



Certified Professional for Requirements Engineering

Lehrplan

Requirements Elicitation

Practitioner | Specialist

Anja Brand, Dominik Häußer
Kim Lauenroth, Hans van Loenhoud,
Patrick Steiger

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

Version 1.0 dieses Lehrplans wurde durch Oliver Creighton, Dominik Häußer, Kim Lauenroth, Henriette Katharina Lingg, Thomas Mödl, Michael Richter, Chris Rupp, Dirk Schüpferling, Patrick Steiger und Malik Tayeh verfasst.

Bei Version 2.0 handelt es sich um eine umfassende Überarbeitung. Autoren dieser Version sind Dominik Häußer, Kim Lauenroth, Hans van Loenhoud, Anja Brand und Patrick Steiger.

Zu dieser Überarbeitung trugen Juliane Blechinger, Nikolaos Kaintantzis, Kostas Kolovos, Michael Richter, Stefan Sturm und Roger Wouterse durch ihr Feedback bei.

Das inhaltliche Review dieser Version erfolgte durch Birgit Penzenstadler.

Die Freigabe der englischen Version wurde am 11. Juli 2018 durch das IREB Council auf Empfehlung von Thorsten Weyer genehmigt.

Auf Anregung unserer Community wurde die ursprünglich als Advanced Level bezeichnete Zertifizierung in die zwei Stufen Practitioner und Specialist geteilt. Version 3.0 berücksichtigt diese Teilung.

Allen sei für ihr Engagement gedankt.

Das Urheberrecht © 2012–2024 für diesen Lehrplan besitzen die aufgeführten Autoren. Die Rechte sind übertragen auf den IREB International Requirements Engineering Board e.V., Karlsruhe, Deutschland.

Zweck des Dokuments

Dieser Lehrplan definiert die Lernziele und eine Zusammenfassung der Lerninhalte für die Requirements Elicitation 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 Elicitation richtet sich an Fachleute der Berufsbilder *Requirements Engineering, Geschäftsanalyse/Business-Analysis, Business-Engineering, Organisationsgestaltung* u. ä., welche ihre Kenntnisse und Fähigkeiten im Bereich Anforderungsermittlung 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.

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-Engineering besonders hervorzuheben.

Es wird definiert, was das Wissen von Requirements Engineers ausmacht, nicht jedoch die exakten Schnittstellen zu anderen Disziplinen und Prozessen des Software-Engineering.

Detaillierungsgrad

Der Detaillierungsgrad 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.

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 fünf 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 2: Anforderungsquellen (K3)

Dauer: 2,5 Stunden

Begriffe: Stakeholder, Anforderungsquelle, Relationship-Management, Nutzer, Persona

Das Beispiel zeigt, dass in Kapitel 2 Lernziele der Stufe K3 enthalten sind und zweieinhalb 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 Teilinhalte.

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

Beispiel: LZ 2.1.1

Das Beispiel zeigt, dass das Lernziel LZ 2.1.1 im Unterkapitel 2.1 beschrieben wird.

Die Prüfung

Dieser Lehrplan umfasst Lerneinheiten und Lernziele für die Zertifizierungsprüfungen zum

- Requirements Elicitation Practitioner
- Requirements Elicitation Specialist

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

Die Prüfung zum Erlangen des Requirements Elicitation 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 Elicitation Specialist, zu finden unter <https://www.ireb.org/de/downloads/#cpre-advanced-level-elicitation-specialist-written-assignment>.
- 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 Elicitation-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 Elicitation-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

Version	Date	Comment
1.0	23. März 2011	Erste veröffentlichte Version
1.0-1	1. Juni 2011	<p>Kleinere Korrekturen:</p> <ul style="list-style-type: none">▪ LE 3.1.1: Begriffe <i>Storytelling</i> und Zielermittlung entfernt▪ LE 3.1.2: Charakter <i>der</i> Aktivität Zielermittlung in Neutrale Technik korrigiert▪ LE 3.6.6: Einheitlich Begriff Zielermittlung verwendet
1.0-2	20. Dezember 2012	Rechtschreibkorrektur im Zusammenhang mit der Übersetzung ins Englische.
2.0.0	14. Februar 2019	<p>Umfangreiche Überarbeitung</p> <p>Der Name Elicitation and Consolidation der Version 1.0 wurde in Elicitation (alleine) umbenannt, um den mehrdeutigen Begriff „Consolidation“ zu entfernen.</p> <ul style="list-style-type: none">▪ Kapitel 1 (Konzepte zur strukturierten Planung und Durchführung der Anforderungsermittlung) ist neu. Es befasst sich mit der strukturierten Planung und Durchführung von Ermittlungs- und Konfliktlösungsaktivitäten.▪ Kapitel 2 (Anforderungsquellen) wurde in eine strukturiertere Form gebracht.▪ Kapitel 3 (Ermittlungstechniken) wurde komplett überarbeitet, um einen einheitlichen jedoch weniger ins Detail gehenden Überblick über die einzelnen Ermittlungstechniken zu geben.▪ Kapitel 4 (Konfliktlösung) wurde in eine strukturiertere Form gebracht. Der Kapitelname wurde geändert, um die terminologische Konsistenz zu verbessern.▪ Kapitel 5 (Fähigkeiten eines Requirements Engineers) ist eine in Struktur und Ausführlichkeit überarbeitete Version von Kapitel 1 der Lehrplanversion 1.0. <p>Im Vergleich zur Lehrplanversion 1.0, ist der Detailgrad dieser Lehrplanversion geringer. Detaillierte Ausführungen zu den einzelnen Themen sind dem <i>Handbuch Elicitation</i> zu entnehmen.</p> <p>Kognitive Stufen des Wissens entsprechend der neuen IREB Definition angewandt.</p>

Version	Date	Comment
2.0.1	28. März 2019	<p>Kognitive Stufen des Wissens in den Kapitelüberschriften korrigiert für die LE's 1, 2.1, 3, 3.3, 4, 4.1 und 5.</p> <p>Verwendung von Schlüsselwörtern für Stufen des Wissens korrigiert in den LZ's 1.1.1, 1.2.1, 1.2.2, 1.3.1, 1.3.2, 1.3.3, 1.4.1, 1.4.2, 2.1.1, 2.2.1.1, 2.2.4.1, 2.3.1.1, 2.4.1.1, 3, 3.1, 3.2, 3.2.1, 3.2.2, 3.2.3, 3.2.4, 3.3, 3.4, 4.1.1, 4.2.2, 4.3.1, 4.4.1, 5.1.1, 5.2.1, 5.4.1 und 5.5.1.</p> <p>Schreibfehler korrigiert.</p>
2.1.0	1. Januar 2022	<p>Definition von "Anforderung" entfernt, LE 3.3.2 (Problem, Ziel), Verweis auf [Glin2020].</p> <p>LZ 3.1.1 angepasst (Workshops in neue LE 3.1.3 verschoben).</p> <p>Neue LE 3.1.3 Kollaborationstechniken.</p> <ul style="list-style-type: none"> ▪ Kognitive Wissensstufe für Crowd-based Requirements Engineering auf K2 reduziert. ▪ Die frühere LE 3.1.3 über artefaktbasierte Techniken wurde nach LE 3.1.4 verschoben.
3.0.0	1. Juli 2022	Information über Advanced Level Prüfungssplit hinzugefügt. K5 bei den kognitiven Stufen des Wissens hinzugefügt.
3.0.1	17. August 2022	Kapitelnummern korrigiert. Das Kapitel „Stakeholder als Anforderungsquelle identifizieren und auswählen (K3)“ war fälschlicherweise auf Gliederungsebene 1 statt 3.
3.1.0	1. März 2024	Neues Corporate Design implementiert, kognitive Stufen des Wissens synchronisiert, Begriff Advanced Level eliminiert.
3.2.0	18. Juni 2024	Kognitive Stufen des Wissens erneut gefixt, LZ 1.3.4, 4.3.1 und 4.3.2 geändert, neue LZ 3.1.5 und 3.2.5 hinzugefügt.

Inhalt

1	Konzepte zur strukturierten Planung und Durchführung der Anforderungsermittlung und Konfliktlösung (K4).....	10
1.1	Der Anwendungsbereich von Ermittlung und Konfliktlösung im Requirements Engineering (K2).....	10
1.2	Relevante Einflussfaktoren für das Vorgehen zur Planung von Ermittlungs- und Konfliktlösungsaktivitäten (K2).....	11
1.3	Planung und Durchführung der Anforderungsermittlung und Konfliktlösung (K4).....	11
1.4	Prozessmuster (K2).....	13
2	Anforderungsquellen (K3).....	14
2.1	Grundlagen zu Anforderungsquellen (K3).....	14
2.2	Stakeholder identifizieren, klassifizieren und verwalten (K3).....	15
2.2.1	Stakeholder als Anforderungsquelle identifizieren und auswählen (K3).....	15
2.2.2	Stakeholder-Relationship-Management (K3).....	16
2.2.3	Dokumentationsschema für einbezogene Stakeholder (K3).....	16
2.2.4	Die Nutzer als besondere Stakeholdergruppe (K3).....	17
2.3	Dokumente identifizieren, klassifizieren und verwalten (K3).....	18
2.3.1	Dokumente als Anforderungsquelle identifizieren und auswählen (K3).....	18
2.3.2	Dokumentationsschema für Dokumente (K3).....	19
2.4	Systeme identifizieren, klassifizieren und verwalten (K3).....	19
2.4.1	Systeme als Anforderungsquelle identifizieren und auswählen (K3).....	19
2.4.2	Dokumentationsschema für Systeme (K3).....	20
3	Ermittlungstechniken (K4).....	22
3.1	Erhebungstechniken (K4).....	22
3.1.1	Fragetechniken (K3).....	22
3.1.2	Beobachtungstechniken (K3).....	23
3.1.3	Kollaborationstechniken (K3).....	24
3.1.4	Artefaktbasierte Techniken (K3).....	25

3.2	Entwurfs- und Ideenfindungstechniken [K4]	25
3.2.1	Brainstorming [K3]	26
3.2.2	Analogietechniken [K2]	26
3.2.3	Prototyping [K3]	26
3.2.4	Verwendung von Szenarien und Storyboards [K3]	27
3.3	Denkwerkzeuge [K2]	27
3.3.1	In Abstraktionsebenen denken [K2]	27
3.3.2	In Problemen und Zielen denken [K2]	28
3.3.3	Vermeidung von Transformationseffekten [K2]	28
3.3.4	Modelle als Denkwerkzeug [K2]	28
3.3.5	Mind-Mapping [K3]	29
3.4	Beispiel für die Strukturierung von Ermittlungstechniken: Attribute [K2]	29
4	Konfliktlösung [K4]	33
4.1	Konfliktidentifizierung [K2]	33
4.2	Konfliktanalyse [K3]	34
4.3	Konfliktlösung [K4]	35
4.4	Dokumentation der Konfliktlösung [K2]	36
5	Fähigkeiten eines Requirements Engineers [K3]	37
5.1	Benötigte Fähigkeiten im Bereich Ermittlung [K2]	37
5.2	Kommunikationstheorie und Kommunikationsmodelle [K2]	37
5.3	Selbstreflexion zu den eigenen Fähigkeiten in der Anforderungsermittlung [K3]	38
5.4	Möglichkeiten der persönlichen Weiterentwicklung [K2]	39
5.5	Aus vorherigen Erfahrungen lernen [K2]	39
6	Literaturverzeichnis und weitere Lektüre	40

1 Konzepte zur strukturierten Planung und Durchführung der Anforderungsermittlung und Konfliktlösung (K4).

Dauer: 1,5 Stunden

Begriffe: Ermittlungsaktivität, Konfliktlösungsaktivität, Technik, Prozessmuster

Lernziele

- LZ 1.1.1 Den Anwendungsbereich von Ermittlung und Konfliktlösung im Requirements Engineering verstehen
- LZ 1.2.1 Die Herausforderungen in der Ermittlungs- und Konfliktlösungsplanung verstehen
- LZ 1.2.2 Die für das Vorgehen zur Planung von Ermittlungs- und Konfliktlösungsaktivitäten relevanten Einflussfaktoren verstehen
- LZ 1.3.1 Die Informationsstruktur für Ermittlungs- und Konfliktlösungsaktivitäten anwenden
- LZ 1.3.2 Den Unterschied zwischen kurz- und langfristigen Ermittlungs- und Konfliktlösungsaktivitäten verstehen
- LZ 1.3.3 Die Bedeutung der Vorbereitungsphase für Ermittlung und Konfliktlösung verstehen
- LZ 1.3.4 Analysieren von Erhebungs- und Konfliktlösungsaktivitäten, um die richtigen auszuwählen und einzusetzen
- LZ 1.4.1 Die Bedeutung der Anpassung von Ermittlungs- und Konfliktlösungstechniken an den jeweiligen Kontext verstehen
- LZ 1.4.2 Das Konzept der Prozessmuster verstehen

1.1 Der Anwendungsbereich von Ermittlung und Konfliktlösung im Requirements Engineering (K2)

Laut der Definition von Requirements Engineering, wie sie in [PoRu2015] beschrieben ist, ist das Ziel der Anforderungsermittlung und Konfliktlösung „die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen“, und „die Wünsche und Bedürfnisse der Stakeholder zu verstehen“.

Im Rahmen der Anforderungsermittlung hat der Requirements Engineer die Aufgabe, die Wünsche und Bedürfnisse der Stakeholder zu verstehen und gleichzeitig sicherzustellen, dass die Anforderungen aller relevanter Anforderungsquellen ermittelt werden. Dazu gehört es, diese Quellen zu identifizieren, die Beschaffenheit und Bedeutung der einzelnen Anforderungsarten zu verstehen und geeignete Techniken anzuwenden, um sie zu ermitteln. Ein wesentlicher Teil der Ermittlung von Anforderungen ist es, implizite Forderungen, Wünsche und Erwartungen in explizite Anforderungen zu überführen [ISO29148].

Bei der Anforderungsermittlung begegnet man sehr häufig Anforderungen aus unterschiedlichen Quellen, die zueinander im Widerspruch stehen. Diese Anforderungskonflikte müssen aufgelöst werden, um eine konsistente und abgestimmte Zusammenstellung von Anforderungen zu erhalten, die als Ausgangsbasis für die effiziente Entwicklung eines zweckmäßigen Systems dient und auch dessen Wartung und Betrieb effizient unterstützt.

1.2 Relevante Einflussfaktoren für das Vorgehen zur Planung von Ermittlungs- und Konfliktlösungsaktivitäten (K2)

Gemäß Veröffentlichungen zu Aufwandsschätzungen in der Softwareentwicklung [McCo2006] sowie Ergebnissen aus der industriellen Praxis, hat die Disziplin des Requirements Engineerings einen großen Anteil am generellen Erreichen der Ziele eines Entwicklungsprojekts. Im Requirements Engineering, wiederum, hängt das Erreichen der Ziele in erheblichem Maße von der Anforderungsermittlung und Konfliktlösung ab.

Beide erfordern ein bestimmtes Vorgehen bei der Planung, um den folgenden Herausforderungen zu begegnen:

- Die Anforderungsermittlung kann nicht allein auf Basis der erwarteten Größe des Endergebnisses geplant werden, da zu Beginn der Ermittlungstätigkeit keine realistischen Erwartungswerte verfügbar sind.
- Anforderungskonflikte können nicht geplant oder vorhergesagt werden. Der Requirements Engineer muss auf Konflikte reagieren, sobald sie aufkommen.

Somit ist es ratsam, eine Detailplanung zu vermeiden und stattdessen einen groben Vorabplan zur Anforderungsermittlung und Konfliktlösung zu erstellen. Die Planung und Durchführung der Anforderungsermittlung und Konfliktlösung sollten ähnlich einem Forschungsprojekt durchgeführt werden. Dies bedeutet, dass der Plan mit Fortschritt der Aktivitäten und der Verfügbarkeit weiterer Informationen iterativ überarbeitet wird.

1.3 Planung und Durchführung der Anforderungsermittlung und Konfliktlösung (K4)

Obwohl die Anforderungsermittlung und Konfliktlösung einen bestimmten Planungsansatz erfordern, kann ihre Planung und Durchführung nicht unabhängig von anderen Tätigkeiten in der Systementwicklung betrachtet werden. Zur Festlegung eines Planungskonzepts wird davon ausgegangen, dass jedes Entwicklungsprojekt welches Ermittlungs- und Konfliktlösungsaktivitäten enthält, irgendeine Art Plan verwendet, um das Vorhaben und dessen Aufgaben zu strukturieren. Mit Fortschritt des Projekts, muss der Plan angepasst und aktualisiert werden.

Zwei Arten von Tätigkeiten können in jede Art von Plan integriert werden:

- *Ermittlungsaktivitäten*: die Identifizierung von Anforderungsquellen und die Ermittlung von Anforderungen.
- *Konfliktlösungsaktivitäten*: die Tätigkeiten die nötig sind, um Anforderungskonflikte zu lösen und zu einer einheitlichen und abgestimmten Zusammenstellung von Anforderungen zu gelangen.

Eine Ermittlungsaktivität sollte die folgenden Aspekte umfassen: das *Ermittlungsziel*, die angestrebte *Ergebnisqualität*, die gewählte(n) *Quelle(n)* sowie die gewählte *Ermittlungstechnik*.

Eine Konfliktlösungsaktivität sollte die folgenden Aspekte umfassen: die *betroffene(n) Anforderung(en)*, die *betroffene(n) Quelle(n)*, die gewählte *Konfliktlösungstechnik* und das *erreichte Ergebnis*.

Zusätzlich zu den Informationen, die die Ermittlung und Konfliktlösung betreffen, können beide Aktivitäten Verwaltungsinformationen bezüglich Zeit und Ressourcen enthalten.

Generell können drei verschiedene Konstellationen von Ermittlungs- und Konfliktlösungsaktivitäten unterschieden werden:

- Konstellation 1 – *Ausgeführte* Ermittlungs- und Konfliktlösungsaktivitäten
- Konstellation 2 – *Kurzfristige* Ermittlungs- und Konfliktlösungsaktivitäten
- Konstellation 3 – *Langfristige* Ermittlungs- und Konfliktlösungsaktivitäten

Im Laufe der Entwicklungstätigkeiten wächst der Anteil der ausgeführten Ermittlungsaktivitäten dadurch, dass kurzfristige Ermittlungsaktivitäten durchgeführt werden. Langfristige Aktivitäten werden weiter detailliert und zu kurzfristigen Aktivitäten oder werden durch mehrere kurzfristige Aktivitäten verfeinert oder werden völlig aufgegeben, da es keine Notwendigkeit mehr für sie gibt. Es wird empfohlen, zwischen der *Vorbereitungsphase* und der *Durchführungsphase* von Ermittlungs- und Konfliktlösungsaktivitäten zu unterscheiden.

Für die *Vorbereitungsphase* sind die folgenden Hinweise wichtig:

- Verschaffen Sie sich einen Überblick über die Projektsituation/das Geschäftsszenario.
- Legen Sie die Ermittlungsziele fest.
- Planen Sie die systematische Analyse des Systemkontextes ein.
- Planen Sie die systematische Identifizierung der (unterschiedlichen Arten von) Anforderungsquellen ein.
- Beziehen Sie bei der Definition von Aktivitäten relevante Prozessmuster ein.
- Rechnen Sie Zeit und Budget für Konfliktlösungsaktivitäten ein.

Für die *Durchführungsphase* sind die folgenden Hinweise wichtig:

- Betrachten Sie Ermittlung und Konfliktlösung als Tätigkeiten, die in begrenzten Zeitfenstern stattfinden (time-boxed).
- Hinterfragen Sie den Plan nach jeder durchgeführten Aktivität (und passen Sie ihn, sofern nötig, an).
- Planen Sie defensiv, indem Sie kurz- und langfristige Aktivitäten verwenden.
- Bauen Sie Puffer ein, um Luft für Kreativität und unerwartete Ereignisse zu haben.
- Parallelisieren Sie voneinander unabhängige Aktivitäten.
- Kombinieren Sie Ermittlungsaktivitäten, die dieselbe Anforderungsquelle nutzen.
- Suchen Sie nach Konflikten und reagieren Sie auf sie entsprechend eines vereinbarten Vorgehens.

Weiterhin ist es üblich, eine *Abschlussphase* durchzuführen, die zum Ziel hat, aus dem Projekt zu lernen und die Fähigkeiten der Projektteilnehmer weiterzuentwickeln. Richtlinien hierzu sind in 5 beschrieben.

1.4 Prozessmuster (K2)

Jedes Projekt ist einzigartig, so dass es keinen allgemeinen Ansatz gibt, der für alle Ermittlungsbedarfe gültig ist. In diesem Lehrplan wird daher das Konzept der *Prozessmuster* verwendet, um Ähnlichkeiten zwischen bestimmten Situationen zu identifizieren, welche dann als Richtlinien für anstehende Ermittlungsaktivitäten verwendet werden können. Passt ein einzelnes Prozessmuster nicht, kann eine Kombination oder eine Abfolge verschiedener Prozessmuster verwendet werden.

Das Konzept der Muster (Pattern) kommt ursprünglich aus dem Bereich der Architektur [AII1977]. Im Kontext der Anforderungsermittlung beschreibt ein Prozessmuster eine wiederverwendbare Methode zur Ermittlung von Anforderungen in einem bestimmten Rahmen (z. B.: Fachgebiet, Projektsituation).

Das Prozessmuster beschreibt die allgemeine Methode (Phasen, Reihenfolge der einzelnen Tätigkeiten) und bietet eine Anleitung für die einzelnen Ermittlungsaktivitäten, also der Festlegung der Ermittlungsziele, Auswahl der Techniken, Festlegung der Ergebnisqualität und der möglichen Anforderungsquellen.

Prozessmuster entstehen in einem bestimmten Kontext. Unter Prozessmustern werden hier alle Muster verstanden, aus denen neue oder verbesserte Anforderungen hervorgehen können. Sie können auch weitere Tätigkeiten umfassen (z. B.: Test, Design, Konfliktlösung).

Beispiele für Prozessmuster sind:

- Wasserfall-/Meilenstein-getriebene Entwicklung
- Lean Softwareentwicklung
- Agile Softwareentwicklung
- Human-Centered Design
- Design Thinking
- Embedded-Systems-Entwicklung
- Systemwartung

Requirements Engineers sollten aktiv nach Prozessmustern suchen, die für ihre eigene Situation relevant sind. Seien Sie sich über Folgendes bewusst:

- Prozessmuster sind bewährte Verfahren aus Literatur und Praxis, die als Ausgangsbasis dafür dienen, geeignete Ermittlungsaktivitäten in einer vergleichbaren Situation festzulegen.
- Für gewöhnlich sind die verfügbaren Informationen nicht ausreichend, um das Prozessmuster direkt anwenden zu können. Es ist daher nötig zu analysieren, inwiefern sich der Rahmen des Prozessmusters und die gegenwärtige Problemstellung gleichen oder unterscheiden, um ein geeignetes Vorgehen zu bestimmen und passende Techniken auszuwählen.
- Die Liste der oben aufgeführten Prozessmuster ist nicht abschließend. Vielmehr können und sollen Prozessmuster auf verschiedene Weise miteinander kombiniert werden.
- Erfahrene Requirements Engineers werden ermutigt, ihre eigenen Prozessmuster zu entwickeln und zu teilen.

2 Anforderungsquellen (K3)

Dauer: 2,5 Stunden

Begriffe: Stakeholder, Anforderungsquelle, Relationship-Management, Nutzer, Persona

Lernziele

- LZ 2.1.1 Die Bedeutung systematischer und pragmatischer Identifizierung von Anforderungsquellen im Systemkontext verstehen
- LZ 2.2.1.1 Typische Stakeholdergruppen kennen
- LZ 2.2.1.2 Die systematische Identifizierung und Auswahl von Stakeholdern anwenden
- LZ 2.2.2.1 Stakeholder-Relationship-Management zur Vermeidung und Lösung von Schwierigkeiten mit Stakeholdern anwenden
- LZ 2.2.3.1 Ein Dokumentationsschema für die einbezogenen Stakeholder anwenden
- LZ 2.2.4.1 Die Bedeutung des Anwenders als Stakeholder verstehen
- LZ 2.2.4.2 Personas anwenden
- LZ 2.3.1.1 Typische Dokumente als Anforderungsquellen verstehen
- LZ 2.3.1.2 Die systematische Identifizierung und Auswahl von Dokumenten anwenden
- LZ 2.3.2 Ein Dokumentationsschema für die einbezogenen Dokumente anwenden
- LZ 2.4.1.1 Typische Systemarten verstehen
- LZ 2.4.1.2 Die systematische Identifizierung und Auswahl von Systemen anwenden
- LZ 2.4.2 Ein Dokumentationsschema für die einbezogenen Systeme anwenden

2.1 Grundlagen zu Anforderungsquellen (K3)

Die Qualität und Vollständigkeit von Anforderungen hängen erheblich von den einbezogenen Anforderungsquellen ab. Eine relevante Quelle außer Acht zu lassen, führt zu einem unvollständigen Verständnis der Anforderungen. Während eines Entwicklungsprojekts ist es die Aufgabe des Requirements Engineers, alle relevanten Anforderungsquellen zu identifizieren und einzubeziehen. Wie im CPRE Foundation Level Lehrplan [IREB2022] dargelegt, sind die drei wichtigsten Arten von Anforderungsquellen Stakeholder, Dokumente und Systeme. Das Identifizieren von Anforderungsquellen ist ein iterativer und rekursiver Prozess [ISO29148] und muss wieder und wieder neu bewertet werden.

Der Requirements Engineer kann aus zwei verschiedenen Ansätzen zur Identifizierung von Anforderungsquellen wählen:

- Die *pragmatische* Identifizierung: Der Requirements Engineer nutzt sein gegenwärtiges Wissen und seine Erfahrung in Bezug auf das Projekt und seinen Kontext (z. B.: Fachwissen), um relevante Stakeholder, Dokumente und Systeme zu benennen.
- Die *systematische* Identifizierung: Der Requirements Engineer wendet eine bestimmte Strategie an, um durch die Auswahl bestimmter Ermittlungsaktivitäten, welche auf die Identifizierung von Anforderungsquellen ausgerichtet sind, mögliche Anforderungsquellen zu identifizieren.

Pragmatische und systematische Identifizierung ergänzen einander und bergen Risiken, wenn sie alleinstehend verwendet werden. Es wird daher mit Nachdruck empfohlen, eine Mischung aus beiden Ansätzen zu verwenden, um Anforderungsquellen effektiv und effizient zu ermitteln.

2.2 Stakeholder identifizieren, klassifizieren und verwalten (K3)

Im CPRE-Glossar [Glin2024], wird ein Stakeholder definiert als „eine Person oder Organisation, die Einfluss auf die Anforderungen eines Systems hat oder die von diesem System beeinflusst ist. Die Beeinflussung kann auch indirekt erfolgen. Beispielsweise müssen einige Stakeholder möglicherweise Anweisungen ihrer Vorgesetzten oder Organisationen befolgen.“.

2.2.1 Stakeholder als Anforderungsquelle identifizieren und auswählen (K3)

Der Requirements Engineer muss alle relevanten Stakeholder eines Entwicklungsprojekts identifizieren.

Die folgende nicht abschließende Liste führt typische Stakeholder-Rollen auf:

- Direkte Anwender des Systems
- Business-/Prozess-Manager
- Geschäfts- und Privatkunden; Organisationen, die Kunden repräsentieren
- Gegner und Wettbewerber
- IT-Personal
- Regierungs- und regulatorische Institutionen

Mögliche Quellen für relevante Stakeholder-Rollen sind:

- Checklisten mit typischen Stakeholdergruppen und -Rollen (siehe oben)
- Organisationsstrukturen (z. B.: Organigramme des Unternehmens, welches das zu bauende System dann nutzen wird)
- Geschäftsprozessdokumentation (z. B.: eines Geschäftsprozesses, der durch das zu entwickelnde System unterstützt werden soll)
- Stakeholder-Kategorisierungs-Schemata (z. B.: Alexanders Zwiebelschalenmodell [AlBe2009] oder der generische Stakeholder-Plan der Robertsons [RoRo2013])

Bei der *pragmatischen* Identifizierung der Stakeholder nutzt der Requirements Engineer sein gegenwärtiges Wissen und seine Kontexterfahrung (z. B. eines Fachgebiets), um relevante Stakeholder-Rollen und deren Vertreter (die Stakeholder) zu benennen.

Während der *systematischen* Identifizierung von Stakeholdern legt der Requirements Engineer Ermittlungsziele mit der klaren Ausrichtung auf die Identifizierung von Stakeholdern fest. Zwei verschiedene Arten von Ermittlungszielen sollten in Betracht gezogen werden:

- *Informationsbasierte* Ermittlungsziele: Das Finden von Stakeholdern, die für bestimmte Informationen benötigt werden.
- *Stakeholderbasierte* Ermittlungsziele: Das Finden von Stakeholdern, die bestimmte Stakeholder-Rollen repräsentieren.

Die anfänglich ermittelten Stakeholder sind nützliche Quellen für die Identifizierung weiterer Stakeholder.

2.2.2 Stakeholder-Relationship-Management (K3)

Schwierigkeiten mit Stakeholdern entstehen meist dann, wenn die Rechte und Pflichten eines Stakeholders in Bezug auf das zu entwickelnde System oder das aktuelle Projekt unklar sind, oder wenn die Bedürfnisse des Stakeholders unzureichend beachtet werden.

Stakeholder-Relationship-Management ist ein effektiver Weg, um Schwierigkeiten mit Stakeholdern entgegen zu wirken.

[Bour2015] empfiehlt den *Stakeholderkreis*, für erfolgreiches Stakeholder-Relationship-Management. Er besteht aus fünf Schritten:

1. Identifizierung aller Stakeholder
2. Priorisierung, um festzulegen, wer wichtig ist
3. Visualisierung, um die Gesamtheit der Stakeholder zu verstehen
4. Aktives Einbringen durch effektive Kommunikation
5. Prüfung der Wirksamkeit der Kommunikation

Aktives Stakeholder-Relationship-Management [Bour2009] legt genau die Rechte und Pflichten der Stakeholder in Bezug auf das zu entwickelnde System fest. Je nach Ausrichtung des Entwicklungsprojektes, kann es sinnvoll sein, diese auch mit den beteiligten Stakeholdern in einem Stakeholder-Vertrag festzuhalten.

2.2.3 Dokumentationsschema für einbezogene Stakeholder (K3)

Der CPRE Foundation Level Lehrplan [IREB2022] legt fest, welche Informationen über einen Stakeholder mindestens dokumentiert werden sollten. Zusätzlich sollten Informationen zur Klassifikation des Stakeholders und projektspezifische Attribute mit einbezogen werden.

Laut [AlBe2009], können Stakeholder danach klassifiziert werden, wie stark sie von dem neuen oder veränderten System betroffen sind:

- Stakeholder *des Systems*: Diese Stakeholder sind direkt von dem neuen oder geänderten System betroffen (Anwender, Administratoren, Betreiber, etc.).
- Stakeholder *des umgebenden Kontexts*: Diese Stakeholder sind indirekt von dem neuen oder geänderten System betroffen (Vorgesetzte, Projektverantwortliche, Sponsoren, Kunden, etc.).
- Stakeholder *aus dem weiteren Umfeld*: Diese Stakeholder haben eine indirekte Beziehung zu dem neuen oder geänderten System oder zum Entwicklungsprojekt (Gesetzgeber, Standardisierungsinstitute, (Nicht-)Regierungsorganisationen, Wettbewerber, IT).

Weiterhin kann es hilfreich sein, weitere Informationen festzuhalten, die für das konkrete Entwicklungsprojekt relevant sind.

Bei der Festlegung zusätzlicher Informationen müssen die konkreten Umstände des aktuellen Kontexts einbezogen werden. Mögliche Einflussfaktoren sind:

- *Öffentliche Relevanz:* Im Kontext hoher öffentlicher Relevanz kann es hilfreich sein, festzuhalten, wie viel ein Stakeholder weiß oder wie sehr er die öffentliche Meinung beeinflussen kann.
- *Zeitliche Kritikalität:* In einem Kontext mit sehr strengem Zeitrahmen können, wenn kritische Entscheidungen nötig sind, die Erreichbarkeit oder Antwortzeit eines Stakeholders sehr wichtige Informationen sein

Während des Entwicklungsprojekts müssen alle Stakeholderinformationen ständig aktualisiert und auf die konkreten Umstände angepasst werden.

Übliche Arten der Dokumentation sind Stakeholdertabelle, Stakeholderdatenbank und Stakeholder-Mind-Map.

2.2.4 Die Nutzer als besondere Stakeholdergruppe (K3)

Prinzipiell hat jedes System am Ende Anwender. Jedoch treten nicht alle Systeme direkt in Interaktion mit Menschen: Einige machen ihre Funktionalität mittels weiterer Systeme verfügbar. Bei interaktiven Systemen mit Schnittstelle zum Menschen, sind alle direkten Anwender des Systems von besonderem Interesse für den Requirements Engineer.

Interne Anwender (innerhalb des Unternehmens; individuell bekannt und eingebunden) unterscheiden sich maßgeblich von externen Anwendern (z. B. Käufer eines Massenprodukts; außerhalb des Unternehmens; gewöhnlich nicht individuell bekannt und nicht direkt eingebunden).

Die Anzahl (potenzieller) Anwender erlaubt es nicht, jeden Einzelnen in den Ermittlungsprozess einzubeziehen. So können die tatsächlichen Anwender basierend auf Nutzeranalysen oder dem Fachwissen anderer Stakeholder zu Nutzergruppen aggregiert werden.

Nutzergruppen werden häufig mithilfe von Personas dargestellt [Coop2004]. Personas sind fiktive Personen, die typische Nutzergruppen des Systems mit ähnlichen Bedürfnissen, Werten und Gewohnheiten repräsentieren. Personas werden auf Basis von Daten real existierender Anwender erstellt, welche mittels Nutzerforschung (*User-Research*) [BaCC2015] gesammelt wurden. Sind (noch) keine relevanten Daten aus der Nutzerforschung vorhanden, können *provisorische Personas*, auch *Ad-hoc-Personas* [CRCN2014] genannt, erstellt werden.

Die Nutzergruppen oder Personas sollten priorisiert werden, um primäre und sekundäre Nutzergruppen/Personas zu unterscheiden. Das System, besonders die zugehörige Benutzungsoberfläche, wird für die primäre Nutzergruppe optimiert.

Das Konzept der User-Experience (UX) widmet sich vornehmlich dem Schaffen einer positiven Erfahrung für den Anwender. User-Experience ist durch einen ISO-Standard definiert. [ISO9241-210] definiert User-Experience als „die Wahrnehmungen und Reaktionen, die durch den Gebrauch oder antizipierten Gebrauch eines Produkts, Systems oder Services herrühren.“ Für die Entwicklung interaktiver Systeme ist es dienlich, Ideen und Prinzipien der User-Experience zu kennen.

2.3 Dokumente identifizieren, klassifizieren und verwalten (K3)

Dokumente sind eine weitere nützliche Quelle für Anforderungen. Sie werden verwendet, um Informationen zwischen Menschen über Zeit und Raum hinweg zu übermitteln.

2.3.1 Dokumente als Anforderungsquelle identifizieren und auswählen (K3)

Folgende Dokumentenarten können u.a. als Anforderungsquellen dienen:

- Technische Standards, Gesetze, interne Vorschriften
- Anforderungsdokumente (z. B. von ähnlichen Systemen oder von dem zu ersetzenden System)
- Handbücher (z. B. eines Konkurrenzprodukts)
- Strategiepapiere
- Zieldokumentation
- Geschäftsprozess-Dokumentation

Bei der *pragmatischen* Identifizierung der Dokumente nutzt der Requirements Engineer sein gegenwärtiges Wissen und seine Kontexterfahrung (z. B. in einem Fachgebiet), um relevante Dokumente und Dokumentarten zu benennen.

Während der *systematischen* Identifizierung von Dokumenten legt der Requirements Engineer Ermittlungsziele mit der klaren Ausrichtung auf die Identifizierung von Dokumenten fest. Dabei müssen zwei verschiedene Arten von Ermittlungszielen in Betracht gezogen werden:

- *Informationsbasierte* Ermittlungsziele: Das Finden von Dokumenten, die für bestimmte Informationen benötigt werden.
- *Dokumentenbasierte* Ermittlungsziele: Das Finden von Vertretern bestimmter Dokumententypen, welche als relevant für das Projekt angesehen werden.

Um *Dokumente systematisch zu identifizieren*, kann der Requirements Engineer:

- Nach Repräsentanten typischer Dokumentenarten suchen
- Nach Verweisen auf andere möglicherweise relevante Dokumente in bereits identifizierten Dokumenten suchen
- Bereits identifizierte Stakeholder nach relevanten Dokumenten fragen
- Nach Dokumentation von bereits identifizierten relevanten Systemen suchen (vgl. 2.4)

Um zu entscheiden, ob ein Dokument als Anforderungsquelle relevant ist, oder nicht, muss der Requirements Engineer bestimmte Kriterien festlegen.

2.3.2 Dokumentationsschema für Dokumente (K3)

Für Dokumente, die potenziell als Anforderungsquellen verwendet werden, sollten mindestens die folgenden Informationen festgehalten werden:

- Titel des Dokuments
- Ort, an dem das Dokument aufbewahrt wird
- Version des Dokuments
- Kurze Beschreibung (welche Art von Information das Dokument mutmaßlich enthält)
- Relevanz

Abhängig vom Kontext können auch weitere Informationen relevant sein.

Dokumente stehen immer in einer bestimmten Beziehung zu Stakeholdern. Diese kann ebenfalls festgehalten werden, z. B.:

- Stakeholder, die auf die Relevanz des Dokuments hingewiesen haben
- Autor oder die publizierende Organisation
- Organisationen, die das Dokument in ihren Prozessen verwenden
- Organisationen, die daran beteiligt sind, die Einhaltung der im Dokument festgelegten Vorschriften zu prüfen

Der Requirements Engineer muss die dokumentationsbezogenen Informationen aktuell halten. Dies beinhaltet zu überprüfen, ob zwischenzeitlich weitere Dokumente relevant geworden sind, oder ob zuvor identifizierte Dokumente inzwischen an Relevanz verloren haben. Besondere Aufmerksamkeit sollte dabei auf Änderungen, Aktualisierungen und Versionsnummern gelegt werden.

2.4 Systeme identifizieren, klassifizieren und verwalten (K3)

Im direkten oder weiteren Kontext eines Systems können andere Systeme als Anforderungsquellen identifiziert werden.

2.4.1 Systeme als Anforderungsquelle identifizieren und auswählen (K3)

U.a. folgende Systemarten können als Anforderungsquellen dienen:

- Angrenzende Systeme, einschließlich Altsystemen
- Systeme, die auf derselben Plattform oder in derselben Umgebung/Umwelt betrieben werden
- Konkurrenzsysteme
- Systeme, die ähnliche Daten verarbeiten, ähnliche Funktionalitäten haben oder ähnliche Benutzungsschnittstellen bereitstellen
- Vorgängersysteme, die ersetzt werden sollen
- Zukünftige Systeme (bereits in Erstellung oder sogar nur in Planung)

Bei der *pragmatischen* Identifizierung der Systeme nutzt der Requirements Engineer sein gegenwärtiges Wissen und seine Kontexterfahrung (z. B. in einem Fachgebiet), um relevante Systeme und Systemarten zu benennen.

Während der *systematischen* Identifizierung von Systemen legt der Requirements Engineer Ermittlungsziele mit der klaren Ausrichtung auf die Identifizierung von Systemen fest. Zwei verschiedene Arten von Ermittlungszielen sollten in Betracht gezogen werden:

- *Informationsbasierte* Ermittlungsziele: Das Finden von Systemen, die für bestimmte Informationen benötigt werden.
- *Systembasierte* Ermittlungsziele: Das Finden von Vertretern bestimmter Systemtypen, welche als relevant für das Projekt angesehen werden.

Um Systeme *systematisch* zu *identifizieren*, kann der Requirements Engineer:

- Die Dokumentation zum Systemkontext verwenden
- Bereits identifizierte Stakeholder nach relevanten Systemen fragen
- Bereits identifizierte Dokumente nach Informationen über relevante Systeme durchsuchen
- Techniken zur Ideenfindung verwenden, um mögliche analoge Systeme zu identifizieren
- Eine Marktanalyse durchführen, um Konkurrenzsysteme zu ermitteln
- Altsysteme in Betracht ziehen

2.4.2 Dokumentationsschema für Systeme (K3)

Zu Systemen, welche als Anforderungsquelle dienen, müssen mindestens die folgenden Informationen dokumentiert werden:

- Name des Systems
- Art des Systems (z. B. Konkurrenzsystem, Vorgängersystem, Nachbarsystem, etc.)
- Eine kurze Beschreibung der Daten, Funktionen, Prozesse, Nutzergruppen, etc.

Abhängig vom Kontext können auch weitere Informationen relevant sein.

Besondere Aufmerksamkeit sollte den direkten Nachbarsystemen gewidmet werden. Diese lassen sich wie folgt kategorisieren:

- Datenquellen, welche Daten zur Verfügung stellen
- Datensinken, welche Daten nutzen
- Unterstützende Systeme, wie Betriebssysteme oder Datenbank-Management-Systeme

Systeme stehen immer in einer bestimmten Beziehung zu Stakeholdern. Diese kann ebenfalls festgehalten werden, z. B.:

- Stakeholder/Organisationen, die das System direkt oder indirekt in ihren Prozessen nutzen
- Stakeholder/Organisationen, die das System betreiben
- Stakeholder/Organisationen, die das System entwerfen, entwickeln oder vermarkten

- Stakeholder/Organisationen, die das System administrieren, es supporten oder schulen
- Organisationen, die das System beaufsichtigen (z. B. Regierung, Nichtregierungsorganisationen)

Informationen über Systeme finden sich für gewöhnlich in Dokumenten. Solche Dokumente sollten separat als Anforderungsquellen verwaltet werden (siehe 2.3).

Der Requirements Engineer muss die systembezogenen Informationen aktuell halten. Dies beinhaltet zu überprüfen, ob zwischenzeitlich weitere Systeme relevant geworden sind, oder ob zuvor identifizierte Systeme inzwischen an Relevanz verloren haben. Besondere Aufmerksamkeit sollte dabei auf Änderungen, Aktualisierungen und Versionsnummern gelegt werden.

3 Ermittlungstechniken (K4)

Dauer: 8,0 Stunden

Begriffe: Ermittlungstechnik, Attribut, Klassifizierung, Denkwerkzeuge

Lernziele

- LZ 3.1 Den Unterschied zwischen Erhebungstechniken, Entwurfs- und Ideenfindungstechniken und unterstützenden Techniken verstehen
 - LZ 3.1.1 Interview und Fragebogen als Beispiele für Fragetechniken anwenden
 - LZ 3.1.2 Feldbeobachtung, Apprenticing und Contextual Inquiry als Beispiele für Beobachtungstechniken anwenden
 - LZ 3.1.3 Workshops und crowd-based Requirements Engineering als Beispiele für Kollaborationstechniken anwenden
 - LZ 3.1.4 Systemarchäologie, Perspektivenbasiertes Lesen und Wiederverwendung als Beispiele für artefaktbasierte Techniken anwenden
 - LZ 3.1.5 Analysieren von Erhebungstechniken, um die am besten geeigneten auszuwählen.
- LZ 3.2 Voraussetzungen für Kreativität schaffen
 - LZ 3.2.1 Brainstorming als Beispiel für Ideenfindungstechniken anwenden
 - LZ 3.2.2 Analogietechniken als Beispiele für Ideenfindungstechniken verstehen
 - LZ 3.2.3 Prototypen als Beispiele für Entwurfstechniken anwenden
 - LZ 3.2.4 Szenarien und Storyboards als Beispiele für Entwurfstechniken anwenden
 - LZ 3.2.5 Analysieren von Entwurfs- und Ideenfindungstechniken, um die am besten geeigneten auszuwählen.
- LZ 3.3.1 Abstraktionsebenen, Probleme und Ziele, Modelle, Transformationseffekte und Mind-Mapping als Beispiele für Denkwerkzeuge verstehen
- LZ 3.4 Attribute für Ermittlungstechniken als beispielhaften Ansatz zur Strukturierung von Ermittlungstechniken verstehen

Dieses Kapitel unterscheidet zwischen Erhebungstechniken (3.1), Entwurfs- und Ideenfindungstechniken (3.2) und Denkwerkzeugen (3.3). Dies ist natürlich eine künstliche Unterscheidung. In der Praxis gibt es keine klare Trennung zwischen diesen Techniken. Zur Darstellung und Lehre ist diese Unterscheidung jedoch notwendig, um die Techniken zu strukturieren und um den Hauptschwerpunkt der Techniken zu vermitteln.

3.4 vermittelt typische identifizierende Eigenschaften von Ermittlungstechniken. Diese können verwendet werden, um neue Techniken zu beschreiben und um allgemeine Richtlinien dafür zu geben, welche identifizierenden Merkmale voraussichtlich in einer bestimmten Projektsituation hilfreich sind.

3.1 Erhebungstechniken (K4)

Erhebungstechniken sind etablierte Methoden zur Anforderungsermittlung. Sie helfen, Leistungs- und Basisfaktoren zu ermitteln.

3.1.1 Fragetechniken (K3)

Fragetechniken zielen darauf ab, Stakeholdern die passenden Fragen zu stellen. Dabei ist die Unterscheidung zwischen offenen und geschlossenen Fragen wichtig.

3.1.1.1 Interview (K3)

In einem Interview stellt der Requirements Engineer einem oder mehreren Stakeholdern Fragen, um neue Anforderungen zu ermitteln, oder bestehende weiter zu verfeinern. Es erfordert eine gründliche Vorbereitung. Während des Interviews müssen die Antworten auf geeignete Weise festgehalten werden, um die Nachbereitung der Interviewergebnisse zu unterstützen. Es gibt verschiedene Interview-Arten, z. B. Interviews mit einer genau festgelegten oder einer offenen Zusammenstellung von Fragen [Port2013], [BaCC2015].

3.1.1.2 Fragebogen (K2)

Mehrere Personen werden darum gebeten, dieselbe strukturierte Zusammenstellung von Fragen zu beantworten. Quantitative Fragebögen werden vor allem dazu verwendet, zuvor ermittelte Anforderungen zu bestätigen, während qualitative Fragebögen dazu geeignet sind, neue Anforderungen zu ermitteln. Erstere können schnell ausgewertet werden und liefern statistische Informationen, während Letztere eher komplexe Ergebnisse liefern und daher in der Regel zeitaufwändiger in Vorbereitung und Auswertung sind [BaCC2015], [Harr2014].

3.1.2 Beobachtungstechniken (K3)

Beobachtungstechniken zielen darauf ab, Anforderungen durch Beobachtung, z. B. von Prozessen, Anwendern oder typischen Anwendungssituationen, zu ermitteln.

Besondere Aufmerksamkeit sollte hier auf die mögliche Vereinfachung durch Voreingenommenheit des Beobachters („Simplification Bias“) gelegt werden [BaCC2015]: unerfahrene Beobachter (neu in dem Fachgebiet) tendieren dazu, die Problemlösungsstrategien des Experten während der Beobachtung zu vereinfachen. Daher wird sehr empfohlen, mit dem Experten vor Anwendung der Beobachtungstechnik zu sprechen und/oder den Experten die Beobachtungsnotizen reviewen zu lassen, um die Voreingenommenheit zu minimieren.

3.1.2.1 Feldbeobachtung (K3)

Der Requirements Engineer beobachtet die Stakeholder bei ihrer Tätigkeit in deren gewohnter Umgebung, ohne einzugreifen. Die Beobachtungen, die er dabei macht, verwendet er zur Ableitung von Anforderungen, welche durch Reviews oder weitere Ermittlungstechniken zu bestätigen sind.

3.1.2.2 Apprenticing (K2)

Der Requirements Engineer führt ein kurzes Praktikum in dem Umfeld durch, in dem das zu entwickelnde/verbessernde System später eingesetzt wird, oder bereits im Einsatz ist. Ein erfahrener Experte leitet dabei den Requirements Engineer an, um ihn zu befähigen, das Fachgebiet, und somit die ermittelten Anforderungen, besser zu verstehen.

3.1.2.3 Contextual Inquiry (K3)

Contextual Inquiry (CI) ist eine iterative, Felddaten-sammelnde Technik, bei der der Requirements Engineer einige mit Bedacht ausgesuchte Anwender in der Tiefe studiert, um zu einem besseren Verständnis der Arbeitsvorgänge der gesamten Anwenderbasis zu kommen [BeHo1998].

CI basiert auf vier Prinzipien:

- *Kontext*: Begeben Sie sich in den Kontext des Anwenders, um ihn bei der Durchführung seiner Aufgaben zu beobachten.
- *Partnerschaftlichkeit*: Fragen Sie ihn über seine Tätigkeit und beziehen Sie ihn dabei ein, unausgesprochene Aspekte seiner Tätigkeit aufzudecken.
- *Interpretation*: Entwickeln Sie ein gemeinsames Verständnis mit dem Anwender zu den wichtigen Aspekten der Tätigkeit.
- *Fokus*: Definieren Sie bei der Vorbereitung des CI Ermittlungsziele und richten Sie Ihre Ermittlung darauf aus, die relevanten Daten zu erheben, um die Ziele zu erreichen.

3.1.3 Kollaborationstechniken (K3)

Kollaborationstechniken zielen auf die Erhebung von Anforderungen ab und konzentrieren sich auf die Zusammenarbeit zwischen den Stakeholdern. In diesem kollaborativen Prozess sind Stakeholder mit unterschiedlichen Perspektiven, häufig Nutzer des Systems, direkt an der Erfassung, Entwicklung oder Verfeinerung von Anforderungen beteiligt. Diese Techniken bieten eine Plattform für Diskussionen und ermöglichen ein sofortiges Feedback der Beteiligten. Beispiele für Kollaborationstechniken sind Anforderungsworkshops, Crowd-based Requirements Engineering und Living Laboratories.

3.1.3.1 Anforderungsworkshop (K3)

Workshop ist ein Überbegriff für gruppenorientierte Techniken. Sie können auf verschiedene Weise durchgeführt werden und können andere Ermittlungstechniken oder sogar Prozessmuster enthalten (z. B. ein Design-Thinking-Workshop innerhalb eines agilen Entwicklungsprojekts). Workshop-Formate reichen von kleinen formlosen Meetings bis hin zu organisierten Veranstaltungen mit mehreren hundert Stakeholdern [Gott2002].

3.1.3.2 Crowd-based Requirements Engineering (K2)

Für einige Systeme (z. B. mobile Anwendungen), können Anforderungen aus „der Crowd“, also der Menge der (potenziellen) Anwender, ermittelt werden. Dies beinhaltet sowohl konkrete Informationen (z. B. Feedback oder Rezensionen) als auch implizite Informationen (z. B. Nutzungsdaten, Fehler-Logs) [MNJR2015], [GrDA2015].

3.1.4 Artefaktbasierte Techniken (K3)

Artefakte sind Ergebnisse menschlicher Tätigkeit (auch Arbeitsergebnisse genannt), z. B. IT-Systeme, Dokumente, Bilder, Audio- und Video-Dateien. Einige Arten dieser Arbeitsergebnisse sind als Quellen für Anforderungen relevant. In der Regel ist es eine zeitaufwändige Aufgabe, Arbeitsergebnisse im Detail zu prüfen.

3.1.4.1 Systemarchäologie (K3)

Systemarchäologie ist eine Technik, um Informationen in Bezug auf ein neues System aus der Dokumentation, Benutzungsoberfläche oder dem Code eines Alt- oder Konkurrenzsystems zu ermitteln. Es empfiehlt sich, zunächst Dokumente wie Spezifikationen, Test-Dokumentation oder Benutzerhandbücher zu analysieren, da sie Informationen enthalten, die Anforderungen ähnlich sind. Mit Hilfe der Systemarchäologie kann sichergestellt werden, dass Anforderungen, die bereits im Bestandssystem implementiert sind, nicht verloren gehen.

3.1.4.2 Perspektivenbasiertes Lesen (K3)

Der Requirements Engineer nutzt eine bestimmte Perspektive, z. B. Nutzerperspektive oder Informationsperspektive, um relevante Anforderungen aus einem Dokument herauszulesen [Pohl2010].

3.1.4.3 Wiederverwendung von Anforderungen (K3)

Falls ähnliche Projekte oder Vorgängerversionen des zu entwickelnden Systems innerhalb des Unternehmens existieren, können Anforderungen aus diesen Projekten wiederverwendet werden. Anforderungen, die wiederverwendet werden sollen, müssen durch Review oder weitere Ermittlungstechniken bestätigt werden.

3.2 Entwurfs- und Ideenfindungstechniken (K4)

Herkömmliche Literatur zu Requirements Engineering spricht hier von Kreativitätstechniken. Diese Techniken zielen darauf ab, Ideen zu erzeugen, um Lösungen für eine Problemstellung, Frage oder ein Ziel zu finden. Bekannte Beispiele solcher Techniken sind Brainstorming [Osbo1979] oder 6-Hut-Denken [DeBo2006]. Bei der Anforderungsermittlung werden Kreativitätstechniken verwendet, um neue oder innovative Anforderungen zu gewinnen, welche häufig begeisternde Faktoren sind.

Außerhalb der Software- und Requirements Engineering-Welt hat sich der weiter gefasste Begriff „Entwurfstechniken“ (engl. design techniques) entwickelt. Entwurfstechniken beinhalten Kreativitätstechniken zur Ideenentwicklung und bieten weiterhin zusätzliche oder kombinierte Techniken, um Ideen auszuarbeiten und weitere Erkenntnisse zu einer bestimmten Idee zu gewinnen [Kuma2013]. Beliebte Techniken zu diesem Zweck sind u. a. die Verwendung von Prototypen (z. B. Mock-up Prototypen), Storyboards und Szenarien.

Voraussetzungen für Kreativität

Kreativität funktioniert nicht auf Knopfdruck, sondern entsteht durch Zufall. Sie kann vor allem dort entstehen, wo alle vier der folgenden Voraussetzungen erfüllt sind [KrSc2017]:

- *Zufall* – und somit Zeit – damit eine Idee entstehen kann.
- *Wissen* zu dem betroffenen Thema, wodurch die Wahrscheinlichkeit steigt, dass eine wirklich relevante Idee entsteht.
- *Motivation*, da unser Gehirn nur kreativ sein kann, wenn für seinen Eigentümer ein direkter Nutzen daraus entsteht.
- *Sicherheit*, da nutzlose Ideen keine negativen Auswirkungen haben dürfen.

Ideenfindungs- und Entwurfstechniken helfen in einigen oder auch allen diesen Aspekten dabei, eine geeignete Umgebung zu schaffen, damit neue und innovative Ideen entstehen können.

3.2.1 Brainstorming (K3)

Brainstorming wurde in der Zeit der 40er und 50er Jahre des 20. Jahrhunderts durch Alex F. Osborn [Osbo1979] entwickelt. Wie bei den meisten Kreativitätstechniken, ist der entscheidende Aspekt des Brainstormings die Trennung der Ideenfindung von der Analyse der Ideen. Es wird in Gruppen von etwa 5–10 Personen durchgeführt und dauert etwa 20 Minuten. Ein Moderator stellt die ordnungsgemäße Durchführung des Brainstormings und die Dokumentation der Ideen sicher.

Über die Jahre sind viele verschiedene Varianten des Brainstormings entstanden, z. B. Brainstorming Paradox, die 6–3–5-Methode und das Brainwriting.

3.2.2 Analogietechniken (K2)

Analogietechniken (z. B. Basissoziation [Koes1964]) sind Techniken, die dabei unterstützen, Ideen für kritische oder auch komplexe Themen zu entwickeln. Sie nutzen Analogien, um das Denken und das Finden von Ideen zu unterstützen. Erfolg oder Misserfolg werden dabei vor allem durch die Qualität der Analogie beeinflusst. Die Relevanz ähnlicher Systeme ist in 2.4 behandelt.

3.2.3 Prototyping (K3)

Der Begriff Prototyping ist ein Überbegriff, der die Erzeugung verschiedener Arten früherer Muster oder Modelle zum Gewinnen konkreter Erfahrungen mit einem Konzept oder Prozess beschreibt.

Im Kontext der Anforderungsermittlung bezieht sich der Begriff Prototyp bewusst nicht allein auf die Implementierung eines Softwareprototypen. Stattdessen bezieht er sich auf alles, was Anforderungen an ein zu entwickelndes System repräsentieren kann (z. B. Skizze einer Benutzungsoberfläche, ein Physikalischer Mock-up-Prototyp oder ein Video).

Der Zweck der Erstellung eines Prototypen in der Anforderungsermittlung ist die Simulation eines neuen Systems und die Auseinandersetzung mit dessen Anforderungen, indem Zustimmung und Widerspruch oder Klärung und Ergänzung angeregt wird.

Ein Prototyp kann durch ein User Walkthrough [ShRP2007] oder einen Usability Test [RuCh2008] evaluiert werden. Das Ergebnis einer solchen Evaluation sind häufig ein Satz neuer Anforderungen.

3.2.4 Verwendung von Szenarien und Storyboards (K3)

Anwendungsszenarien beschreiben in Form realistischer Beispiele, wie ein Nutzer mit dem zu entwickelnden System interagieren wird [RoCa2002].

Ein Storyboard ist ein visualisiertes Szenario. Es sieht aus wie ein Comic, mit verschiedenen Bildern und/oder Screenshots, und visualisiert auf diese Weise, wie ein System oder Produkt verwendet werden soll. Die Auseinandersetzung mit einem konkreten Beispiel ermöglicht es Kunden und Anwendern, sich Anforderungen in der späteren Anwendungssituation vorzustellen und sie auf diese Weise zu prüfen und zu ergänzen [RiFi2014].

3.3 Denkwerkzeuge (K2)

Die bisher vorgestellten Ermittlungstechniken beschreiben ein bestimmtes Vorgehen, um Informationen zu erheben, oder ein bestimmtes Arbeitsergebnis zum Zweck der Anforderungsermittlung zu erstellen.

In diesem Abschnitt werden Techniken vorgestellt, die einen Querschnitt über diese Techniken bilden, indem sie eine bestimmte Art zu denken begünstigen. Wir nennen sie unterstützende Techniken und Denkwerkzeuge, da sie nicht alleinstehend, sondern in Kombination mit anderen Techniken angewendet werden.

3.3.1 In Abstraktionsebenen denken (K2)

Abstraktionsebenen sind ein hervorragendes Denkwerkzeug für die Anforderungsermittlung [GoWo2005], [Laue2014]. Sie werden häufig als eine Art Prozessmodell verwendet, um die Ermittlungsarbeit zu strukturieren, d. h. zunächst lediglich die Anforderungen auf der obersten Ebene zu ermitteln, um dann mit tieferen Ebenen fortzufahren. Weiterhin können sie verwendet werden, um erhaltene Anforderungsinformationen zu strukturieren, Lücken in Anforderungen oder unnötige Anforderungen zu identifizieren und um Ermittlungsaktivitäten auf eine bestimmte Abstraktionsebene zu konzentrieren. So ist es beispielsweise in einem Workshop mit Anwendern ratsam, sich auf den Systemkontext zu konzentrieren, da die Anwender Experten für den Systemkontext sind. Mit ihnen über die Datenstrukturen eines Systems zu sprechen, ist wohl eher nicht angebracht, da für die Anwender die internen Datenstrukturen nicht von Interesse sind.

3.3.2 In Problemen und Zielen denken (K2)

In Problemen und Zielen zu denken, ist eine Kernkompetenz des Requirements Engineers.

Ein *Problem* ist "eine Schwierigkeit, eine offene Frage oder ein unerwünschter Zustand, der untersucht, erwogen oder gelöst werden muss" [Glin2024]; mit anderen Worten, der Zustand eines bestimmten Aspekts im Kontext eines Stakeholders, der als negativ erlebt wird. Ein Problem kann in der Gegenwart existieren (ein gegenwärtiges Problem).

Ein *Ziel* ist "ein gewünschter Zustand (den ein Stakeholder erreichen möchte)" [Glin2024]: der Zustand eines bestimmten Aspekts im Kontext eines Stakeholders, von dem erwartet wird, dass er positiv ist. Ein Ziel existiert nur in der Zukunft.

Probleme und Ziele existieren nicht in der Realität: Sie sind mentale Konstrukte des Stakeholders. Derselbe Aspekt kann für den einen Stakeholder ein Problem und für den anderen Stakeholder ein Ziel sein. Probleme und Ziele können nur durch Kommunikation mit den betreffenden Stakeholdern aufgedeckt werden.

Probleme und Ziele sind durch ein weiteres mentales Konstrukt miteinander verbunden: Die *Lösung* ist eine Strategie für eine bestimmte Intervention im Kontext des Stakeholders. In der Regel kann es (zu einem gewissen Grad) mehr als eine Lösung für ein Problem oder zur Erreichung eines Ziels geben. Weiterführende Informationen dazu bietet [LoLS2017].

In Problemen und Zielen zu denken, ermöglicht es dem Requirements Engineer, das gesamte Netzwerk aus Problemen, Lösungen und Zielen zu analysieren und aufzudecken.

In der Literatur finden sich diverse Ansätze, die sich mit Problemen auseinandersetzen, z. B. Problemrahmen [Jack2001], oder Ziele, z. B. KAOS [Lams2009].

3.3.3 Vermeidung von Transformationseffekten (K2)

Im CPRE Foundation Level Lehrplan [IREB2022], werden Transformationseffekte im Kontext der Ergebnisse des Requirements Engineerings, also der Dokumentation, diskutiert.

Requirements Engineers sollten sich allerdings auch auf Seiten der eingehenden Informationen dieser (und weiterer) Transformationseffekte bewusst sein, da sie regelmäßig während Ermittlungsaktivitäten bei der Kommunikation mit Stakeholdern oder beim Lesen von Dokumenten vorkommen. Sobald man solchen Effekten begegnet, ist dies ein Anzeichen, um weitere Ermittlungstätigkeiten durchzuführen, die wahrscheinlich zusätzliche oder genauere Anforderungen aufdecken.

3.3.4 Modelle als Denkwerkzeug (K2)

Der CPRE Foundation Level Lehrplan [IREB2022] vermittelt verschiedene Arten von Modellen (z. B. Klassendiagramme, Aktivitätsdiagramme), um Anforderungen zu dokumentieren. Modelle erlauben es, sich auf eine bestimmte Perspektive des Systems zu konzentrieren: Daten, Funktionen oder Verhalten.

Modelle können ebenso als Denkwerkzeug genutzt werden, wenn der Requirements Engineer sich auf eine bestimmte Perspektive während einer bestimmten Ermittlungsaktivität konzentrieren möchte, z. B. ein Klassendiagramm in einem Stakeholder-Interview besprechen, oder ein Datenflussdiagramm in einem Workshop mit Stakeholdern entwickeln. Jedoch sollte sich der Requirements Engineer darüber im Klaren sein, dass Modelle nur dann nützlich sind, wenn die Modellierungssprache von allen betroffenen Stakeholdern verstanden wird.

3.3.5 Mind-Mapping (K3)

Mind-Mapping ist ein graphisches Denkwerkzeug [Buza1993]. Indem das Hauptthema in das Zentrum gesetzt wird und die Ideen in Zweigen von diesem Thema weg verzweigen, können Gedanken und Ideen sortiert und strukturiert werden. Es sollten Texte und Bilder sowie Farben verwendet werden. Die Darstellung für das Gehirn wird „anregender“, wenn man „langweilige“ Darstellungen (gerade, einfarbige Linien) vermeidet.

3.4 Beispiel für die Strukturierung von Ermittlungstechniken: Attribute (K2)

Requirements Engineers sollten, abhängig von dem konkreten Kontext und Bedarf der vorliegenden Situation, die anzuwendende Ermittlungstechnik mit Bedacht wählen. Um diese Auswahl zu unterstützen, kann eine Klassifizierung der Techniken mittels bestimmter Attribute vorgenommen werden. Ein Beispiel für sinnvolle Attribute zeigt Tabelle 1.

Die konkrete Ausprägung einer Ermittlungstechnik kann durch eine Kombination dieser Attribute beschrieben werden.

Beispielsweise kann die Technik *Interview* durch die Attribute „*dialogorientiert*“ und „*fragend*“ beschrieben werden. Ein Interview könnte auch „*beobachtend*“ sein, sofern der Requirements Engineer es in der Anwendungsumgebung eines zukünftigen Endanwenders durchführt.

Doch ist „*beobachtend*“ keine Kerneigenschaft des Interviews, da es auch telefonisch oder in einer anderen Umgebung durchgeführt werden kann, in der keine relevanten Beobachtungen möglich sind.

Tabelle 1 beschreibt relevante Attribute. Der Requirements Engineer sollte die Verfügbarkeit und Eigenschaften von Stakeholdern, Kundenbedürfnisse, die Projektziele und – randbedingungen, das Fachgebiet und den Kontext in dem er arbeitet (siehe 1.3) bei der Auswahl der Ermittlungstechnik in Betracht ziehen. Eine lange Liste verfügbarer Techniken mittels relevanter Attribute zu klassifizieren, kann helfen, die einzusetzenden Techniken für eine bestimmte Situation auszuwählen. „*Es gibt 'Good Practices' in einem bestimmten Kontext, doch es existieren keine 'Best Practices'*“ [KaBa2012]: Jede Anwendungssituation kann eine bestimmte Kombination von Techniken erfordern, um erfolgreich zu sein.

Tabelle 1 Attribute zur Klassifizierung von Ermittlungstechniken.

Attribut	Kurze Beschreibung	Geeignet, um folgende Ziele zu erreichen	Geeignet in folgenden Situationen
Dialogorientiert	Ein Dialog zwischen Requirements Engineer und Stakeholder(n)	Um den Systemkontext zu verstehen; um Ziele zu ermitteln und einen Überblick über die Leistungsfaktoren (Kano) zu erhalten	Wenn (relevante) Stakeholder für einen mündlichen Informationsaustausch verfügbar sind
Fragend	Stakeholdern (zumindest teilweise) vorbereitete Fragen stellen, um Fakten oder ihre Meinung zu erfahren	Um Ziele und Leistungsfaktoren zu ermitteln; um Basisfaktoren zu verifizieren; um die Meinung von Stakeholdern oder zusätzliche Informationen zu zuvor ermittelten Anforderungen zu erhalten; um detaillierte Informationen zu ermitteln; um bestimmte Anforderungen zu klären;	Wenn relevante Fragen vorab formuliert werden können; wenn irgendeine Art von Kommunikation mit Stakeholdern möglich ist; wenn ein komplizierter Sachverhalt gegeben ist;

Attribut	Kurze Beschreibung	Geeignet, um folgende Ziele zu erreichen	Geeignet in folgenden Situationen
Beobachtend	Das Verhalten von Stakeholdern in einer realen Situation zu beobachten, gewöhnlich bei der Anwendung eines existierenden Systems oder bei der Durchführung bestimmter Aufgaben	Um Informationen über das tatsächliche Verhalten eines Stakeholders zu erhalten; um Basisfaktoren zu ermitteln; um Usability-Anforderungen zu ermitteln; um Daten über das Umfeld des Anwenders zu sammeln.	Wenn ein Stakeholder nicht direkt kontaktiert werden kann oder falls er nicht in der Lage ist, seine Bedürfnisse und Aktionen (detailliert genug) zu beschreiben; wenn Zweifel an der Kongruenz zwischen tatsächlicher und beschriebener Situation bestehen; um die Anwenderbedürfnisse besser zu verstehen; um das Projekt besser zu verstehen (z. B. in Vorbereitung auf weitere Ermittlungstechniken);
Zustimmung oder Widerspruch hervorrufend	Relevante Aspekte einer Lösungsmöglichkeit demonstrieren, um Zustimmung oder Widerspruch des Stakeholders zu erhalten.	Um Anforderungen für Stakeholder greifbar zu machen; um zuvor ermittelte Anforderungen zu prüfen; um Feedback zu unterschiedlichen Lösungsvarianten zu erhalten	Wenn Stakeholder Schwierigkeiten haben, sich Dinge vorzustellen; wenn der Requirements Engineer den Stakeholdern Aspekte der vorgeschlagenen Lösung beschreiben oder zeigen kann (oder sie sie sogar anwenden lassen kann); wenn Stakeholder Schwierigkeiten damit haben, zu erklären, was sie brauchen;

Attribut	Kurze Beschreibung	Geeignet, um folgende Ziele zu erreichen	Geeignet in folgenden Situationen
Artefaktbasiert	Analyse der Arbeitsergebnisse (z. B. Dokumente, Modelle, Produkte oder eingesetzte Systeme)	Um Anforderungen von bestehenden Arbeitsergebnissen abzuleiten; um Basisfaktoren und Leistungsfaktoren zu ermitteln, insbesondere Randbedingungen	Wenn relevante Arbeitsergebnisse verfügbar sind und Zugriff darauf besteht; um das Projekt und das Fachgebiet besser zu verstehen (z. B. in Vorbereitung auf weitere Ermittlungstechniken); wenn Stakeholder nicht direkt verfügbar sind;
Kreativitätsfördernd	Fördert Kreativität und Innovation.	Um Begeisterungsfaktoren zu erheben; um zu neuen Ansätzen zu gelangen;	Wenn Innovation nötig ist; wenn es keine vorbestimmte Richtung gibt; wenn andere Ansätze fehlschlagen;
Erlebend	Umwelt und Problemraum des zu entwickelnden Systems erfahren	Um Anforderungen aus den tatsächlichen Umständen abzuleiten; um das zu lösende Problem durch Anwender in deren Arbeitsumfeld zu verstehen; um Empathie zu gewinnen;	Wenn Anwender und Usability zentrale Gesichtspunkte des Projekts sind; wenn es möglich ist, Zugang zur tatsächlichen Anwendungsumgebung zu bekommen;

Es gibt auch weitere Möglichkeiten zur Kategorisierung von Ermittlungstechniken, z.B.

- Kanomodell, siehe [IREB2022]
- Design Innovation Process [Kuma2013]

4 Konfliktlösung (K4)

Dauer: 2,0 Stunden

Begriffe: Notwendigkeit, Konsistenz, Vollständigkeit, Machbarkeit, Anforderungskonflikt, sozialer Konflikt

Lernziele

LZ 4.1.1 Den Unterschied zwischen Anforderungskonflikten und sozialen Konflikten verstehen

LZ 4.1.2 Die Konfliktidentifizierung anwenden

LZ 4.2.1 Die Klassifizierung von Konfliktarten anwenden

LZ 4.2.2 Verstehen, welche Konflikte als Requirements Engineer gelöst werden können und welche delegiert werden müssen

LZ 4.3.1 Die Verwendung von Einigung, Kompromiss, Variantenbildung, Abstimmung und Ober-sticht-Unter als Beispiele für Konfliktlösungstechniken anwenden

LZ 4.3.2 Analysieren von Konfliktlösungstechniken um die, basierend auf den Eigenschaften des Konflikts, geeignetste auszuwählen

LZ 4.4.1 Die Dokumentation der Lösung eines Anforderungskonflikts verstehen

Während der Anforderungsermittlung entdeckt, erhebt und entwirft der Requirements Engineer eine große Sammlung von Anforderungen. Ermittlungstechniken allein stellen nicht sicher, dass die Sammlung als Ganzes klar, vollständig, konsistent, eindeutig und ausreichend ist. Die abschließende Sammlung von Anforderungen müssen alle Stakeholder verstehen und ihnen zustimmen, insbesondere jene Anforderungen, die für sie relevant sind. Stimmen einige der Stakeholder nicht zu, so handelt es sich um einen Anforderungskonflikt, der entsprechend aufgelöst werden sollte.

Konfliktlösung im weiteren Sinne besteht aus vier Tätigkeiten:

- Konfliktidentifizierung
- Konfliktanalyse
- Konfliktlösung
- Dokumentation der Konfliktlösung

Die Identifizierung und Analyse von Konflikten ist eine laufende Tätigkeit im Requirements Engineering und Voraussetzung für die Lösung von Konflikten. Sobald ein Anforderungskonflikt identifiziert wurde, sollte der Requirements Engineer Konfliktlösungsaktivitäten initiieren, um eine geeignete Lösungstechnik auszuwählen und das Ergebnis zu dokumentieren.

4.1 Konfliktidentifizierung (K2)

Konflikte sind im Allgemeinen ein Thema der Sozialwissenschaften und werden typischerweise als „soziale Konflikte“ bezeichnet, um zu zeigen, dass Konflikte zwischen Menschen entstehen.

Ein sozialer Konflikt kann wie folgt definiert werden: „... eine Interaktion zwischen Akteuren (Individuen, Gruppen, Organisationen usw.), wobei wenigstens ein Akteur Unvereinbarkeiten im Denken/ Vorstellen/ Wahrnehmen und/oder Fühlen und/oder Wollen mit dem anderen Akteur (den anderen Akteuren) in der Art erlebt, – dass im Realisieren eine Beeinträchtigung durch einen anderen Akteur (die anderen Akteuren) erfolge.“ [Glas2004]

Ein Anforderungskonflikt kann als eine bestimmte Ausprägung eines sozialen Konflikts interpretiert werden und ist wie folgt definiert: „Ein Konflikt im Requirements Engineering (Anforderungskonflikt) ist eine Unvereinbarkeit von Anforderungen, die auf einer widersprüchlichen Wahrnehmung zweier oder mehrerer Stakeholder basiert.“ [RueA2014]. Konflikte können durch verschiedene Indikatoren entdeckt werden. Indikatoren lassen sich sowohl in der Kommunikation als auch in der Dokumentation beobachten.

Häufig anzutreffende Indikatoren in der *Kommunikation* sind:

- Abstreiten
- Gleichgültigkeit
- Pedanterie
- Zerfragen
- Fehlinterpretation
- Verschleierung
- Delegation

Häufig anzutreffende Indikatoren in der *Dokumentation* sind:

- Widersprüchliche Aussagen von Stakeholdern
- Widersprüchliche Ergebnisse aus Dokumenten- und System-Analyse
- Auf Detailebene inkonsistente Anforderungen
- Inkonsistente Verwendung von Begriffen in der Spezifikation

Meist sind Konflikte versteckt und können nur durch die sorgfältige Überwachung dieser Indikatoren aufgedeckt werden. Tritt einer dieser Indikatoren auf, muss dies nicht automatisch heißen, dass auch ein Anforderungskonflikt vorliegt. Doch sollte der Requirements Engineer stets aufmerksam sein. Mittels der meisten Ermittlungsaktivitäten bringt er Stakeholder dazu, klar Position zu beziehen, wodurch in einigen Fällen unerwartete Probleme oder existierende Konflikte aufgedeckt werden.

4.2 Konfliktanalyse (K3)

Sobald ein Konflikt identifiziert wurde, muss der Requirements Engineer klären, ob es sich bei dem identifizierten Konflikt um einen Anforderungskonflikt handelt. Diese Unterscheidung ist wichtig, da die Auflösung eines Anforderungskonflikts in der Hauptverantwortung des Requirements Engineers liegt, während andere Konflikte durch andere Beteiligte gelöst werden müssen (z. B. Projektleiter).

Es hilft dem Requirements Engineer, die Eigenschaften eines Anforderungskonflikts zu analysieren, um dessen Art zu verstehen. Die folgenden Eigenschaften [RueA2014] eines Konflikts können dabei helfen, die Art des Konflikts zu verstehen und eine passende Lösung dafür zu finden:

- Konfliktart
- Konfliktgegenstand
- Betroffene Anforderungen
- Beteiligte Stakeholder
- Konfliktpositionen
- Konfliktursache
- Konflikthistorie und -entwicklung
- Eingetretene Konfliktfolgen
- Konfliktrisiken

Die Konfliktart ist wichtig, um zu entscheiden, ob es sich bei einem vorliegenden Konflikt um einen Anforderungskonflikt handelt. Es lassen sich sechs Konfliktarten unterscheiden (siehe [IREB2022]):

- Sachkonflikt
- Datenkonflikt
- Interessenkonflikt
- Wertekonflikt
- Strukturkonflikt
- Beziehungskonflikt

Die meisten Anforderungskonflikte können als Daten-, Interessen- oder Wertekonflikt kategorisiert werden. Sachkonflikte, falls vorhanden, werden oft in einer frühen Projektphase aufgedeckt. Struktur- und Beziehungskonflikte haben normalerweise keinen direkten Bezug zu Anforderungen und sollten daher durch andere Beteiligte gelöst werden.

Jedoch zeigen die meisten Konflikte Eigenschaften von mehr als einer Art, da verschiedene Ursachen miteinander in Wechselwirkung treten können. Daher sollte der Requirements Engineer allen Konfliktarten Aufmerksamkeit schenken, auch wenn die Lösung nicht in seinem Verantwortungsbereich liegt.

4.3 Konfliktlösung (K4)

Eine Voraussetzung für die Auswahl der passenden Konfliktlösungstechnik ist das tiefgreifende Verständnis des Anforderungskonflikts. Die folgenden allgemeinen Konfliktlösungstechniken werden unterschieden (siehe [IREB2022]):

- Einigung
- Kompromiss
- Abstimmung
- Variantenbildung
- Ober-sticht-Unter

Zusätzlich gibt es verschiedene unterstützende Techniken, zum Beispiel:

- Gewaltfreie Kommunikation [Rose2015]
- Verhandlungstechniken [FiUP2012]
- Consider-All-Facts [DeBo2006]
- Plus-Minus-Interesting [DeBo2006]
- Entscheidungsmatrix [BiAB2006], [IsNe2013]

Ausgehend von den Konflikteigenschaften, sollten passende Konfliktlösungstechniken ausgewählt werden.

4.4 Dokumentation der Konfliktlösung (K2)

Nach der Konfliktauflösung sollte der Konflikt genau dokumentiert werden. Abgesehen von den in 4.2 erwähnten Konflikteigenschaften sollte die Dokumentation folgende Informationen enthalten:

- Annahmen in Bezug auf den Konflikt und seine Auflösung
- Randbedingungen, die die Wahl der Konfliktlösungstechnik oder die Konfliktlösung beeinflussen
- in Betracht gezogene Alternativen
- Konfliktauflösung inklusive der Gründe, die zur Wahl der Konfliktlösungstechnik geführt hat
- Entscheider und weitere Beteiligte

Ohne Dokumentation besteht die Gefahr, dass Stakeholder getroffene Entscheidungen einfach vergessen oder ignorieren oder versuchen, die Entscheidung im Nachhinein zu ändern. Dies geschieht vor allem dann, wenn der Anforderungskonflikt selbst zwar gelöst ist, jedoch ein darunterliegender (weiterer) sozialer Konflikt nicht gelöst ist.

5 Fähigkeiten eines Requirements Engineers (K3)

Dauer: 1,5 Stunden

Begriffe: Fähigkeiten, Kommunikationsmodelle, Interpretation, Reaktion, Selbst-Reflexion

Lernziele

- LZ 5.1.1 Die benötigten Fähigkeiten im Bereich Ermittlung verstehen
- LZ 5.2.1 Die Grundlagen der Kommunikationstheorie verstehen
- LZ 5.3.1 Selbstreflexion zu den eigenen Fähigkeiten in der Anforderungsermittlung anwenden
- LZ 5.4.1 Maßnahmen zur persönlichen Entwicklung verstehen
- LZ 5.5.1 Das Lernen aus vorherigen Erfahrungen verstehen

5.1 Benötigte Fähigkeiten im Bereich Ermittlung (K2)

Der CPRE Foundation Level [IREB2022] beschreibt Kommunikationsfähigkeit, analytisches Denken, Empathie, Konfliktlösungsfähigkeit, Moderationsfähigkeit, Selbstbewusstsein und Überzeugungsfähigkeit als die benötigten Fähigkeiten („Softskills“) eines Requirements Engineers. Zur Ermittlung von Anforderungen auf dem Practitioner-/Specialist-Niveau, sind auch die folgenden Eigenschaften relevant:

- Ichbewusstsein
- Kontextbewusstsein
- Motivierendes Wesen
- Führungspersönlichkeit
- Flexibilität
- Reflexion
- Neutralität
- Interkulturelle Kompetenz
- Ethisches Gewissen

Von all diesen Eigenschaften ist die Kommunikationsfähigkeit die Schlüsseleigenschaft des Requirements Engineers. Jede Interaktion zwischen Requirements Engineer und Stakeholdern, als der Hauptquelle für Anforderungen, ist eine Form von Kommunikation und alle zuvor genannten Fähigkeiten spielen darin eine Rolle.

5.2 Kommunikationstheorie und Kommunikationsmodelle (K2)

Bei Kommunikation geht es darum *aussagekräftige Konzepte* zwischen Individuen zu *teilen*. Während der Kommunikation kann Information verloren gehen, hinzugefügt, verzerrt oder fehlinterpretiert werden. Der Requirements Engineer sollte dafür Sorge tragen, diese Probleme so gut wie möglich zu verhindern.

Die Theorie und Modelle zu verstehen und fähig zu sein, dieses Wissen in die tägliche Kommunikation zu integrieren, verbessert die Kommunikation des Requirements Engineers und führt zu besseren Ergebnissen.

Ein grundlegendes Verständnis der Kommunikationstheorie kann durch das Studium der folgenden Kommunikationsmodelle erlangt werden:

- Das Shannon–Weaver–Modell [ShWe1971]
- Das zirkuläre Kommunikationsmodell [Schr1971]
- Das „vier-Seiten“-Modell nach Schulz von Thun [Schu1981]

5.3 Selbstreflexion zu den eigenen Fähigkeiten in der Anforderungsermittlung (K3)

Der Lehrplan und die darauf aufbauende Schulung legen die Grundlage für eine erfolgreiche Anwendung der vorgestellten Techniken. Die Entwicklung und Verbesserung persönlicher Fähigkeiten in der Anforderungsermittlung ist jedoch ein langfristiger Lernprozess.

Selbst wenn das Requirements Engineering in einem Projekt als Erfolg gewertet wird, gibt es normalerweise verschiedene Möglichkeiten der Verbesserung. Beispiele sind:

- Hat eine Technik die erwarteten Ergebnisse geliefert oder zur Entwicklung beigetragen?
- Haben die Stakeholder die verwendeten Ermittlungs- oder Konfliktlösungstechniken angenommen?
- War der Aufwand für eine gewählte Technik im Vergleich zu ihrem Beitrag zum Projekterfolg gerechtfertigt?
- Welche Technik hätte es erlaubt, Anforderungen, die erst spät im Projekt aufkamen, früher zu erheben?

Die eigenen Fähigkeiten können einerseits durch unmittelbare Beobachtung des eigenen Verhaltens und andererseits durch nachgelagerte Analyse geprüft werden. In der unmittelbaren Beobachtung sollte der Fokus auf eine, maximal zwei Eigenschaften gelegt werden, um ein genaues und zuverlässiges Beobachtungsergebnis zu erhalten (z. B. Beobachtung der eigenen reflektierenden Kommunikation während eines Interviews). Um die eigenen Fähigkeiten in einer nachgelagerten Analyse einzuschätzen, ist die Reaktion anderer eine wichtige Quelle (z. B. 360°-Feedback [LeLu2009]). Ein weiteres sinnvolles Meßinstrument ist ein Einschätzungsbogen in Bezug auf die zuvor festgelegten Fähigkeiten [SmMa2004].

5.4 Möglichkeiten der persönlichen Weiterentwicklung (K2)

Fehlende praktische Erfahrung wird häufig als Grund dafür angeführt, warum eine bestimmte Ermittlungs- oder Konfliktlösungstechnik nicht angewandt wurde. Solch eine Haltung ist zwar verständlich hinsichtlich des Projekterfolgs (der Requirements Engineer wendet die Techniken an, die er am besten kennt, um den Erfolg des Projekts sicher zu stellen); in Bezug auf die persönliche Weiterentwicklung ist diese Haltung jedoch wenig hilfreich. Eine bewährte Alternative ist die Anwendung unbekannter Techniken in einem Rahmen mit geringem Risiko (z. B. Apprenticing mit einer kleinen Teilmenge der Stakeholder durchführen). Weiterhin ist es möglich, eine unbekannte Technik parallel zu einer bekannten Technik einzusetzen (z. B. einen Fragebogen parallel zu mehreren Interviews).

5.5 Aus vorherigen Erfahrungen lernen (K2)

Grundlegende Bestandteile des persönlichen Weiterbildungsprozesses, der Lernen aus vorherigen Erfahrungen fördert, sind:

- Verbesserung in der alltäglichen Arbeit
- Regelmäßiges Messen des eigenen Fähigkeitsprofils
- Schulungsmaßnahmen
- Mentoring-Maßnahmen

6 Literaturverzeichnis und weitere Lektüre

- [AlIS1977] C. Alexander, S. Ishikawa, M. Silverstein: A Pattern Language – Towns – Buildings – Construction. Oxford University Press, New York, 1977.
- [Alex2005] I.F. Alexander: A Taxonomy of Stakeholders – Human Roles in System Development. International Journal of Technology and Human Interaction, Vol 1, 1: 23–59, Hershey, 2005.
- [AlBe2009] I.F. Alexander, Ljerka Beus–Dukic: Discovering requirements: how to specify products and services. John Wiley & Sons Ltd, Hoboken, 2009.
- [BaCC2015] K. Baxter, C. Courage, K. Caine: Understanding Your Users: A Practical Guide to User Research Methods, 2. Auflage. Morgan Kaufmann, Burlington, 2015.
- [BaGr1975] R. Bandler, J. Grinder: Metasprache und Psychotherapie: Die Struktur der Magie I. Junfermann Verlag, Paderborn, 2011.
- [BaGr1976] R. Bandler, J. Grinder: Kommunikation und Veränderung: Die Struktur der Magie II, Junfermann Verlag, Paderborn, 2010.
- [BeHo1998] H. Beyer, K. Holtzblatt: Contextual Design: Defining Customer–Centered Systems. Morgan Kaufmann, Burlington, 1998.
- [Beve1957] W. I. B. Beveridge: The Art of Scientific Investigation. The Blackburn Press, Cambridge, 1957. Der vollständige Text dieses Buches ist verfügbar auf: www.archive.org Zuletzt besucht im Februar 2019.
- [BiAB2006] S. Biffi, A. Aurum, B. Boehm: Value–Based Software Engineering. Springer–Verlag, Berlin, 2006.
- [Bour2009] L. Bourne: Stakeholder Relationship Management: A Maturity Model for Organizational Implementation. Gower Publishing Ltd, Burlington, 2009.
- [Bour2015] L. Bourne: Making Projects Work: Effective Stakeholder and Communication Management. CRC Press, Boca Raton, 2015.
- [Buza1993] T. Buzan: Das Mind–Map–Buch: Die beste Methode zur Steigerung Ihres geistigen Potenzials. mvg Verlag, München, 2013.
- [CaGr2012] E. Cameron, M. Green: Making Sense of Change Management: A Complete Guide to the Models Tools and Techniques of Organizational Change. 3. Auflage, Kogan Page, London, 2012.
- [Cock2001] A. Cockburn: Use Cases effektiv erstellen. mitp, Heidelberg, 2003.
- [CRCN2014] A. Cooper, R. Reimann, D. Cronin, C. Noessel: About Face: The Essentials of Interaction Design. 4. Auflage, John Wiley & Sons, Inc, Hoboken, 2014.
- [Cohn2004] M. Cohn: User Stories: für die agile Software–Entwicklung mit Scrum, XP u.a. mitp, Heidelberg, 2010.

- [Coop2004] A. Cooper: The Inmates are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity. Que, Indianapolis, 2004.
- [CoSh2007] T. Colburn, G. Shute: Abstraction in Computer Science. Minds & Machines, Vol. 17, S. 169–184, Springer Science+Business Media B.V., Dordrecht, 2007.
- [CrOB2006] O. Creighton, M. Ott, B. Bruegge: Software Cinema: Video-based Requirements Engineering – 14th IEEE International Requirements Engineering Conference. IEEE Computer Society, Washington DC, 2006.
- [DeBo2006] E. de Bono: De Bonos neue Denkschule: Kreativer Denken, Effektiver Arbeiten, Mehr Erreichen. mvg Verlag, München, 2010.
- [GrDA2015] Eduard C. Groen, J. Doerr, S. Adam: Towards Crowd-Based Requirements Engineering – A Research Preview– Requirements Engineering– Foundation for Software Quality – 21st International Working Conference – REFSQ 2015. Essen, Deutschland, 23.–26. März 2015. Proceedings. S. 247–253., Cham, 2015.
- [FiUP2012] R. Fisher, W. Ury, B. Patton: Das Harvard-Konzept: Die unschlagbare Methode für beste Verhandlungsergebnisse, Deutsche Verlags-Anstalt, München, 2018.
- [Glas1982] F. Glas: The process of conflict escalation and roles of third parties. In: G.B.J. Bomers and R.B. Peterson, (eds) Conflict management and industrial relations, Springer-Science+Business Media, Dordrecht, 1982.
- [Glas2004] F. Glas: Konfliktmanagement: Ein Handbuch für Führungskräfte, Beraterinnen und Berater. 11. Auflage, Freies Geistesleben, Stuttgart, 2004.
- [Glin2024] M. Glinz: A Glossary of Requirements Engineering Terminology (Version 2.1.0). IREB e.V., Karlsruhe, 2021. <https://www.ireb.org/de/downloads/#cpre-glossary>. Zuletzt besucht im Februar 2024.
- [Gott2002] E. Gottesdiener: Requirements by Collaboration: Workshops for Defining Needs, Addison-Wesley Professional, Boston, 2002.
- [GoWo2005] T. Gorschek, C. Wohlin: Requirements Abstraction Model. Requirements Engineering Journal Vol. 11, No. 1, S. 79–101. <http://dx.doi.org/10.1007/s00766-005-0020-7>, 2005. Zuletzt besucht im Februar 2019.
- [Harr2014] D. F. Harris: The Complete Guide to Writing Questionnaires– How to Get Better Information for Better Decisions, I&M Press, North Carolina, 2014.
- [HoDC2007] P. Holman, T. Devane, S. Cady: Change Handbook. Zukunftsorientierte Großgruppen-Methoden. Carl-Auer Verlag GmbH, Heidelberg, 2006.
- [IGKCD2002] A. Iles, D. Glaser, M. Kam, J. Cannyand, E. Do: Learning via Distributed Dialogue: Live Notes and Handheld Wireless Technology. Proc. Conf. Computer Support for Collaborative Learning, Hillsdale, 2002.
- [IREB2022] IREB Certified Professional for Requirements Engineering Foundation Level Syllabus (version 3.1.0). IREB e.V., Karlsruhe, 2022.

<https://www.ireb.org/de/downloads/#cppe-foundation-level-syllabus-3-0>.

Zuletzt besucht im Februar 2024.

- [IsNe2013] A. Ishizaka, P. Nemery: Multi-criteria decision analysis. Methods and software. John Wiley & Sons, Ltd, Chichester, 2013.
- [ISO29148] ISO/IEC/IEEE29148:2011. Systems and software engineering – Life cycle processes – Requirements engineering, International Organization for Standardization, Geneva, 2011.
- [ISO9241-210] ISO9241-210:2010. Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems, International Organization for Standardization, Geneva, 2010.
- [Jack2001] M. Jackson: Problem Frames – Analyzing and structuring software development problems. Addison-Wesley, Boston, 2001.
- [KaBa2012] C. Kaner, J. Bach: The Seven Basic Principles of the Context-Driven School. <http://context-driven-testing.com/>, 2012.
- [Kell1984] J. F. Kelley: An iterative design methodology for user friendly natural language in office information applications. ACM Transactions on Office Information Systems, March1984, 2:1, 26–41. New York, 1984.
- [Kell2002] H. Kellner: Kreativität im Projekt. Hanser Fachbuch, München, 2002.
- [Koes1964] A. Koestler: The Act of Creation. Penguin Books, London, 1964.
- [KoSo1998] G. Kotonya, I. Sommerville: Requirements Engineering: Processes and Techniques. Wiley Publishing, Hoboken, 1998.
- [KrSc2017] I. Kreß, A. Schwarz: To Brainstorm or Not to Brainstorm – Neuropsychological Insights on Creativity. Requirements Engineering Magazine Vol. 2017–02. <https://re-magazine.ireb.org/issues/2017-02-staying-on-the-right-path/to-brainstorm-or-not-to-brainstorm-neuropsychological-insights-on-creativity/>. Zuletzt besucht im Februar 2019.
- [Kuma2013] V. Kumar: 101 Design Methods – A Structured Approach for Driving Innovation in Your Organization. Wiley, 2013.
- [Kuni2003] M. Kuniavsky: Observing the User Experience: A Practitioners’s Guide to User Research, Morgan Kaufmann, Burlington, 2003.
- [Lams2009] A. van Lamsweerde: Requirements Engineering: From System Goals to UML Models to Software Specifications. JohnWiley&Sons, Hoboken, 2009.
- [Laue2014] K. Lauenroth: What does it mean to say “requirement”? – An inquiry into the abilities of the human mind and the meaning of the word “requirement”. Requirements Engineering Magazine Vol. 2014–01. <http://re-magazine.ireb.org/issues/2014-1-learning-to-fly/what-does-it-mean-to-say-requirement>, 2014. Zuletzt besucht im Februar 2019.

- [Leff2011] D. Leffingwell: Agile Software Requirements. Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison–Wesley, Boston, 2011.
- [LeLu2009] R. Lepsinger, A.D. Lucia: The Art and Science of 360 Degree Feedback. 2. Auflage, Wiley, San Francisco, 2009.
- [LeWi2003] D. Leffingwell, D. Widrig: Managing Software Requirements: A Use Case Approach. Addison–Wesley, Boston, 2003.
- [LoLS2017] H. van Loenhoud, K. Lauenroth, P. Steiger: The goal is to solve the problem – Some thoughts on problems and goals in the context of Requirements Engineering. Requirements Engineering Magazine Vol. 2017–02. <http://re-magazine.ireb.org/issues/2017-02-staying-on-the-right-path/the-goal-is-to-solve-the-problem/>. Zuletzt besucht im Februar 2019.
- [MNJR2015] W. Maalej, M. Nayebi, T. Johann, G. Ruhe: Toward Data–Driven Requirements Engineering. IEEE Software Vol. 33, No. 1, S.48–54, 2015.
- [MaGi2001] N. Maiden, A. Gizikis: Where Do Requirements Come From? IEEE Software Vol. 18, 5: 10–12, Geneva, 2001.
- [Mayh1999] D. J. Mayhew: The Usability Engineering Lifecycle. Morgan Kaufmann, Burlington, 1999.
- [McCo2006] S. McConnell: Software Estimation – Demystifying the Black Art. Microsoft Press, Redmond, 2006.
- [Moor2014] C. W. Moore: The Mediation Process – Practical Strategies for Resolving Conflicts. 4. Auflage, John Wiley & Sons, Hoboken, 2014.
- [Niel1993] J. Nielsen: Usability Engineering. Nielsen: Usability Engineering. Morgan Kaufmann, Burlington, 1993.
- [Osbo1979] A. F. Osborn: Applied Imagination. 3. überarb. Auflage, Charles Scribner’s Sons, New York, 1979.
- [Parn1972] D.L. Parnas: On the Criteria To Be Used in Decomposing Systems into Modules (PDF). Communications of the ACM 15 (12), S. 1053–58. doi:10.1145/361598.361623, ACM New York, 1972.
- [Pohl2010] K. Pohl: Requirements Engineering: Grundlagen, Prizipien, Techniken. dpunkt.Verlag GmbH, Heidelberg, 2008.
- [PoRu2015] K. Pohl, C. Rupp: Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB–Standard zum Certified Professional for Requirements Engineering Foundation Level, dpunkt Verlag, Heidelberg, 2015.
- [Port2013] S. Portigal: Interviewing Users: How to Uncover Compelling Insights. Rosenfeld Media, Brooklyn, 2013.
- [Reif 2012] D.J. Reifer: Software Change Management: Case Studies and Practical Advice. Microsoft Press, Redmond, 2012.

- [RiFl2014] M. Richter, M. Flückiger: Usability und UX kompakt: Produkte für Menschen. Springer Vieweg, Wiesbaden, 2016.
- [Robs2011] C. Robson: Real World Research. John Wiley & Sons, Hoboken, 2011.
- [RoCa2002] M. Rosson, J. M. Carroll: Usability Engineering: Scenario-Based Development of Human Computer Interaction. Morgan Kaufmann, Burlington, 2002.
- [Rohr1969] B. Rohrbach: Kreativ nach Regeln-Methode 635, eine neue Technik zum Lösen von Problemen. Absatzwirtschaft 12, Heft 19:73-75, Meedia GmbH & Co.KG, Hamburg, 1969.
- [RoRo2013] S. Robertson, J. Robertson: Mastering the Requirements Process: Getting Requirements Right. Dritte Auflage, Pearson Education, London, 2013.
- [Rose2015] M. B. Rosenberg: Gewaltfreie Kommunikation: Eine Sprache des Lebens. Junfermann Verlag, Paderborn, 2016
- [Royc1972] W. Royce: Managing the Development of Large Software Systems. Technical Papers of Western Electronic Show and Convention (WesCon) August 25-28, Los Angeles, 1972.
- [RuCh2008] J. Rubin, D. Chisnell: Handbook of Usability Testing- How to Plan, Design, and Conduct Effective Tests. Wiley; Indianapolis, 2008.
- [RueA2014] C. Rupp, die SOPHISTen: Requirements Engineering und -Management - Aus der Praxis von klassisch bis agil. 6. Aufl., Carl Hanser Verlag, München, 2014. (ausgewählte Kapitel der englischen Version siehe <http://www.sophist.de/en/infopool/downloads/>). Zuletzt besucht im Februar 2019.
- [Schr1971] W. L. Schramm: How communication works. in W. L. Schramm, Ed., The Process and Effects of Mass Communication. Überarb. Auflage, University of Illinois Press, Champaign, 1971.
- [Schu1981] F. Schulz von Thun: Miteinander reden 1 - Störungen und Klärungen. Psychologie der zwischenmenschlichen Kommunikation. Rowohlt, Reinbek, 1981.
- [Shac1991] B. Shackel: Usability - Context, Framework, Definition, Design and Evaluation. In B. Shackel, S. Richardson (Eds.): Human Factors for Informatics Usability (S. 21-37), University Press, Cambridge, UK, 1991.
- [ShRP2007] H. Sharp, Y. Rogers, J. Preece: Interaction Design: Beyond Human-Computer Interaction. John Wiley & Sons, Hoboken, 2007.
- [ShWe1971] C. E. Shannon, W. Weaver: The Mathematical Theory of Communication University of Illinois Press, Champaign, 1971.
- [SmMa2004] S. Smith, R. Mazin: The HR Answer Book: An Indispensable Guide for Managers and Human Resources Professionals, AMACOM, Hertogenbosch, 2004.
- [Wieg2003] K. E. Wiegers: Software-Requirements. Microsoft Press, Redmond, 2005.

