswesens kennen (K2)

8.1 Ziele und Nutzen des Berichtswesens im Requirements Management (K1)

Berichte sind Teil des Projekt- und Unternehmenscontrollings. Sie dienen dazu, Informationen über Projekte oder über Organisationseinheiten zusammengetragen und für bestimmte Zielgruppen passend aufzubereiten, um deren Informationsbedarf zu decken.

[ZIEG1998] definiert Berichtswesen als „die Erstellung und Weiterleitung von funktionsübergreifenden Berichten im Sinne von geordneter Zusammenstellung von Nachrichten ausschließlich für das Management“. Eine andere Definition stellt die Erstellung und Zielsetzung des Berichtswesens besonders heraus: „Man kann unter ihm alle Personen, Einrichtungen, Regelungen, Daten und Prozesse verstehen, mit denen Berichte erstellt und weitergegeben werden. Dabei stellen Berichte unter einer übergeordneten Zielsetzung, einem Unterrichtungszweck, zusammengefasste Informationen dar.“ [KÜPP2005].

Berichtswesen im Requirements Management (RM) ist also das Sammeln, Auswerten und Darstellen von Informationen über Anforderungen oder den Anforderungsmanagement-Prozess und die Bereitstellung dieser Informationen.

Ein Bericht ist ein Dokument, das eine oder mehrere Sichten für bestimmte Stakeholder und einen bestimmten Zweck zusammenfasst.

Die nun folgenden Ausführungen zum Thema Berichtswesen im Requirements Management fokussieren primär auf die Entwicklung von neuen Systemen. Die Vorgehen bzw. Prinzipien sind jedoch in anderen Kontexten wie einer kontinuierlichen Weiterentwicklung von Systemen sowie auch im Rahmen einer projektübergreifenden Verwaltung von Anforderungen grundsätzlich auch anwendbar und zielführend.

8.2 Etablierung eines Berichtswesens im Requirements Management (K2)

8.2.1 Schnittstellen (K2)

Requirements Management ist eng verzahnt mit dem Projektmanagement, Produktmanagement und dem Qualitätsmanagement. Somit bestehen diese Schnittstellen auch innerhalb des Berichtswesens. Deshalb macht es Sinn, das Berichtswesen dieser drei Bereiche und deren Daten zu koordinieren. Projektmanagement und Qualitätsmanagement werden die wichtigsten Berichtsempfänger des Berichtswesens über das Requirements Management sein.

Oft werden auch Berichte aus dem Requirements Management und dem Qualitätsmanagement übergreifend erstellt.

8.2.2 Inhalte eines Berichts (K1)

Berichte können prinzipiell in beliebiger Form erstellt werden (z.B. formlos in einem E-Mail-Text). Oft gibt es jedoch Vorlagen, so dass jeder Bericht dieselbe Struktur aufweist. Das macht die Berichte einfach und effizient zu lesen und zu erstellen. Dieselbe Information steht immer an derselben Stelle im Bericht. Besonders praktisch ist es für den Autor, wenn der Bericht automatisch aus dem Werkzeug heraus erzeugt werden kann, in dem ohnehin die nötigen Informationen verwaltet werden.

Die üblichen Inhalte von Berichten sind die folgenden:

- **Projektname:** Der Bericht muss konkret angeben, auf welches Projekt er sich bezieht. (Handelt es sich um den Bericht einer Organisationseinheit, z.B. Abteilung, dann steht hier statt einem Projektnamen der Name der Abteilung.)
- **Datum der Berichtserstellung:** Die im Bericht dargestellten Inhalte ändern sich täglich oder sogar stündlich. Darum ist es wichtig anzugeben, zu welchem Zeitpunkt die Daten extrahiert wurden, d.h. auf welchem Informationsstand der Bericht beruht.
- **Versionsnummer:** Gibt es mehrere Versionen des Berichts, etwa weil jemand noch eine Ergänzung gemacht hatte, so muss die neue Version eine neue Versionsnummer tragen für die Eindeutigkeit des Berichts und zur besseren Verfolgbarkeit von Änderungen.
- **Berichtszeitraum:** Berichte können sich auf Tage, Wochen, Monate, Jahre oder jedes andere Zeitintervall beziehen. Wöchentliche und monatliche Berichte sind am häufigsten, aber in kritischen Projektphasen können Berichte auch täglich oder halbtäglich erstellt werden. Natürlich macht es bei der Interpretation der Berichtsinhalte einen Unterschied, ob sie sich auf das beziehen, was innerhalb von einer Woche oder innerhalb eines Monats geleistet wurde.
- **Ersteller und Empfänger:** Ein Bericht hat einen Ersteller (Autor) und Empfänger (Verteiler). Unter den Empfängern kann man auch noch unterscheiden zwischen den Personen, die ihn nur zur Information erhalten, und denen, die ihn genehmigen

müssen. Die Namen dieser Personen sind üblicherweise auf dem Bericht genannt und somit dokumentiert.

- **Freigabestatus:** Benötigt der Bericht eine Freigabe, so ist dieser Status hier zu vermerken. Es kann sein, dass der Bericht in verschiedenen Freigabestatus unterschiedliche Inhalte enthält.
- **Gesamtstatus:** Gleich am Anfang des Berichts möchte – insbesondere der eilige Leser – einen Überblick erhalten, wie kritisch das Projekt ist. Gerade viel beschäftigte Manager lesen den Bericht nur, wenn das Projekt kritisch ist. Berichte über Projekte, die nach Plan verlaufen, enthalten für den Vorgesetzten keinerlei Informationswert, da seine Unterstützung nicht benötigt wird. Beliebt sind hierbei Ampelskalen mit den bekannten Farben grün, gelb und rot.
- **Fachlicher Inhalt:** Dies ist der eigentliche Kern des Berichts.

Der fachliche Inhalt eines Requirements Management Berichts besteht aus einer oder mehrerer der folgenden Inhaltsklassen:

- Selektive oder verdichtende **Sichten auf Anforderungen** (vgl. 3.5). Diese können in der Regel automatisch durch ein Requirements Management Werkzeug generiert werden.
- **Kennzahlen** über Anforderungen oder den RE-Prozess. Diese Kennzahlen werden in der Regel mittels automatisierter Auswertungen gewonnen. Diese Auswertung kann in einem Requirements Management Werkzeug oder außerhalb erfolgen.
- Deskriptive oder wertende **Beschreibungen**.

8.2.3 Tipps für die Entwicklung und Anwendung des Berichtswesens (K2)

In der Entwicklung und Anwendung des (anforderungsbasierten) Berichtswesens gibt es einige praktische Tipps oder Hinweise, die zu beachten sind:

- **Fokussierung auf das Wesentliche:** Auch wenn die Stakeholder und der Nutzen des Berichtswesens bekannt sind, so besteht die Kunst doch darin, sich auf die wesentlichen Inhalte zu konzentrieren. Dabei hilft die GQM-Methode, die wir in Kapitel 8.3.2 beschreiben.
- **Abstimmung:** Die für den Bericht nötigen Informationen müssen im Requirements Informationsmodell und Attributierungsschema vorgesehen sein. Da das nachträgliche Ändern von Informationsmodell und Attributierungsschema schwierig ist und das Einführen eines neuen Attributes eine aufwändige Nachpflege der Inhalte verlangt, sollten frühzeitig, noch vor einem potentiellen Entwicklungsprojekt, die Datenmodelle des RM geklärt werden. Hilfreich ist es dabei, Referenzmodelle zu verwenden, die bereits zuvor miteinander abgestimmt wurden.
- **Datenerfassung:** Diejenigen, welche die Daten zu erfassen haben, sind nicht dieselben, welche den Informationsbedarf haben und den Bericht erstellen oder lesen. Die Datenerfasser haben also keine intrinsische Motivation dafür, die Daten einzupflegen. Umso wichtiger ist es, dass die Datenerfassung in ihre täglichen Arbeitsprozesse gut integriert ist und klar ist, wer wann welche Daten einzutragen hat.

- **Datenqualität:** Das reine Vorhandensein der Attribute führt noch nicht selbstverständlich dazu, dass auch alle Inhalte gepflegt, aktuell und richtig sind. Während es im Sinne eines effizienten Arbeitsprozesses nicht sinnvoll ist, zu viele Pflichtfelder einzuführen, zumal manche Informationen beim Anlegen einer Anforderung noch nicht vorhanden sind, wäre es für das Berichtswesen wichtig, dass die Attribute gepflegt sind. Fehlende Inhalte führen zu unvollständigen Aussagen der Berichte.

8.2.4 Berichtsdefinitionsprozess (K2)

Für die Definition der anforderungsbasierten Berichte und die Abstimmung mit dem Requirements Informationsmodell ist der Requirements Manager verantwortlich oder er delegiert diese Aufgabe an eine geeignete Person.

Laut ISO 15288 ([ISO 15288], 6.3.7.3 a) 1) bis 4)) definiert man ein Mess- bzw. Berichtswesen entlang dieser Schritte:

1. Die Eigenschaften der Organisation beschreiben, die für die Messung relevant sind.
2. Informationsbedürfnisse identifizieren und priorisieren.
3. Kennzahlen auswählen und dokumentieren, die diese Informationsbedürfnisse erfüllen.
4. Vorgehensweisen für die Datensammlung, -analyse und das Berichtswesen definieren.

8.3 Kennzahlen im Requirements Engineering (K3)

Kennzahlen oder Maßzahlen (oft auch als fälschlicherweise als Metriken bezeichnet) sind ein wichtiger Bestandteil von Berichten. [EBER2012], Seite 436 definiert ein Maß als:

„(1) Eine formale, präzise, reproduzierbare, objektive Zuordnung einer Zahl oder eines Symbols zu einem Objekt, um eine spezifische Eigenschaft zu charakterisieren.

(2) Mathematisch: Abbildung M eines empirischen Systems C und seiner Relationen R in ein numerisches System M .

(3) Die Nutzung (Erhebung, Analyse, Bewertung) eines Maßes. Beispiele: Maß für ein Produkt (z.B. Fehler, Dauer, Planabweichung) oder einen Prozess (z.B. Fehlerkosten, Effizienz, Effektivität).“

8.3.1 Kennzahlen im Requirements Management (K1)

Man kann Kennzahlen im Requirements Management in zwei große Klassen einteilen:

- **Produkt-Kennzahlen** (d.h. Kennzahlen über Anforderungen):
Hierzu gehören Kennzahlen über die Anzahl an Anforderungen sowie die Eigenschaften von Anforderungen.

Beispiele: Anzahl sicherheitskritischer Anforderungen, durchschnittliche Länge einer Anforderungen, Anzahl fehlerhafter Anforderungen

- **Prozess-Kennzahlen** (d.h. Kennzahlen über den RE-Prozess):

Beispiele: Anforderungsänderungen pro Monat, durchschnittlicher Aufwand zur Prüfung einer Anforderung, durchschnittliche Häufigkeit der Änderungen einer Anforderung in den letzten n Monaten

8.3.2 Ableiten von Kennzahlen mittels Goal-Question-Metric-Methode (K3)

Bei der Berichtserstellung ist die „Goal-Question-Metric“ (GQM) [BaCR1984] eine mögliche Methode um sicherzustellen, dass nur zielführende Kennzahlen für Berichte bzw. Berichtsinhalte definiert werden. Die GQM ist eine systematische Vorgehensweise zur Identifikation solcher Kennzahlen. Hierbei wird eine geeignete Kennzahl durch das Beantworten der folgenden Fragen identifiziert:

- Welches Ziel soll durch die Messung erreicht werden? (*Goal*)
- Was soll gemessen werden bzw. welche Fragen soll die Messung beantworten? (*Question*)
- Welche Kennzahl(en) sind in der Lage, die notwendigen Eigenschaften zu beschreiben? (*Metric*)

8.4 Risiken und Probleme im Berichtswesen (K2)

In der Praxis treten praktische Schwierigkeiten bei der Erhebung und Auswertung der Daten auf, die dazu führen, dass Berichte die Realität nicht adäquat abbilden. Da aufgrund der Berichte wichtige Managemententscheidungen getroffen werden sollen, kann ein lückenhafter oder sogar absichtlich geschönter Bericht weitreichende Folgen nach sich ziehen.

Auswertung der Daten: komprimiertes Abbild der Realität

Ein Bericht stellt immer ein stark komprimiertes Modell der Realität dar, in dem Gleichartiges zu Kategorien zusammengefasst wird und unwesentliche Details weggelassen werden. Es ist sehr schwer, dies so zu tun, dass jederzeit jede zukünftig auftretende Frage gut beantwortet werden kann.

Darum muss man sich der Oberflächlichkeit eines Berichts immer bewusst sein. Insbesondere sollte man vermeiden, falsche Schlüsse aus den vorhandenen Daten zu ziehen. So erlaubt z. B. ein Bericht, der eine 99%ige Verfolgbarkeit aller Anforderungen ausweist, noch keine Aussage über den Fortschritt des Projektes oder die Qualität der Verlinkungen. Gerade die noch nicht verlinkten Anforderungen könnten die wichtigsten bzw. aufwandintensivsten Anforderungen sein, die maßgeblich zum Projekterfolg beitragen.

Dieser Problematik sollte man sich bei einer Reduktion der Komplexität auf Kennzahlen immer bewusst sein. Oft sind nur sehr grobe Aussagen und Schlussfolgerungen möglich.

Schlechte Datenqualität

Fehlende Daten sind meist leicht zu entdecken. Weniger leicht ist es, die Qualität der Daten zu beurteilen: Stimmen die Daten mit der Realität überein? Sind sie aktuell? Messen sie genau das, was sie sollen oder wurde z.B. im Attribut „Aufwand“ nur der Implementierungsaufwand erhoben, obwohl auch der Testaufwand mit berücksichtigt werden soll? Ist die Kritikalität tatsächlich das Ergebnis einer Expertenbefragung oder vorläufig gesetzt worden?

Unentdeckte, aber auch bekannte Mängel in der Datenqualität führen dazu, dass der Bericht die Wirklichkeit in Neu- und Weiterentwicklungen nicht richtig wiedergibt. Aufgrund dieser falschen Daten können nur schwer richtige Managemententscheidungen getroffen werden. Und selbst wenn die mangelnde Datenqualität bekannt ist, sind Entscheidungen schwer zu treffen.

Eine schlechte Datenqualität entsteht oft dadurch, dass die Beteiligten die Datenpflege unwichtig nehmen, weil sie selbst wenig Nutzen davon haben. Manchmal haben sie im Gegenteil sogar ein Interesse daran, die Daten zu schönen, oder zumindest ein Interesse daran, Zeit bei der Datenpflege zu sparen, indem sie keine sorgfältigen Analysen durchführen, sondern hastig plausibel erscheinende Daten eingeben.

Schlechte Datenqualität kann aber auch dadurch entstehen, dass nicht alle Beteiligten dieselbe Vorstellung haben. In der agilen Entwicklung (vgl. 10) ist die „*definition of done*“ ein wichtiges Diskussionsthema. Es muss eindeutig definiert sein, wann eine Anforderung als erledigt gewertet ist. Mögliche Kriterien dafür, dass eine Anforderung umgesetzt ist, sind z.B.: Der Code ist erstellt, Unit Tests sind erstellt und erfolgreich durchgelaufen, die Dokumentation ist angepasst und die Codekonvention ist eingehalten.

Datenschutzbestimmungen

Geltende Datenschutzbestimmungen grundsätzlicher wie auch unternehmensspezifischer Art müssen bei der Definition und der Umsetzung des Berichtswesens eingehalten werden. Werden personenbezogene Daten von Beteiligten eingegeben und ohne deren Wissen im Unternehmen in Form von Berichten weiterkommuniziert, kann dies zu Problemen führen. In diesem Zusammenhang ist es wichtig, mit den Datenautoren klar zu vereinbaren, wer welche Daten im Rahmen von welchen zu tätigen Entscheidungen erhält. Grundsätzlich sollten personenbezogene und personenbeziehbare Daten sparsam verwendet werden, also schon gar nicht erst eingegeben werden. Auch bei der Definition von Sichten sollte darauf geachtet werden, dass keine Aussagen über einzelne Personen getroffen werden können, um nicht unbeabsichtigt Regeln des Datenschutzes zu verletzen.

Inflationäres Berichtswesen

Steigt die Menge an Berichtsinformationen stetig an, könnte dies auch dazu führen, dass die Berichtsempfänger diese Daten aus zeitlichen Gründen nicht mehr verarbeiten können und dadurch wichtige Entscheidungen nicht mehr fundiert getroffen werden.

Weniger ist darum mehr! Eine Fokussierung auf die wirklich nötigen Informationen ist anzustreben. Dies kann auch bedeuten, dass verschiedene Zielgruppen unterschiedliche Berichte erhalten, in denen nur bestimmte Aspekte dargestellt werden oder in verschiedener Detailtiefe dargestellt sind.

9 Management von Requirements Engineering Prozessen (K3)

Dauer: 2,5 Stunden

Begriffe: iteratives RE, upfront, leichtgewichtige Anforderungen, Requirements Engineering Prozess (RE-Prozess), PDCA (Plan Do Check Act)-Zyklus, Kontinuierliche Prozessverbesserung (KPV)

Lernziele

- LZ 9.1 Requirements Engineering als Prozess verstehen (K2)
- LZ 9.2.1 Wählbare Parameter des RE-Prozesses kennen (K1)
- LZ 9.2.2 Eignung eines RE-Prozesses im Hinblick auf die Prozessparameter beurteilen können (K3)
- LZ 9.3 Verschiedene Methoden für das Dokumentieren des RE-Prozesses kennen und anwenden können (K3)
- LZ 9.4 Notwendigkeit zur Überwachung und Steuerung des RE-Prozesses kennen (K1)
- LZ 9.5.1 Methoden zur Verbesserung des RE-Prozesses kennen und anwenden können (K3)
- LZ 9.5.2 Notwendigkeit eines Requirements Management Plans RMP kennen (K2)

9.1 Requirements Engineering als Prozess (K2)

Ein Prozess besteht aus voneinander abhängigen Aktivitäten, die ausgeführt werden, um ein bestimmtes Ziel zu erreichen. Bei jeder Aktivität werden jeweils Eingaben (Information, Material, Energie, Ressourcen) in Ausgaben (Ergebnisse) transformiert [ISO 9000]. Jede Aktivität ist eindeutig einer verantwortlichen Organisationseinheit, z.B. einer Rolle, zugeordnet. Somit stellt auch das Requirements Engineering und Requirements Management einen Prozess dar.

Unter dem RE-Prozess versteht man einen Prozess zur Erhebung der Anforderungen sowie deren Verwaltung: „A systematic process of developing requirements through an iterative co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained.“ [LoKa1995], Abschnitt 13.

Zu diesem RE-Prozess gehören die folgenden Haupttätigkeiten [IREB2020], Kapitel 1]:

- Ermittlung von Anforderungen
- Anforderungen dokumentieren,
- Anforderungen prüfen und abstimmen,
- Requirements Management

In jedem konkreten Projekt gibt es mehrere Ermittlungs-Aktivitäten wie beispielsweise Workshops und Meetings mit Stakeholdern, Dokumentenanalyse und so weiter. Gleiches gilt auch für die anderen Haupttätigkeiten.

Der RE-Prozess hat als Eingangsinformationen die Bedürfnisse und Ideen der Stakeholder. Außerdem spielen auch der Status quo vor dem Projektstart (z.B. das Altsystem) und Konkurrenzprodukte eine Rolle. Das Ergebnis des RE-Prozesses ist eine validierte,

konfliktfreie, konsistente, priorisierte, qualitätsgesicherte Anforderungsspezifikation, die als verlässliche Grundlage für die weiteren Projektarbeiten dienen kann.

Im Allgemeinen haben die vier Haupttätigkeiten die folgenden Eingangsinformationen und Ergebnisse, die natürlich verschieden aussehen können, insbesondere wenn firmenspezifische Vorgaben oder Standards eingehalten werden müssen (siehe Tabelle 2):

Haupttätigkeit	Eingabe	Ergebnis
Anforderungen ermitteln	Stakeholder und deren Bedürfnisse und Ideen; gegebenenfalls: ein vorhandenes Altsystem und dessen Dokumentation; Konkurrenzprodukte	Mündliche und schriftliche Anforderungen einschließlich der System-Vision
Anforderungen dokumentieren,	Mündliche und schriftliche Anforderungen	Schriftliche Anforderungs-Spezifikation – textuell oder modellbasiert oder beides
Anforderungvalidierung und -abstimmung	Schriftliche Anforderungs-Spezifikation	Validierte, konfliktfreie, konsistente, priorisierte, qualitätsgesicherte Anforderungsspezifikation
Requirements Management	Schriftliche Anforderungsspezifikation sowie Änderungsanträge	Stets aktuelle, validierte, konfliktfreie, konsistente, priorisierte, qualitätsgesicherte Anforderungsspezifikation; Aufbereitung der Anforderungen für einzelne Stakeholder-Gruppen

Tabelle 2: Vier Haupttätigkeiten des Requirements Engineerings sowie deren Eingaben und Ergebnisse

Diese Haupttätigkeiten sind immer durchzuführen, egal ob ausdrücklich dokumentiert oder implizit. Verschiedene Standards verlangen eine unterschiedliche Durchführung dieser Aktivitäten und machen unterschiedliche Vorgaben hinsichtlich der Artefakte (vgl. auch 1.4).

Die Ergebnisse des RE-Prozesses müssen Qualitätskriterien in drei voneinander unabhängigen Dimensionen erfüllen: in der Dimension der Spezifikation, Darstellung und Zustimmung (engl. specification, representation and agreement dimension) [POHL1994]. Die Anforderungen sollen im Zeitverlauf innerhalb dieser Dimensionen reifer werden.

- **Spezifikation:** Diese Dimension beschreibt die Vollständigkeit der Spezifikation. Zu Anfang des RE-Prozesses sind die Anforderungen vage und unklar (engl. opaque).

Mit Fortschreiten des Prozesses werden die Anforderungen vollständiger im Sinne von einer vollständigen Abdeckung des zu lösenden Problems sowie einer Beschreibung, die detailliert genug ist, um richtig verstanden zu werden. Verschiedene Standards geben Richtlinien dafür vor, welche Anforderungen die Anforderungen erfüllen müssen, damit sie als vollständig gelten können. Jedoch ist es nicht möglich, die Vollständigkeit von Anforderungen zu beweisen.

- **Darstellung:** Hier variiert die Skala von informell bis formal. Als informelle Darstellung gelten Skizzen, Freitext und Prototypen. Als semi-formal gelten graphisch dargestellte Modelle, beispielsweise Klassendiagramme, Zustandsmaschinen, Anwendungsfalldiagramme oder Datenflussdiagramme. Auch tabellarisch dargestellte Anwendungsfälle, welche strikt einer vorgegebenen syntaktischen Struktur folgen, sind semi-formal. Formale Spezifikationen beschreiben Anforderungen mit Hilfe von Logiksprachen und formaler Semantik völlig eindeutig. Auch die Erstellung einer formalen Spezifikation beginnt üblicherweise mit informellen Darstellungsformen.
- **Zustimmung:** Während des RE-Prozesses ist das Herstellen von Zustimmung ein weiteres Ziel. Man bewegt sich hierbei auf der Zustimmungs-Dimension von der persönlichen Sicht zur gemeinsamen Sicht auf die Anforderungen.

Die Anforderungsspezifikation soll auf allen drei Dimensionen optimiert werden. Dabei tragen die Aktivitäten der Ermittlung vor allem zur Verbesserung in Richtung der Spezifikations-Dimension bei, das Dokumentieren zur Darstellung und die Prüfung und Abstimmung zur Verbesserung in der Zustimmungs-Dimension. Die Anforderungsverwaltung zielt darauf ab, das Qualitätsniveau in allen drei Dimensionen zu erhalten.

9.2 Parameter des Requirements Engineering Prozesses (K3)

Der RE-Prozess kann ganz verschieden gestaltet werden und muss sich insbesondere den gegebenen Randbedingungen anpassen. Bei aller Vielfalt existierender RE-Prozesse in verschiedenen Vorgehensmodellen gibt es nur eine übersichtliche Anzahl an Prozess-Parametern, die man bei der Wahl bzw. Anpassung des RE-Prozesses variieren kann:

- Zeitverlauf der Ermittlung,
- Detaillierungstiefe der Dokumentation, d.h. schwergewichtige versus leichtgewichtige Spezifikation,
- Eingliederung von Änderungen, konkret: Änderungsantrag versus Product Backlog,
- Verteilung der Verantwortlichkeit.

Diese Parameter sollten Sie passend zu den Randbedingungen wählen. Solche Randbedingungen sind:

- Größe des Projektes.
- Handelt es sich um eine Neuimplementierung oder eine geringfügige Weiterentwicklung, Verbesserung oder Variation eines bestehenden, reifen Systems oder Produkts?
- Wurde ein Festpreis vereinbart oder nicht?
- Steht ein seit Jahren zusammenarbeitendes stabiles Team zur Verfügung?

- Verfügbarkeit von Personen und deren Qualifikation.

Zeitverlauf der Ermittlung (upfront oder iterativ)

Die Anforderungen können entweder komplett am Projektanfang (engl. upfront) erhoben werden oder iterativ (iteratives Requirements Engineering): Im ersteren Fall (Upfront) erstellt man zu Projektbeginn eine Anforderungsspezifikation (z.B. ein Lastenheft), die den kompletten geplanten Projektumfang vollständig beschreibt. Beim iterativen Requirements Engineering verfolgt man nicht den Anspruch, die Anforderungen oder auch nur den Projektumfang (englisch: Project Scope) anfangs komplett festzulegen, sondern sieht die Anforderungsdokumentation (z.B. das Product Backlog) als eine vorläufige Liste. Anforderungen können jederzeit dazu kommen oder sich ändern, auch während der Implementierung.

Hinweis: Es besteht ein Unterschied zwischen iterativem Requirements Engineering und iterativer Entwicklung. So ist es denkbar, zunächst upfront eine vollständige Anforderungsspezifikation zu erstellen und später die Anforderungen in iterativer Entwicklung umzusetzen.

Handelt sich bei dem Projekt um die geringfügige Weiterentwicklung, Verbesserung oder Variation eines bestehenden, reifen Systems oder Produkts, dann ist zu erwarten, dass die Anforderungen für das Gesamtprojekt stabil definiert werden können und wenig Überraschungen zu erwarten sind. Hier ist die upfront Anforderungsermittlung möglich und sinnvoll.

Handelt es sich jedoch um ein sehr innovatives Projekt mit vielen Unwägbarkeiten, um ein volatiles Umfeld, unentschlossene oder einander widersprechende Stakeholder oder führen andere Risikofaktoren dazu, dass eine verlässliche Upfront-Spezifikation nicht möglich ist, dient das iterative Requirements Engineering zur Risikoverringerung.

Detaillierungstiefe der Anforderungs-Dokumentation

Die Detaillierungstiefe der Dokumentation bzw. Spezifikation kann variieren zwischen schwergewichtigen und leichtgewichtigen Anforderungen: Die schwergewichtige Spezifikation beschreibt alle Anforderungen im Detail, einschließlich aller ihrer Attribute und Verfolgbarkeitsbeziehungen, wodurch die Spezifikation sehr umfangreich wird.

Die leichtgewichtigen Spezifikation beschreibt Anforderungen nur so umfangreich wie nötig und nicht früher als nötig. Wann bestimmte Informationen nötig sind, hängt vom Vorgehensmodell ab. Was benötigt wird, hängt von den Stakeholdern, deren Bedürfnissen und Vorkenntnissen ab. Eine projektspezifische Stakeholder-Analyse unterstützt dabei, zu definieren, wie detailliert die Requirements-Spezifikation sein muss.

So erfüllt die Spezifikation u.a. den Zweck, dass der Entwickler versteht, was die Stakeholder wünschen. Die Details der Implementierung werden bei der leichtgewichtigen Spezifikation entweder dem Entwickler überlassen (insbesondere wenn er sich mit der Fachdomäne sehr gut auskennt), mündlich diskutiert ohne sie zu dokumentieren oder anhand eines Prototypen verfeinert.

Die leichtgewichtige Anforderungsspezifikation beschreibt Anforderungen beispielsweise als User Stories. Anforderungen werden erst dann im Detail spezifiziert, wenn ihre Implementierung unmittelbar bevor steht.

Auch wenn üblicherweise upfront schwergewichtig spezifiziert wird (z.B. im Wasserfallmodell und V-Modell XT) und iterativ leichtgewichtig (wie in Scrum und anderen agilen Methoden, vgl. Lerneinheit 10), so sind die beiden Parameter Zeitverlauf und Detaillierungstiefe doch unabhängig voneinander. Man könnte sowohl upfront eine leichtgewichtige Spezifikation erstellen als auch iterativ eine schwergewichtige (wie im Rational Unified Process).

Änderungsmanagement: Eingliederung von Änderungen (Änderungsantrag versus Product Backlog)

Anforderungen ändern sich während des Projektes. Manche RE-Prozesse integrieren neue oder geänderte Anforderungen als Änderungsanträge in die Anforderungsspezifikation und den Entwicklungsprozess. Dabei handelt es sich meist um Projekte mit Festpreis und Upfront-Anforderungsspezifikation, d.h. organisatorisch und juristisch ist die Anforderungsermittlung zu einem bestimmten Zeitpunkt abgeschlossen. Spätere Änderungen sind rechtlich gesehen eine Auftragsänderung. Der Änderungsantrag (engl. change request) bedeutet rechtlich eine Neubeauftragung. Üblicherweise wird bereits im Vertrag festgelegt, wie Änderungsanträge zu handhaben sind. Sie durchlaufen üblicherweise ein vereinfachtes Genehmigungsverfahren mit den Schritten Analyse (der Anforderungen und deren Nutzen), Auswirkungsanalyse / Impact Analyse (d.h. Analyse der Änderungen am System, deren Kosten und Risiken), Entscheidung durch das Change Control Board und dann die Umsetzung. Beschrieben wird der Änderungsantrag oft in einer Änderungsantrag-Vorlage, die ihm eine eindeutige Nummer und Namen zuweist, das zu lösende Problem und die vorgeschlagene Lösung beschreibt, Kosten, Nutzen und Risiken quantifiziert und den Status verwaltet (beantragt, angenommen, abgelehnt, verschoben, umgesetzt).

Im iterativen Requirements Engineering werden dagegen Anforderungen im Product Backlog gesammelt und alle Anforderungen – die älteren und die neuen – gleichwertig behandelt. Dies wird dadurch ermöglicht, dass man sich nie auf einen definierten Systemumfang festlegt. Trotzdem ist es nicht zwingend nötig, dass durch eine Upfront-Anforderungs-Spezifikation spätere Anforderungen als Änderungsanträge zu behandeln sind.

Es wäre denkbar, auch das upfront erstellte Anforderungsartefakt später anzupassen, ohne Änderungen als Änderungsantrag zu erfassen und zu genehmigen. Änderungen an den Anforderungs-Artefakten müssen natürlich dokumentiert werden und nachvollziehbar sein.

Verteilung der Verantwortlichkeit

Für den RE-Prozess kann eine einzelne Rolle (z.B. der „Requirements Manager“) verantwortlich sein in dem Sinne, dass sie den RE-Prozess plant, steuert und verbessert.

Eventuell führt sie auch selbst alle Aktivitäten des RE-Prozesses durch. Es kann aber auch ein ganzes Team oder mehrere Rollen geben, die für das Requirements Engineering verantwortlich sind, entweder für verschiedene Aktivitäten oder verschiedene Inhalte (z.B. funktionale Anforderungen versus Usability- oder Benutzbarkeitsanforderungen).

Das Requirements Engineering kann aber auch in den Entwicklungsprozess eng integriert sein, ohne dass ein separater RE-Prozess oder die Rolle des Anforderungsanalytikers existiert. Dann führt das Entwicklungsteam die Requirements Engineering Aktivitäten durch, d.h. die Teammitglieder erheben, dokumentieren, prüfen und verwalten Anforderungen.

9.3 Den Requirements Engineering Prozess dokumentieren (K3)

Der RE-Prozess besteht aus zahlreichen Aktivitäten der oben genannten vier Typen, beispielsweise Ermittlungsworkshops, Spezifikations-Reviews, etc., wie im Foundation Level Lehrplan erklärt [IREB2020]. Geplant werden viele dieser Aktivitäten in Form von Meetings, da an vielen Aktivitäten mehrere Personen beteiligt sind. Die Reihenfolge dieser Aktivitäten ergibt sich aus der Wahl der Prozessparameter (siehe 9.2), die projektspezifisch oder auch firmenweit festgelegt wird. Die Aktivitäten und deren Reihenfolge kann man als UML Aktivitätsdiagramm darstellen. Im Aktivitätsdiagramm kann man auch die Zuordnung der Aktivitäten zu Rollen darstellen.

Die Zuordnung der Verantwortlichkeiten für die Aktivitäten zu Rollen kann auch differenzierter mit einer RACI-Matrix wie der folgenden dargestellt werden. RACI steht für:

- **R** = *responsible* = verantwortlich für die Durchführung
- **A** = *accountable* = rechenschaftspflichtig, d.h. man genehmigt beispielsweise die Aktivität und deren Budget
- **C** = *consulted* = (wird) konsultiert, insbesondere im Sinne von fachlicher, inhaltlicher Verantwortung
- **I** = *informed* = zu informieren, d.h. die Person ist zu informieren

Tabelle 3 zeigt ein Beispiel eines Auszugs aus einer RACI-Matrix.

Aktivität	Requirements Engineer	Projektleiter	Key User
Dokumenten-Analyse: Handbuch des Altsystems	R, A	I	
Kreativitätsworkshop mit Key Usern	R	A	C

...

Tabelle 3: Beispiel einer RACI-Matrix für das Requirements Engineering.

Will man Termine und Budgets quantitativ verwalten, lässt sich der RE-Prozess auch als Projektplan darstellen.

Weitere Dokumente, die den RE-Prozess darstellen und unterstützen können, sind: Projektplan, Checklisten, Vorlagen, Beispieldokumente und Richtlinien.

Sind viele Personen am RE-Prozess beteiligt, dann ist es ebenfalls sinnvoll, diesen Prozess mit einem Werkzeug zu unterstützen. Geeignet sind hierbei alle Workflow Management Systeme im weitesten Sinne.

9.4 Den Requirements Engineering Prozess überwachen und steuern (K1)

Den RE-Prozess zu überwachen bedeutet sicherzustellen, dass alle Aktivitäten durchgeführt werden und die definierten Ergebnisse pünktlich erbracht werden sowie dass die Aktivitäten im Budget bleiben. Hilfreich hierzu sind Berichte, welche die Termine, das verbrauchte Budget, den Status und den Fertigstellungsgrad des RE-Prozesses und dessen einzelner Aktivitäten regelmäßig erfassen und die Ist-Werte mit den Soll-Werten aus der Planung vergleichen (siehe 8).

Den RE-Prozess zu steuern bedeutet, ihn entsprechend dem Plan durchzuführen oder, wenn der Verlauf vom Plan abweicht, korrigierend einzugreifen. Zeichnet sich z.B. ab, dass Termin oder Budget nicht eingehalten werden können, so sind die Auswirkungen auf das Gesamtprojekt zu ermitteln und – wenn sinnvoll – Gegenmaßnahmen zu ergreifen. Um den Verlauf an den Plan anzupassen, müssen ggf. geplante Aktivitäten wegfallen, vorgezogen werden oder mit weniger Aufwand durchgeführt werden. Hierbei müssen kompetent Abstriche gemacht werden, wo sie am wenigsten Schaden anrichten, z.B. einzelne Stakeholder-Gruppen werden nicht befragt, einzelne offene Fragen nicht geklärt, Details nicht spezifiziert, unwichtige Änderungsanträge abgelehnt und so weiter. Wichtig ist hierbei eine Risikobetrachtung: Überwiegt der Nutzen der Einsparung deren möglichen Schaden?

9.5 Prozessverbesserung im Requirements Engineering Prozess (K3)

Ein Prozess lässt sich immer noch weiter verbessern. Die Grundlage für jede Prozessverbesserung ist die Analyse und Dokumentation des Ist-Zustands (vgl. 9.3). Der Ist-Zustand des Requirements Managements kann in einem Requirements Management Plan RMP dokumentiert werden. Dieser beschreibt das Requirements Management eines Unternehmens oder eines Projektes: den RE-Prozess, das Requirements Information Model, die Attribute und Sichten sowie alle anderen in diesem Lehrplan beschriebenen Festlegungen. Dieser RMP dient als Dokumentation des Ist-Zustands als Grundlage für die Reflektion des aktuellen Vorgehens und kann auch die geplante Prozessverbesserung strukturieren.

Eine Prozessverbesserung kann man abrupt als Prozessumstellung durchführen oder kontinuierlich. Eine Prozessumstellung verändert viele Aktivitäten und Parameter des Prozesses zugleich. Das hat den Vorteil, dass man so eine starke Effizienzsteigerung bewirken kann, die sich allerdings meist erst einstellt, nachdem sich alle Beteiligten an den neuen Prozess gewöhnt haben. Es besteht aber auch die Gefahr, dass der neue Prozess sich nicht bewährt und die Effizienz senkt. Eine Rückumstellung ist dann wieder mit großem Aufwand verbunden.

Die kontinuierliche Prozessverbesserung vermeidet dieses Risiko und führt mit wenig Aufwand kurzfristig zu Verbesserungen. Nach dem Prinzip der kontinuierlichen Prozessverbesserung (KPV) optimiert man Prozesse allmählich, indem man die folgenden vier Aktivitäten (PDCA) iterativ wiederholt:

- **Plan / Planen:** Man analysiert den Ist-Prozess und insbesondere vorhandenen Verbesserungsbedarf. Darauf aufbauend plant und dokumentiert man den Soll-Prozess.
- **Do / Umsetzen:** Man entwickelt und testet Verbesserungsmaßnahmen in einem Pilotprojekt und begleitet sie durch Messungen.
- **Check / Überprüfen:** Man prüft, ob die Maßnahmen die gewünschte Verbesserung bewirkt haben. Dafür vergleicht man die Ist-Werte mit den Plan-Werten.
- **Act / Handeln:** Aufgrund der Ergebnisse des Ist-Plan-Vergleichs werden Verbesserungsmaßnahmen durchgängig eingeführt oder falls nötig neue Maßnahmen geplant. Die Umsetzung der Maßnahmen wird überwacht und durch Messungen begleitet.

Den Ist- und Soll-Prozess charakterisiert man anhand von Messgrößen (vgl. 8). Solche Messgrößen können sein:

- Der Anteil des Projektbudgets, der in das Requirements Engineering investiert wurde. Sowohl ein zu hoher Anteil als auch zu wenig kann bedenklich sein. Normal sind 10–30 % des Projektbudgets.
- Die (nach voraussichtlichem Aufwand gewichtete) Anzahl der noch zu implementierenden Anforderungen. Sie misst die bis Projektende noch zu erledigende Arbeit.
- Burndown-Rate bzw. Velocity, d.h. die nach Aufwand gewichtete Menge der Anforderungen, die pro Zeiteinheit implementiert werden. Zusammen mit der Kenntnis der noch zu implementierenden Anforderungen lassen sich so Prognosen über die Restdauer des Projektes machen.
- Änderungsrate der Anforderungen. Als normal gilt eine Rate von 1–5% der Anforderungen pro Monat (gemessen in Aufwand) und 30–50% über die Projektlaufzeit [EBER2012]. Gibt es weniger Änderungen, kann das bedeuten, dass sich niemand wirklich für die Anforderungen interessiert und die Stakeholder nicht genügend eingebunden sind. Zu viele Änderungen sind ebenfalls ein Alarmzeichen: Die Anforderungen sind noch nicht stabil, eventuell sind die Stakeholder-Gruppen zu heterogen oder in Konflikt, und es ist noch zu früh, um die Anforderungen zu implementieren.
- Durchlaufzeit von Änderungsanträgen von der Beauftragung bis zur Umsetzung.

Mit Hilfe von Benchmarking lässt sich herausfinden, welche Werte als Soll-Wert sinnvoll und erreichbar sind.

Verbesserungsmaßnahmen können sich entweder auf die in 9.2 beschriebenen Prozessparameter beziehen oder darauf, wie die einzelnen Aktivitäten im Detail durchgeführt werden, z.B. mit Hilfe welcher Methoden.

Eine weitere Möglichkeit der Prozessverbesserung besteht darin, die Fehler zu analysieren, die im Requirements Engineering gemacht werden, z.B. die bei der Lastenheft-Inspektion gefundenen Fehler oder auch die mit dem System ausgelieferten Fehler, die auf das Requirements Engineering zurück zu führen sind. Dann fragt man nach deren Ursachen und den Ursachen der Ursachen. So kommt man auf Ideen für Verbesserungsmaßnahmen.

Eine konkretere Hilfe zur Prozessverbesserung im Requirements Engineering (aber nicht nur dort) bieten Reifegradmodelle wie das CMMI [CMMI 2011] oder auch ITIL [BEIM2012], [EBEL2014]. Diese beschreiben Aktivitäten oder Praktiken, die zur Erreichung eines bestimmten Reifegrades durchgeführt werden müssen. Bisher nicht umgesetzte Aktivitäten und Praktiken neu einzuführen stellt dann eine Prozessverbesserung dar. Auch alle anderen Methoden der Prozessverbesserung sind einsetzbar wie TQM (Total Quality Management) und Six Sigma.

Speziell die Verbesserung des Requirements Engineerings unterstützt die Sammlung von Best Practices von Sommerville und Sawyer [SoSa1997].

Bei der konkreten Planung der Prozessverbesserung unterstützt der Action Plan [WIEG2005], Seite 66. Diese Action Plan enthält folgende Inhalte:

- Name des Verbesserungsprojektes,
- Datum,
- Ziele (der Verbesserung, ausgedrückt in geschäftlichen Zielen),
- Kenngrößen für den Erfolg (d.h. Erreichung der Ziele),
- Organisatorischer Einfluss der Veränderung,
- Teilnehmer (Mitarbeiter, deren Rollen und zeitliche Budgets),
- Mess- und Berichtsprozess (wann wird wer wie den Fortschritt der Maßnahmen dieses Plans überwachen und berichten),
- Abhängigkeiten, Risiken und Randbedingungen,
- Geschätztes Abschlussdatum aller Maßnahmen dieses Plans,
- Maßnahmen (3-10 pro Plan) mit verantwortlicher Person, Zieldatum, Zweck, Beschreibung, Liefergegenständen und Ressourcenbedarf.

Zu beachten ist bei der Verbesserung des RE-Prozesses, dass er nicht für sich allein optimiert werden kann, sondern nur in Zusammenarbeit mit den anderen Tätigkeiten des Projektes wie Projektmanagement, Entwicklung und Testen. Veränderungen im RE-Prozess werden sich auch auf deren Arbeit auswirken.

10 Anforderungsmanagement in agilen Projekten (K2)

Dauer: 1,5 Stunden

Begriffe: User Story, Sprint, Product Backlog, Burndown-Chart

Lernziele

- LZ 10.1 Grundprinzipien agiler Softwareentwicklung kennen (K1)
- LZ 10.2 Zentrale Aktivitäten und Artefakte im Anforderungsmanagement in agilen Projekten kennen (K1)
- LZ 10.3 Abbildung von RM-Tätigkeit auf Scrum-Tätigkeiten kennen (K2)

10.1 Vorwissen (K1)

Klassische Ansätze zur Entwicklung von Software und damit auch zum Requirements Engineering legen Wert auf einen Plan, der weit in die Zukunft reicht. Ende der 1990er und Anfang der 2000er Jahre kam eine Gegenbewegung hierzu auf. Diese rückte von langfristigen Plänen ab und legte Wert auf kurzfristige Pläne mit vielen Rückkopplungsschleifen. Nach diesem Unterschied richtet sich die Klassifikation von **Fehler! Verweisquelle konnte nicht gefunden werden.**: Die beiden Gruppen werden als „agil“ (wie im agilen Manifest vorgeschlagen) und als „plangetrieben“ (im Original: plan-driven) bezeichnet. Im Allgemeinen gilt das Agile Manifest als die gemeinsame Grundlage von allen agilen Ansätzen. Im agilen Manifest wird aus Sicht der Softwareentwickler gefordert [AgiMa2001]:

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

***Individuen und Interaktionen** mehr als Prozesse und Werkzeuge*

***Funktionierende Software** mehr als umfassende Dokumentation*

***Zusammenarbeit mit dem Kunden** mehr als Vertragsverhandlung*

***Reagieren auf Veränderungen** mehr als das Befolgen eines Plans*

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“

Diese Werte zeigen, dass es bei dem agilen Vorgehen eher um die Zusammenarbeit, Produktivität und individuelle Stärken des Teams als um Verträge und Dokumentationen geht. Hierin grenzen sich die agilen Methoden von plangetriebenen Ansätzen ab, die klare Vertragsgegenstände (z.B. wie z.B. Projektumfang, Release-Pläne oder einen definierten Änderungsprozess) fordern.

10.2 Anforderungsmanagement in agilen Projekten (K1)

Sowohl die Gruppe der agilen als auch die plangetriebenen Ansätze sind in sich heterogen (siehe z.B. [KoBE2013]). Daher lässt sich nicht grundsätzlich festlegen, wie in agilen Projekte mit Anforderungen umgegangen wird. Die folgenden Erklärungen sind daher nicht als vollständig oder umfassend zu verstehen. Im Folgenden wird in daher in erster Linie Scrum [SuSc2013] als typischer Vertreter agiler Vorgehensweisen herangezogen.

Scrum ist ein „Rahmenwerk, mit dessen Hilfe Menschen komplexe adaptive Aufgabenstellungen angehen können, und durch das sie in die Lage versetzt werden, produktiv und kreativ Produkte mit dem höchstmöglichen Wert auszuliefern“ [SuSc2013].

Die am meisten diskutierte Form der Anforderung in agilen Projekten ist die User Story. Aus einer User Story wird die anfordernde Rolle, die Anforderung selbst und der Nutzen ersichtlich. Der Product Owner (PO) ist für die User Story verantwortlich. Das Entwicklungsteam (Development Team, abgekürzt als DT) leitet sich hieraus konkrete Arbeitsanweisungen (Tasks) für einen Entwicklungszyklus (Sprint) ab.

Eine User Story hat in der Regel die folgende Form:

Als <ROLLE> möchte ich <ZIEL/WUNSCH>, so dass <NUTZEN>.

Dabei ist die Angabe des Nutzens optional. Zu einer User Story werden die Akzeptanzkriterien bzw. Testfälle genauer spezifiziert. Diese haben die folgende Form:

Unter Voraussetzung, dass <VORBEDINGUNG>, wenn <TRIGGER> dann <ERGEBNIS>.

User Stories werden oftmals annotiert mit einer Implementierungsschätzung („Wie komplex ist die Umsetzung der User Story?“) und ihrem Wertbeitrag zum Projekt/ zur Vision. User Stories sind in der Regel sortiert nach ihrer geplanten Implementierungsreihenfolge. User Stories können gruppiert sein, z.B. in Epics (eine User Story, die zu groß ist, um alleine umgesetzt zu werden und daher geteilt wird) oder in Themes (inhaltlich zusammengehörige User Stories).

Die meisten agilen Ansätze geben nicht streng vor, mit welchen Artefakten zu arbeiten ist. Der Use Case, der als Ausdrucksform auch in der klassischen Softwareentwicklung etabliert ist, wird ebenso als Artefakt im agilen Umfeld verwendet (z.B. [COCK2001], [JaSB2011]).

Bewertet und priorisiert werden die Artefakte (z.B. User Stories oder Use Cases) oftmals in einem Product Backlog. In Scrum obliegt dies dem Product Owner. Der Product Owner ist die Person, die die zu implementierenden Artefakte verwaltet. Dazu priorisiert der Product Owner die Artefakte im Product Backlog, wobei sich die Priorität am Wertbeitrag des Artefakts und den Bedürfnissen der Stakeholder orientiert. Die Implementierungsreihenfolge der Artefakte ergibt sich einerseits aus ihrer Priorität und andererseits aus technologischen Abhängigkeiten.

Ein schriftliches Berichtswesen wird in agilen Ansätzen durch stark interaktives Arbeiten minimiert: Die Kommunikation aller Beteiligten soll den Aufwand zum schriftlichen Berichten verringern. Dazu werden regelmäßige Treffen („Events“) genutzt.

Auch dies unterscheidet sich zwischen den verschiedenen Ansätzen. In Scrum werden tägliche Treffen des Entwicklungsteams („Daily Standup“), Reviews und Retrospectives genutzt. Bei Reviews wird das Ergebnis eines Entwicklungszyklus (in Scrum: „Sprint“) vorgestellt und vom Product Owner abgenommen. In einer Retrospective wird das Zusammenspiel eines Teams rückblickend bewertet. In schriftlicher Form wird z.B. durch Burndown-Charts („Wie viel Arbeit ist in diesem Sprint noch zu erledigen?“) oder durch ein Impediment-Backlog („Was behindert das Team in seiner Arbeit?“) berichtet. Der Scrum Master (abgekürzt SM) sichert die regelkonforme Durchführung dieser Treffen.

Der Begriff des Prozesses wird zwar in agilen Ansätzen selten verwendet, jedoch implizieren die Leitlinien der Ansätze einen konkreten Arbeitsablauf. Die Verbesserung dieser Abläufe wird durch regelmäßige Retrospectives erreicht.

Die Skalierung von agilen Ansätzen auf große und verteilte Teams ist in den Anfängen, aktuell bilden sich einige Frameworks heraus. Ansätze hierzu finden sich z.B. in [ECKS2004], [ECKS2010], [LEFF2011] sowie [KoBE2013].

10.3 Abbildung von Requirements Management-Tätigkeiten auf Scrum-Tätigkeiten (K2)

Scrum gibt nur allgemeine Arbeitsprozesse vor. In der folgenden Tabelle 4 sind die RM-Tätigkeiten den Scrum-Tätigkeiten bzw. Artefakten zugeordnet. Weiterhin wird die ausführende Rolle in Scrum angegeben. Nicht zu allen RM-Tätigkeiten macht der Scrum Guide eine Aussage. Neben dem Scrum-Guide gibt es eine unübersichtliche Anzahl von Literatur, die mehr oder weniger erfolgreiche Ergänzungen zu Scrum beschreiben. Ob und wie die entsprechende RM-Tätigkeit dann in einem Scrum-Projekt ausgeführt wird, bleibt dem Scrum-Team überlassen.

RM-Aktivität	Scrum-Tätigkeit bzw. Artefakt	Scrum-Rolle
Attribut	User Stories im Backlog: Beschreibung, Reihenfolge, Schätzung, Status und Wert. Optional: Gruppierung	PO, DT
Bewertung und Priorisierung	<ul style="list-style-type: none"> ▪ Schätzung von Nutzen und Aufwand durch Planning Poker (DT) ▪ Anordnung der User Stories im Product-Backlog (PO) ▪ Auswahl der User Stories für einen Sprint (PO und DT) ▪ Priorisierung innerhalb eines Sprints (DT) 	<ul style="list-style-type: none"> ▪ DT ▪ PO ▪ PO und DT ▪ DT

RM-Aktivität	Scrum-Tätigkeit bzw. Artefakt	Scrum-Rolle
Verfolgbarkeit	<p>Es besteht eine implizite Verfolgbarkeit von User Stories zu den zugehörigen Abnahmetestfällen sowie, bei geeigneter Attributierung, zurück zu den Quellen der User Stories.</p> <p>Darüber hinaus ist eine Verfolgbarkeit innerhalb des Product Backlog (Abhängigkeiten) und von User Stories zum Sourcecode möglich.</p> <p>Scrum sagt nichts über die Verbindung von User Stories innerhalb des Product Backlog aus. Denkbar wäre eine Verfolgbarkeit über Epics (gruppierte User Stories).</p> <p>Verfolgbarkeit wird nur dokumentiert, wenn deren Notwendigkeit gegeben ist.</p>	Keine
Versionierung von Anforderungen	Eine Versionierung von User Stories ist unnötig. Relevant ist immer die jeweils aktuelle Version einer User Story.	Keine bzw. PO
Änderungen	Änderungen können jederzeit vorgeschlagen werden. Neue Anforderungen führen zu neuen User Stories, Anforderungsänderungen dazu, dass eine User Story geändert oder durch eine neue ersetzt wird.	PO
Variantenmanagement	Agile Methoden unterstützen Variantenmanagement nicht ausdrücklich. Der Einsatz üblicher Methoden des Variantenmanagements ist jedoch möglich.	PO
Berichtswesen	<p>Berichte erfolgen vor allem mündlich. Die zur Verfolgung des Bearbeitungsstands verwendeten Artefakte können gleichzeitig auch als Berichte dienen:</p> <ul style="list-style-type: none"> ▪ Daily Standup ▪ Sprint Review ▪ Sprint Retrospective ▪ Produkt-Auftragsbestand (Produkt-Backlog) ▪ Sprint-Auftragsbestand, Sprint-Backlog ▪ Erledigungsdiagramm, Burndown Chart 	DT
Prozessmanagement	Sprint Retrospective und Impediment Backlog	SM, DT

Tabelle 4: Abbildung von Tätigkeiten im Anforderungsmanagement zu SCRUM-Tätigkeiten

11 Werkzeugeinsatz im Anforderungsmanagement (K1)

Dauer: 0,75 Stunden

Begriffe: Werkzeug (Tool), Werkzeugauswahl, Datenaustausch

Lernziele

- LZ 11.1 Rolle von Werkzeugen im Anforderungsmanagement und deren Kernfunktionen kennen (K1)
- LZ 11.2 Prinzipielle Vorgehensweise bei der Werkzeugauswahl von Requirements Management Werkzeugen kennen (K1)
- LZ 11.3 Notwendigkeit eines werkzeugübergreifenden Datenaustauschs kennen (K1)

11.1 Rolle von Werkzeugen im Anforderungsmanagement (K1)

Der Einsatz von Werkzeugen soll dem Requirements Manager die Dokumentation und Verwaltung von Anforderungen vereinfachen. Werkzeuge für das Anforderungsmanagement ermöglichen durch ihre speziellen Funktionalitäten eine ganzheitliche Betrachtung von Anforderungen, indem u.a. Beziehungen zwischen unterschiedlichen Anforderungen sowie der Lebenszyklus einzelner Anforderungen abgebildet werden können.

Ein Werkzeug für das Anforderungsmanagement ist eine softwaretechnische Anwendung, deren Hauptziel die Unterstützung von Aktivitäten im RM ist.

In der Software- und Systementwicklung werden traditionell viele Anwendungen eingesetzt. Viele davon beinhalten Teilaspekte des Requirements Engineerings und/oder des Requirements Managements. Die Abgrenzung zu Werkzeugen für das Anforderungsmanagement ist dabei nicht immer trennscharf.

Werkzeugen für das Anforderungsmanagement liegen oft Annahmen über die Vorgehensweise bei der Entwicklung zugrunde. So sind manche Werkzeuge ausgerichtet auf bestimmte:

- Vorgehensmodelle (z.B. agile Entwicklung, Prototyping oder Produktmanagement)
- Anwendungsdomänen (z.B. Automobilbranche, Medizintechnik oder Verteidigungssysteme)
- Arbeitsumgebungen (z.B. virtuelles Büro, lokale Zusammenarbeit oder globale Zusammenarbeit)

Hieraus ergeben sich sehr unterschiedliche Requirements Management Werkzeuge, die oft nur einen Teil der Aufgaben eines Requirements Managers unterstützen.

Zentrale Funktionen eines Requirements Management Werkzeugs sind [SoSa1997]:

- Editor für Anforderungen inkl. deren Attribute
- Import von Anforderungen aus bestehenden Dokumenten in das Werkzeug und Export von verwalteten Anforderungen in andere Formate
- Verfolgung von Anforderungen

- Versionierung von Anforderungen sowie der Bildung von Konfigurationen
- Bildung von Sichten auf Anforderungen

Wie in [RuSo2009] beschrieben, muss beachtet werden, dass das Requirements Management Werkzeug zu den im Unternehmen etablierten Vorgehensweisen und Prozessen passt. Sofern der Vorgehensprozess und das grundlegende Datenmodell des Unternehmens erstellt wurden, kann eine Werkzeugevaluierung erfolgen.

11.2 Prinzipielle Vorgehensweise bei der Werkzeugauswahl (K1)

In Anlehnung an [RuSo2009] sind prinzipiell folgende Schritte bei der Auswahl eines Werkzeugs für das Anforderungsmanagement notwendig:

- Projekt zur Werkzeugauswahl aufsetzen
- Festlegung von Kriterien für eine Grobauswahl, indem grundlegende Anforderungen formuliert werden.
- Durchführung der Grobauswahl (Longlist), um erste potenzielle Systeme zu identifizieren.
- Verfeinerung des Kriterienkatalogs anhand neuer und verfeinerter Anforderungen an das Werkzeug.
- Durchführung einer Feinauswahl (Shortlist), bis hin zu einem favorisierten Software-Kandidaten.
- Sollte kein Werkzeug genau den Anforderungen entsprechen, ist eine Werkzeuganpassung der softwaretechnischen Anwendung notwendig (Customizing).
- Um die Akzeptanz im Unternehmen zu stärken und mögliche letzte Zweifel zu beseitigen, wird abschließend ein Pilotprojekt aufgesetzt.

Sollte das Pilotprojekt die Erkenntnis bringen, dass das ausgewählte Werkzeug doch nicht die gewünschte Unterstützung bringt, muss die Werkzeugauswahl wiederholt werden. Wenn kein Werkzeug genau den Anforderungen entspricht, kann statt einer Anpassung eines Werkzeuges auch eine Prozessanpassung erfolgen.

11.3 Datenaustausch zwischen Requirements Management Werkzeugen (K1)

Arbeiten im Rahmen eines Projektes mehrere Unternehmen zusammen, ergibt sich oft die Notwendigkeit, Anforderungen zwischen verschiedenen Requirements Management Werkzeugen auszutauschen.

Dies ist beispielsweise der Fall, wenn ein Auftragnehmer die Anforderungen des Auftraggebers im Rahmen seiner System-Entwicklung weiter verfeinert oder eine Verfolgung zu nachfolgenden Entwicklungsartefakten wie Design oder Testfällen aufbauen möchte.

Hierbei ist der Austausch über Standard-Dokumentformate, wie sie üblicherweise exportiert und importiert werden können unzureichend, da hierbei in der Regel relevante Informationen

wie ein eindeutiger Bezeichner, Informationen über den Bearbeiter, Strukturinformationen oder die Versionsangaben von Anforderungen verloren gehen. Vor allem aber unterstützt so ein Vorgehen einen aktualisierenden Austausch (z.B. nachdem Änderungen eingearbeitet wurden) nur völlig unzureichend.

Um einen werkzeugübergreifenden Austausch von Anforderungen unter Beibehaltung zentraler Anforderungseigenschaften zu ermöglichen, wurde der Industriestandard Requirements Interchange Format (ReqIF) geschaffen und durch die Object Management Group standardisiert.

Wesentliche Vorteile, die sich durch den Einsatz von ReqIF ergeben, sind dabei:

- Mit ReqIF kann die Zusammenarbeit zwischen Unternehmen verbessert werden, indem Methoden des RM unternehmensübergreifend angewandt werden.
- Die Partner müssen nicht mit dem gleichen Werkzeug arbeiten. Die Lieferanten müssen nicht für jeden Kunden ein eigenes Requirements Management Werkzeug besitzen.
- Innerhalb einer Organisation können Anforderungen übertragen werden, auch über Werkzeuggrenzen hinweg.
- Mit ReqIF können Anforderungen mit allen Attributen und Metainformationen, im Gegensatz zu Dokumentenexporten in Word, PDF etc., verlustfrei ausgetauscht werden.

Mittlerweile unterstützt eine Vielzahl an Werkzeugen das ReqIF Format.

12 Literaturverzeichnis

- [AgiMa2001] Manifesto for Agile Software Development. Abrufbar unter <http://agilemanifesto.org/>. Zuletzt besucht im Februar 2024.
- [ATKI2002] C. Atkinson: Component-based Product Line Engineering with UML. Addison-Wesley, 2002.
- [BaCR1984] V. Basili, G. Caldiera, and H. D. Rombach: The Goal Question Metric Approach. In: Encyclopedia of Software Engineering. John Wiley & Sons, 1994, S. 528–532.
- [BEIM2012] Martin Beims: IT-Service Management mit ITIL®: ITIL® Edition 2011, ISO 20000:2011 und PRINCE2® in der Praxis, Carl Hanser Verlag GmbH & Co. KG, 3., aktualisierte Auflage, 2012.
- [BoTu2003] B. Boehm und R. Turner: Using Risk to Balance Agile and Plan-Driven Methods. In: IEEE Computer, Volume 36, Issue 6, June 2003, pp. 57–66.
- [BOUT2011] E. Boutkova: Experience with Variability Management in Requirement Specifications. In: D.E. Almeida, T. Kishi, C. Schwanninger, I. John, and K. Schmid (eds): Software Product Lines – 15th International Conference (SPLC), München, 2013, S. 303–312.
- [BoHo2011] E. Boutkova and F. Houdek: Semi-automatic identification of features in requirement specifications. In: Proceedings of the 19th International Requirements Engineering Conference, Trento, Italy, September 2011, S. 313–318.
- [COCK2001] A. Cockburn: Use Cases effektiv erstellen. Addison-Wesley, 2001.
- [CzEi2000] K. Czarnecki and U.W. Eisenecker: Generative Programming: Methods, Tools, and Applications. Addison-Wesley, 2000.
- [CHSM2013] J. Cleland-Huang, R. S. Hanmer, S. Supakkul, and M. Mirakhorli: The Twin Peaks of Requirements and Architecture. IEEE Software, vol. 30, no. 2, S. 24–29, March–April, 2013
- [CINo2007] P. Clements and L. Northrop: Software Product Lines: Practices and Patterns. Addison-Wesley, Boston, 6. Auflage, 2007.
- [CMMI 2011] CMMI® für Entwicklung, Version 1.3. CMU/SEI-2010-TR-033. Abrufbar unter https://insights.sei.cmu.edu/documents/853/2010_005_001_15287.pdf. Zuletzt besucht im Februar 2024.
- [CoHW1998] J. Coplien, D. Hoffmann, and D. Weiss: Commonality and Variability in Software Engineering. In: IEEE Software, Volume 15, Issue 6, 1998, S. 37–45.
- [DAVI2003] A. Davis: The Art of Requirements Triage. IEEE Computer, Volume 36, Issue 3, 2003, S. 42–49.
- [DAVI2005] A. Davis: Just Enough Requirements Management, Dorset House, 2005.

- [EBEL2014] N. Ebel: ITIL®(R) 2011 Edition: Grundlagen und Know-how für das IT Service Management und die ITIL®(R)-Foundation-Prüfung, dpunkt.verlag GmbH, 1. Auflage, 2014
- [EBER2012] C. Ebert: Systematisches Requirements Engineering. Dpunkt, 4. Auflage, 2012.
- [ECKS2010] J. Eckstein: Agile Software Development with Distributed Teams. Dorset House Publishing, 2010.
- [ECKS2004] J. Eckstein: Agile Software Development in the Large. Dorset House Publishing, 2004.
- [GABLI2014] Springer Gabler Verlag (Herausgeber), Gabler Wirtschaftslexikon, Stichwort: Produktlinie, online im Internet:
<https://wirtschaftslexikon.gabler.de/definition/produktlinie.-43488>
- [GLIN2008] M. Glinz: Ein risikobasierter, wertorientierter Ansatz für Qualitätsanforderungen. IEEE Software, Nr. 2, S. 34-41, 2008.
- [GoFi1994] O.C.Z. Gotel and A.C.W Finkelstein: An Analysis of the Requirements Traceability Problem. In Proceedings of IEEE International Conference on Requirements Engineering, 1994, S. 94-101.
- [HuJD2011] E. Hull, K. Jackson, und J. Dick: Requirements Engineering. Springer, 3. Auflage, 2011.
- [DIN 61508] IEC DIN EN 61508-2 Funktionale Sicherheit sicherheitsbezogener elektrischer /elektronischer/ programmierbarer elektronischer Systeme. VDE Verlag, 2002.
- [IREB2020] Syllabus IREB Certified Professional for Requirements Engineering Foundation-- Foundation Level <https://www.ireb.org/de/downloads/#cprefoundation-level-syllabus-3-0>. Zuletzt besucht im Februar 2024.
- [IREB Glossary] M. Glinz: Wörterbuch der Requirements Engineering Terminologie. <https://www.ireb.org/de/downloads/tag:glossary#top>. Zuletzt besucht im Februar 2024.
- [ISO 12207] International Organization for Standardization (ISO): ISO/IEC 12207: 2008, Systems and software engineering – Software life cycle processes. 2008.
- [ISO 15288] International Organization for Standardization (ISO): ISO/IEC 15288: 2015, Systems and software engineering – Software life cycle processes. 15288.
- [ISO 26262] International Organization for Standardization (ISO): ISO 26262, Road vehicles -- Functional safety. 2011.
- [ISO 26550] International Organization for Standardization (ISO): ISO/IEC 26550:2013: Software and systems engineering -- Reference model for product line engineering and management. 2013.
- [ISO 9000] International Organization for Standardization (ISO): ISO 9000:2005: Quality management systems — Fundamentals and vocabulary. 2005.
- [JaSB2011] I. Jacobson, I. Spence, and K. Bittner: Use Cases 2.0. Ivar Jacobson International, 2011.

- [KCeA1990] C. Kang, S. Cohen, J. Hess, W. Novak, and A. Peterson: Feature-Oriented Domain Analysis (FODA) – Feasibility Study. Software Engineering Institute, 1990.
- [KaRy1997] J. Karlsson and K. Ryan: A Cost-Value Approach for Prioritizing Requirements. IEEE Software 14, Nr. 5, S. 67–74, 1997.
- [KoBE2013] H.-P. Korn and J.P. Berchez (eds.): Agiles IT-Management in großen Unternehmen. Symposium, 2013.
- [KÜPP2005] H.-U. Küpper: Controlling: Konzeption, Aufgaben, Instrumente. Schäffer-Poeschel, 4. Auflage, 2005.
- [LEFF2011] D. Leffingwell: Agile Software Requirements, Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison-Wesley Professional, 2011.
- [LoKa1995] P. Loucopoulos and V. Karakostas: System Requirements Engineering. McGraw-Hill, 1995.
- [MaGP2009] P. Mäder, O. Gotel, and I. Philippow: Getting Back to Basics: Promoting the Use of a Traceability Information Model in Practice. In: Proceedings of 5th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE2009), Vancouver, Canada, May 2009, S. 21–25.
- [MJZC2013] P. Mäder, P.L. Jones, Y. Zhang, and J. Cleland-Huang: Strategic Traceability for Safety-Critical Projects. In: IEEE Software, Volume 30, Issue 3, May / June 2013, S. 58–66.
- [MOIS2002] F. Moisiadis: The fundamentals of prioritizing requirements. In: Systems Engineering, Test & Evaluation Conference, Sydney, October 2002.
- [NEMN2006] B. Nemnich: Der Sarbanes-Oxley Act und die Wirkung auf die IT. Diplomarbeit, Grin Verlag, 2006.
- [NUSE2001] B. Nuseibeh: Weaving the Software Development Process between Requirements and Architecture. In: Proc. of ICSE2001 Workshop STRAW-01, Toronto, May 2001.
- [OMG2013] Requirements Interchange Format (ReqIF). Object Management Group, Version 1.1., 2013, available at <http://www.omg.org/spec/ReqIF/1.1/PDF/> Zuletzt besucht im Februar 2024.
- [PMI2021] PMI: Project Management Book of Knowledge (PMBOK). Project Management Institute, 7th Ed., 2021.
- [POHL1994] K. Pohl: The Three Dimension of Requirements Engineering: A Framework and its Application. Information Systems 3, 19 (June 1994), S. 243–258.
- [POHL1996] K. Pohl: Process-Centered Requirements Engineering. John Wiley Research Science Press, 1996.
- [POHL2010] K. Pohl: Requirements Engineering: Grundlagen, Prinzipien, Techniken. Springer, 2010.

- [PoBL2005] K. Pohl, G. Böckle, F. van der Linden: Software Product Line Engineering – Foundations, Principles, and Techniques. Springer, 2005.
- [PoRu2011] K. Pohl and Chris Rupp: Requirements Engineering Fundamentals. Rocky Nook, 2011.
- [RoRo2014] S. Robertson, J. Robertson: Mastering the Requirements Process – Getting Requirements Right. Addison–Wesley, 3rd Edition, 2014.
- [RuSo2009] C. Rupp & die SOPHISTen: Requirements–Engineering und –Management, Hanser, 5. aktualisierte und erweiterte Auflage, 2009.
- [SEI1999] Carnegie Mellon SEI (1999): The Capability Maturity Model, Guidelines for Improving the Software Process. Addison–Wesley, 1999.
- [SEI2010] Carnegie Mellon SEI (2010): CMMI for Services, Version 1.3, Improving processes for providing better services, 2010.
- [SCHI2001] B. Schienmann: Kontinuierliches Anforderungsmanagement. Addison–Wesley, 2001.
- [ScHT2006] P.–Y. Schobbens, P. Heymans, and J.C. Trigaux: Feature Diagrams: A Survey and a Formal Semantics. In: Proceedings of the 14th International Requirements Engineering Conference (RE’06), September 2006, S. 139–148.
- [SoSa1997] I. Sommerville and P. Sawyer: Requirements Engineering: A Good Practice Guide. John Wiley & Sons, 1997.
- [SuSc2013] J. Sutherland and K. Schwaber: Scrum Guide, July 2013, available at www.scrum.org. Zuletzt besucht im Februar 2024.
- [USCO2002] US Congress: Sarbanes–Oxley Act. Washington, USA, 107th Congress of the United States of America, 23.01.2002.
- [VANL2009] A. van Lamsweerde: Requirements Engineering – from System Goals to UML Models to Software Specifications. John Wiley and Sons, 2009.
- [WeLa1999] D. Weiss and C. Lai: Software product–line engineering: a family–based software development process. Addison–Wesley, 1999.
- [WIEG2005] Karl Wiegers: Software Requirements, Microsoft Press Deutschland, 1. Auflage, 2005.
- [WiBe2013] K. Wiegers and J. Beatty: Software Requirements, 3rd edition. Microsoft Press, 2013.
- [YOUN2004] R. Young: The Requirements Engineering Handbook, Artech House, Boston, 2004.
- [ZIEG1998] K. Ziegbein: Controlling. Kiehl Friedrich Verlag, 6. Auflage, 1998.