



Certified Professional for Requirements Engineering

Handbuch

Requirements Elicitation

Practitioner | Specialist

Anja Brand, Dominik Häußer,
Patrick Steiger, Michael Tesar

Nutzungsbedingungen

Dieses Handbuch ist einschließlich aller seiner Teile urheberrechtlich geschützt. Die Nutzung dieses Handbuchs ist mit Zustimmung der Rechteinhaber und gemäß geltendem Urheberrecht erlaubt, es sei denn, dies ist ausdrücklich nicht gestattet. Dies gilt insbesondere für Vervielfältigungen, Anpassungen, Übersetzungen, Mikroverfilmung, Speicherung und Verarbeitung in elektronischen Systemen sowie für Veröffentlichungen.

Ausbildungsanbieter dürfen das Handbuch als Grundlage für Seminare verwenden, sofern der Inhaber des Urheberrechts anerkannt und die Quelle und der Besitzer des Urheberrechts genannt werden. Das Handbuch darf zudem mit vorheriger Zustimmung des IREB für Werbezwecke verwendet werden.

Jede Einzelperson oder Gruppe von Einzelpersonen darf das Handbuch als Grundlage für das Studium, für Artikel, Bücher oder andere abgeleitete Veröffentlichungen verwenden, sofern der Inhaber des Urheberrechts anerkannt und die Quelle und der Besitzer des Urheberrechts genannt werden.

Danksagung

Die erste Version dieses Handbuchs wurde verfasst von Dominik Häußer, Kim Lauenroth, Hans van Loenhoud, Anja Brand und Patrick Steiger. Review durch Birgit Penzenstadler. Übersetzung aus dem Englischen ins Deutsche von Tanja Scheuermann. Deutsch-Review durch Marta Tayeh. Die Freigabe der englischen Version wurde am 8. Oktober 2018 durch das IREB Council auf Empfehlung von Thorsten Weyer genehmigt.

Die aktuelle Version wurde am 1. August 2025 vom IREB-Council auf Empfehlung von Norbert Seyff zur Veröffentlichung freigegeben.

Allen sei für ihr Engagement gedankt.

Das Urheberrecht © 2017 – 2025 für dieses Handbuch besitzen die aufgeführten Autoren.

Die Rechte sind übertragen auf das IREB International Requirements Engineering Board e. V.

Die Erstellung dieses Handbuches wurde unterstützt durch



Vorwort

Dieses *Handbuch* ergänzt den Lehrplan für CPRE das Requirements Elicitation Modul.

Es richtet sich an Ausbildungsanbieter, die Seminare oder Schulungen zum CPRE Requirements Elicitation Practitioner und/oder Specialist gemäß IREB-Standard anbieten möchten. Zur weiteren Zielgruppe zählen Schulungsteilnehmer und interessierte Anwender, die einen detaillierten Einblick in den Inhalt dieses Moduls erhalten möchten.

Das Handbuch ist kein Ersatz für Schulungen zu diesem Thema. Es stellt ein Bindeglied zwischen dem Lehrplan (in dem die Lernziele des Moduls aufgeführt und erläutert werden) und der umfangreichen Literatur dar, die zu diesem Thema veröffentlicht wurde.

Ausbildungsanbieter können den Inhalt dieses Handbuchs sowie die Verweise auf weiterführende Literatur zur Vorbereitung von Teilnehmern auf die Zertifizierungsprüfung nutzen. Das Handbuch bietet Schulungsteilnehmern und interessierten Anwendern eine Möglichkeit, ihre Kenntnisse über Requirements Engineering in einer agilen Umgebung zu

vertiefen und den detaillierten Inhalt durch die empfohlene Literatur zu ergänzen. Darüber hinaus kann das Handbuch zur Auffrischung bereits vorhandener Kenntnisse in den verschiedenen Themen der Anforderungsermittlung genutzt werden, beispielsweise nachdem das Zertifikat Requirements Elicitation Practitioner oder Specialist erworben wurde.



Die Entwicklung von Künstlicher Intelligenz (KI) eröffnet spannende Ergänzungspotenziale für das Requirements Engineering. Moderne KI-Methoden wie Data-Mining, Natural Language Processing und Machine Learning können den Erhebungs- und Analyseprozess unterstützen, indem sie große Datenmengen aus verschiedenen Quellen effizient auswerten und so wertvolle Einblicke in Muster und Trends liefern. Dennoch bleibt die Rolle des Requirements Engineers zentral und unverzichtbar, denn die methodische Planung, das kritische Hinterfragen von Anforderungen sowie der zwischenmenschliche Austausch und die Kontextualisierung von Ergebnissen erfordern nach wie vor das fundierte Fachwissen und die Erfahrung des Menschen. KI dient in diesem Kontext als ergänzendes Werkzeug, das den Requirements Engineer dabei befähigt, fundierte Entscheidungen zu treffen und innovativ auf die Herausforderungen komplexer Projekte zu reagieren, ohne dabei den essenziellen menschlichen Input zu ersetzen.

Die Fortschritte im Bereich der KI werden immer schneller. Die IREB Special Interest Group Artificial Intelligence (SIGAI) setzt sich daher intensiv mit dem Zusammenspiel von KI und Requirements Engineering auseinander. Ihre Erkenntnisse werden schrittweise in das Modul CPRE Requirements Elicitation aufgenommen. In dieser Fassung des Handbuchs wurden Info-Boxen ergänzt, die den möglichen Einsatz von KI in den jeweiligen Kapiteln aufzeigen. Diese optionalen Abschnitte enthalten praxisnahe Beispiele und Ideen, wie KI das Requirements Engineering unterstützen kann. Gleichzeitig bleibt die Rolle des Requirements Engineers mit seinem Methodenwissen nach wie vor essenziell. KI bietet neue Möglichkeiten, ersetzt jedoch nicht die Erfahrung und Kompetenz des Menschen.

Konkrete Anwendungsbeispiele sind u.a. dem SIG #AIREB Booklet zu entnehmen, verfügbar auf der [Webseite der SIG #AIREB](#).

Über Vorschläge für Verbesserungen oder Korrekturen freuen wir uns!

E-Mail-Kontakt: info@ireb.org

Wir wünschen Ihnen viel Freude beim Studium dieses Handbuchs und einen erfolgreichen Abschluss der Zertifizierungsprüfung zum IREB Requirements Elicitation Practitioner oder Specialist.

Weitere Informationen zum IREB CPRE Requirements Elicitation finden Sie unter:

<http://www.ireb.org>

Versionshistorie

Version	Datum	Kommentar	Autor
1.0.0	2. Juli 2019	Erste deutsche Version, basierend auf der englischen Version 1.0.2.	Dominik Häußer, Kim Lauenroth, Hans van Loenhoud, Anja Brand, Patrick Steiger
1.0.1	15. April 2019	SOPHIST-Logo und Abspann auf Englisch hinzugefügt. Rechtschreibung korrigiert	Stefan Sturm
1.0.2	2. Juli 2019	Rechtschreibung korrigiert Größe der Zahlen angepasst	Ruth Rossi, Stefan Sturm
1.0.3	12. September 2019	Falscher Verweis in Kapitel 3.1.3.3 [Robles2012] ersetzt durch [RoRo2013]. Fehlerhafter Link in [TiSi2017] behoben.	Stefan Sturm
1.1.0	1. Januar 2022	Aktualisierungen auf der Grundlage von CRPE Foundation Level V3 und CPRE Glossary V2 <ul style="list-style-type: none"> ▪ Abschnitt 3.1: Die Klassifizierung der Techniken zur Anforderungsermittlung wurde um "Kollaborationstechniken" erweitert. ▪ Abschnitt 4.1: Aktualisierung des Konfliktindikators "Abstreiten". ▪ Abschnitt 4.2.2: Aktualisierte Klassifizierung der Konfliktarten. ▪ Allgemeines: Geringfügige Aktualisierungen des Wortlauts in Bezug auf. CPRE Foundation Level V3 und CPRE Glossar V2. ▪ Dieses Handbuch ist auf den CPRE AL Elicitation Syllabus V2.1.0 abgestimmt. 	Dominik Häußer, Kim Lauenroth, Hans van Loenhoud, Anja Brand, Patrick Steiger

Version	Datum	Kommentar	Autor
2.0.0	1. Juli 2022	Berücksichtigung des Advanced Level Splits in Practitioner und Specialist	Stefan Sturm
2.1.0	1. März 2025	Neues Corporate Design umgesetzt, Anzahl der Fragetechniken in 3.1.1 gefixt. Begriff "Artefakte" durch "Arbeitsprodukte" ersetzt.	Stefan Sturm
2.2.0	August 12, 2025	AI-Infoboxen hinzugefügt, fehlerhafte Verweise korrigiert, Tippfehler beseitigt, Qualität der Abbildungen 5, 17, 22, 26, 27, 28, 29 und 30 verbessert	Dominik Häußer, Anja Brand, Patrick Steiger, Michael Tesar
2.2.1	27. November 2025	Einleitung zu Kapitel 2.4 korrigiert. Übersetzungsfehler, Systeme statt Dokumente	Stefan Sturm

Inhalt

1	Konzepte zur strukturierten Planung und Durchführung der Anforderungsermittlung	9
1.1	Der Anwendungsbereich der Ermittlung im Requirements-Engineering ..	10
1.2	Relevante Einflussfaktoren für das Vorgehen zur Planung von Ermittlungsaktivitäten	11
1.3	Planung und Durchführung der Anforderungsermittlung	12
1.3.1	Ermittlungsaktivität	13
1.3.2	Konfliktlösungsaktivität	18
1.3.3	Richtlinien für den Ermittlungsteil eines Projekts	20
1.4	Prozessmuster	28
1.4.1	Struktur und Vorteile von Prozessmustern für die Anforderungsermittlung	29
1.4.2	Das Wasserfall-Modell	30
1.4.3	Human-Centered Design	36
1.4.4	Design Thinking	42
2	Anforderungsquellen	48
2.1	Grundlagen zu Anforderungsquellen	48
2.2	Stakeholder identifizieren, klassifizieren und verwalten	51
2.2.1	Stakeholder als Anforderungsquelle identifizieren und auswählen	52
2.2.2	Stakeholder-Relationship-Management	55
2.2.3	Dokumentationsschema für einbezogene Stakeholder	58
2.2.4	Der Nutzer als besondere Stakeholder-Gruppe	59
2.3	Dokumente identifizieren, klassifizieren und verwalten	60
2.3.1	Dokumente als Anforderungsquelle identifizieren und auswählen	60
2.3.2	Dokumentationsschema für Dokumente	64
2.4	Systeme identifizieren, klassifizieren und verwalten	65
2.4.1	Systeme als Anforderungsquelle identifizieren und auswählen	65
2.4.2	Dokumentationsschema für Systeme	68
3	Anforderungsermittlung	70

3.1	Erhebungstechniken	72
3.1.1	Fragetechniken	72
3.1.2	Beobachtungstechniken	84
3.1.3	Kollaborationstechniken	95
3.1.4	Artefaktbasierte Techniken	103
3.2	Entwurfs- und Ideenfindungstechniken (K2)	112
3.2.1	Brainstorming	113
3.2.2	Analogietechnik	115
3.2.3	Prototyping	117
3.2.4	Verwendung von Szenarien und Storyboards	122
3.3	Denkwerkzeuge	125
3.3.1	In Abstraktionsebenen denken	126
3.3.2	In Problemen und Zielen denken	128
3.3.3	Vermeidung von Transformationseffekten	132
3.3.4	Modelle als Denkwerkzeug	138
3.3.5	Mind-Mapping	139
3.4	Ermittlungstechniken mit Attributen beschreiben	141
4	Konfliktlösung	151
4.1	Konfliktidentifizierung	152
4.2	Konfliktanalyse	153
4.2.1	Eigenschaften von Anforderungskonflikten	154
4.2.2	Die Konfliktarten nach Moore	155
4.3	Konfliktlösung	160
4.3.1	Einigung	160
4.3.2	Kompromiss	162
4.3.3	Abstimmung	163
4.3.4	Variantenbildung	165
4.3.5	Ober-sticht-Unter	166
4.3.6	Unterstützende Techniken	167
4.3.7	Wahl einer geeigneten Konfliktlösungstechnik	168
4.4	Dokumentation der Konfliktlösung	172
5	Fähigkeiten eines Requirements Engineers	173

5.1	Benötigte Fähigkeiten im Bereich Ermittlung	173
5.2	Kommunikationstheorie und Kommunikationsmodelle	175
5.3	Selbstreflexion zu den eigenen Fähigkeiten in der Anforderungsermittlung	179
5.4	Möglichkeiten der persönlichen Weiterentwicklung	182
5.5	Von vorherigen Erfahrungen lernen - lebenslanges Lernen	183
6	Literaturverzeichnis und weitere Lektüre	185

1 Konzepte zur strukturierten Planung und Durchführung der Anforderungsermittlung

Im Laufe der vergangenen Jahre sind IT-Systeme zu einem wesentlichen Faktor für das Funktionieren von Unternehmen, Regierungen und auch der Gesellschaft an sich geworden. Daher ist eine hohe Qualität dieser Systeme essenziell. IT-Experten haben die Erfahrung gemacht, dass die Qualität eines IT-Systems in erster Linie durch die an das System gestellten Anforderungen bestimmt wird.

Basierend auf dieser Erkenntnis ist ein gänzlich neuer IT-Berufszweig entstanden: das Requirements Engineering. Die zentrale Idee des Requirements Engineering ist das Teilen von Informationen. Die Disziplin des Requirements Engineering befasst sich mit der Ermittlung, Dokumentation, Validierung, Abstimmung und Verwaltung sämtlicher Informationen, die Systementwickler und -betreiber benötigen, um erfolgreiche Systeme zu entwickeln, zu betreiben und zu administrieren.

Requirements Engineering hilft allen Beteiligten dabei, zu verstehen, welche Art von System wirklich benötigt wird. In bestimmten Kontexten wird das Requirements Engineering von einem Beteiligten mit dedizierter Rolle durchgeführt: dem „Requirements Engineer“. In anderen Kontexten ist das Requirements Engineering Teil einer umfassenderen Rollendefinition, beispielsweise: Systems Engineer [Walten et al.2015] oder Digital Designer [Bitkom2017]. Der Einfachheit halber wird in diesem Handbuch der Begriff Requirements Engineer verwendet.



Künstliche Intelligenz (KI) bietet neue Möglichkeiten, Requirements Engineers bei der Planung und Durchführung der Anforderungsermittlung zu unterstützen. Besonders nützlich sind hierbei sogenannte „Large Language Models“ (LLM) und generative KI-Systeme. Solche KI-Systeme können beispielsweise Daten aus vorherigen Projekten analysieren und Muster erkennen, die Menschen nur schwer finden. Sie helfen dabei, passende Ermittlungstechniken zu wählen und den Plan für die Anforderungsermittlung flexibel an das jeweilige Projekt anzupassen.

Mit generativer KI und LLM lassen sich außerdem Risiken und Konflikte frühzeitig erkennen. Dies geschieht durch eine automatische Auswertung von Texten und Gesprächen. Die KI kann dabei aufzeigen, welche Themen zu Problemen führen könnten und wo mehr Klärung nötig ist. So erkennt der Requirements Engineer leichter, wo Konflikte entstehen könnten, und kann diese schneller lösen.

Da KI große Mengen an Stakeholder-Daten auswerten kann, ist es wichtig, den Datenschutz und ethische Fragen genau zu beachten. Es muss klar geregelt sein, wie mit den Daten umgegangen wird und dass alle Stakeholder über die Verwendung von KI informiert werden. Ihre Rechte, wie zum Beispiel das Recht auf Privatsphäre, müssen unbedingt geschützt werden. KI soll daher immer nur unterstützend wirken, ohne die Verantwortung und das Urteilsvermögen des Requirements Engineers zu ersetzen.

1.1 Der Anwendungsbereich der Ermittlung im Requirements-Engineering

Laut der Definition von Requirements-Engineering nach [PoRu2015] 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.

Dieses Dokument beschreibt die Anforderungserhebung und Konfliktlösung für den Requirements Elicitation Practitioner und Specialist. Das erste Kapitel dient als Einführung in das Thema und als Richtlinie für die praktische Anwendung. In **Kapitel 2** werden die **Anforderungsquellen** beschrieben. Die Entscheidung darüber, welche Quellen relevant sind, bildet den Ausgangspunkt für jeden Ermittlungsaufwand. **Kapitel 3, Anforderungsermittlung**, bietet einen Überblick über die Techniken, die zur Ermittlung von Anforderungen verwendet werden können, sowie Orientierungshilfen zu deren Verwendung. **In Kapitel 4, Konfliktlösung**, werden Möglichkeiten zur Auflösung von Situationen behandelt, in denen Anforderungen im Widerspruch zueinanderstehen. Das Handbuch schließt mit **Kapitel 5, Fähigkeiten eines Requirements Engineers**, ab, in dem wichtige Aspekte für Experten erörtert werden, die in diesem Bereich tätig sein möchten.

Gemäß dem IREB CPRE Foundation Level Lehrplan [IREB2020] hat Requirements Engineering drei weitere Hauptaktivitäten: die *Dokumentation*, *Validierung* und die *Verwaltung von Anforderungen*. Dokumentation bezieht sich auf die Wege zur Erfassung der Ergebnisse des Ermittlungsprozesses als Mittel für die weitere Kommunikation. Ein separates Handbuch der Anforderungsmodellierung nach IREB Standard [CHQW2016] bietet weitere Informationen zu einem Teil dieses Themas. Bei der Verwaltung geht es darum, eine Reihe von ermittelten, dokumentierten und konsolidierten Anforderungen während ihres gesamten Lebenszyklus in einem guten Zustand zu halten. Diese Aktivität ist im „Handbuch Requirements Management nach IREB Standard“ [BuHe2019] detaillierter beschrieben.

1.2 Relevante Einflussfaktoren für das Vorgehen zur Planung von Ermittlungsaktivitäten

In Veröffentlichungen über die Aufwandsschätzung in Softwareentwicklungsprojekten [McCo2006] und über Ergebnisse aus der industriellen Praxis kommt der Disziplin des Requirements Engineering eine hohe Verantwortung für das Erreichen der Projekterwartungen insgesamt zu. Im Requirements Engineering wiederum hängt das Erreichen der Ziele in erheblichem Maße von der Anforderungsermittlung ab. Dies erfordert aus den folgenden Gründen ein bestimmtes Vorgehen bei der Planung:

1. Die Anforderungsermittlung kann nicht allein auf Basis der erwarteten Größe des Endergebnisses geplant werden. Man kann unmöglich die Feststellung machen, dass 107 Anforderungen ermittelt werden sollen, für die durchschnittlich jeweils 1,25 Stunden gebraucht werden. Der Grund dafür ist einfach: Wir kennen den Umfang und die Form der Anforderungsermittlungsergebnisse nicht. Wir müssen die Anforderungen ermitteln, weil wir sie nicht kennen.
2. Wenngleich Anforderungskonflikte in einem Projekt nicht geplant oder vorhergesagt werden können, werden solche Konflikte in jedem Projekt auftreten. Sobald ein Anforderungskonflikt auftritt und erkannt wird, muss der Requirements Engineer darauf reagieren.

Beides könnte zu dem Missverständnis führen, dass es mithilfe von Projektmanagementtechniken nicht möglich ist, die Anforderungsermittlung zu planen und zu steuern. Es stimmt zwar, dass es nicht immer ratsam ist, einen detaillierten Vorabplan für die Anforderungsermittlung zu definieren (einschließlich der gewählten Ermittlungstechniken, eines detaillierten Budgets und Zeitplans). Die Betonung liegt jedoch auf „Vorab“, da sich ein detaillierter Vorabplan auf Annahmen stützen muss (schließlich ermitteln wir, weil wir nicht alle Fakten kennen) und diese Annahmen häufig bereits kurz nach Projektbeginn hinfällig werden. Daher ist ein detaillierter Plan nur dann ratsam, wenn genügend Vorabwissen vorhanden ist (beispielsweise über die Struktur des geplanten Systems) oder wenn ausreichendes Vertrauen in die zugrunde liegenden Annahmen besteht (z. B. welche Aspekte des neuen Systems wichtig sein werden).

Die Planung und Durchführung der Aktivitäten zur Anforderungsermittlung sind der Planung und Durchführung eines Forschungsprojekts sehr ähnlich. Ein Forschungsprojekt beginnt in der Regel damit, eine oder mehrere Forschungsfragen (oder Problemstellungen) zu formulieren und eine Folge von Aktivitäten festzulegen, um die definierten Forschungsfragen zu beantworten und ausführlich zu erläutern. Die Ähnlichkeit einer Aktivität zur Anforderungsermittlung und einem Forschungsprojekt besteht darin, dass die Anfangsphase beider Aktivitäten von Unsicherheiten und Annahmen (oder eine Hypothese) geprägt ist. Daher kann das Forschungsprojekt nicht vollständig vom Anfang bis zum Ende geplant werden. Bei einem Forschungsprojekt werden stattdessen Aktivitäten definiert, die ausgewählte Forschungsfragen aufgreifen, um im Projektverlauf Unsicherheiten zu klären oder die Annahmen (bzw. die Hypothese) zu verifizieren (oder zu falsifizieren). Das bedeutet insbesondere, dass der Forschungsplan und die Forschungsfragen auf Basis der neuen Feststellungen kontinuierlich geprüft, verfeinert und aktualisiert werden.

Dasselbe gilt für die Ermittlung von Anforderungen. Aktivitäten zur Anforderungsermittlung folgen gewöhnlich einem explorativen Ansatz. Mit dem Anforderungsermittlungsplan werden zu Beginn bestimmte Ziele gesetzt (zu beantwortende „Forschungsfragen“) sowie ein grober Zeitrahmen und allgemeine Ausgangskriterien festgelegt. In einer Reihe von aufeinanderfolgenden Iterationen werden Antworten auf diese Fragen gefunden und verfeinert, dann werden basierend auf diesen Antworten die nächsten Schritte unternommen, bis die Stakeholder die resultierenden Anforderungen als angemessen akzeptieren. Moderne Softwareentwicklungsprozesse und agile Methoden unterstützen einen iterativen Ansatz, der aus kurzen Zyklen besteht, in denen alternative Lösungen erstellt werden und Feedback eingearbeitet wird.

Empfohlene Lektüre

[Beveridge 1957] bietet eine gute Einführung in die Definition von Forschungsprojekten. Der vollständige Text dieses Buches ist verfügbar auf www.archive.org.

1.3 Planung und Durchführung der Anforderungsermittlung

Wie wir zuvor gesehen haben, erfordert die Anforderungsermittlung eine spezielle Herangehensweise bei der Planung. In diesem Abschnitt beschreiben wir ein Konzept für die Planung und Durchführung von Aktivitäten zur Anforderungsermittlung. Der Hauptzweck dieses Konzepts ist didaktischer Natur: Es soll zur Unterstützung von Lehre und Schulung dienen. In dem Konzept werden die für die Planung und Durchführung der Anforderungsermittlung notwendigen Schritte und Informationen verdeutlicht. Das Konzept sollte nicht als direkt anwendbares Prozessinstrumentarium verstanden werden; ebenso wie andere Prozessrahmenkonzepte muss dieses Konzept bei der praktischen Anwendung an die jeweilige Situation angepasst werden.

Schließlich kann die Planung und Durchführung einer Ermittlung nicht unabhängig von anderen Tätigkeiten in einem Systementwicklungsprojekt betrachtet werden. Die Anpassung des Ansatzes an ein bestimmtes Projekt erfordert ein tiefgehendes Verständnis des Projektkontextes; dabei sind Arbeitsergebnisse wie die Ausarbeitung der Produktvision, die Projektbeschreibung und der Business Case zu berücksichtigen. Aus diesem Grund bleibt die Anpassung in diesem Handbuch unberücksichtigt. Der Schwerpunkt liegt im Folgenden vielmehr auf den zentralen Konzepten der Anforderungsermittlung.

Zur Definition unseres Konzepts gehen wir davon aus, dass in jedem Projekt mit Ermittlungsaktivitäten auf irgendeine Art von Plan zurückgegriffen wird, um die Herangehensweise oder die Aufgaben zu strukturieren. Dies kann etwa ein komplexer Projektplan mit Meilensteinen oder ein agiles Backlog sein.

Wir definieren zwei Tätigkeiten, die in jede Art von Plan integriert werden können:

- Ermittlungsaktivitäten zur Identifizierung der Anforderungsquellen und zur Erfassung der Anforderungen
- Konfliktlösungsaktivitäten zur Lösung von auftretenden Anforderungskonflikten

Beide Aktivitäten können zusätzlich Projektmanagement-Informationen bezüglich Zeit und Ressourcen enthalten. Näheres zur Definition dieser Ermittlungsaktivitäten ist in den folgenden Unterabschnitten enthalten.

1.3.1 Ermittlungsaktivität

Eine Ermittlungsaktivität dient zur Planung der Ermittlung von Anforderungen oder zur Identifizierung von Anforderungsquellen. Der Inhalt einer Ermittlungsaktivität wird anhand von fünf Aspekten beschrieben: Ermittlungsziel, Ergebnisqualität, Anforderungsquelle, Ermittlungstechnik und Projektmanagement-Informationen.

Wir verwenden das Beispiel einer Ermittlungsaktivität in einem Projekt, in dessen Rahmen Kundendiensttechniker eines weltweit tätigen Serviceunternehmens für Schiffsmotoren Unterstützung durch Mobilgeräte erhalten sollen:

ID	RS_EA_13
Ermittlungsziel	Bestimmung der Diversität (in jedem relevanten Aspekt) innerhalb der Benutzergruppe „Kundendiensttechniker“
Ergebnisqualität	Geeignete Personas (eine, wenn sie homogen sind; mehrere, wenn sie heterogen sind) für die Benutzergruppe „Kundendiensttechniker“
Anforderungsquelle(n)	Kundendiensttechniker in fünf ausgewählten Servicestellen rund um den Globus: Hamburg, Kapstadt, Buenos Aires, Dubai, Osaka
Ermittlungstechnik	Contextual Inquiry

Abbildung 1: Beispiel einer Ermittlungsaktivität („ID“ ist eine Projektmanagement-Information)

Diese fünf Elemente werden im folgenden Unterabschnitt beschrieben. Die Elemente weisen starke Beziehungen untereinander auf. Das Verständnis dieser Beziehungen hilft bei der Definition guter Ermittlungsaktivitäten. Die Beschreibung der Beziehungen findet sich am Ende dieses Unterabschnitts.

1.3.1.1 Ermittlungsziel

Das *Ermittlungsziel* ist das zentrale Konzept des Planungsansatzes, da sich alle weiteren Elemente daran orientieren. Ein Ermittlungsziel kann in einem Ermittlungsprojektplan mehrmals vorkommen, beispielsweise wenn wir eine bestimmte Frage anhand verschiedener Techniken oder der Analyse mehrerer Anforderungsquellen beantworten wollen.

Das Ermittlungsziel sollte so präzise wie möglich formuliert werden. Es dient folgenden Zwecken:

1. Es beschreibt, was wir mithilfe dieser bestimmten Ermittlungsaktivität erfahren oder verstehen möchten, d. h. die zu identifizierenden Anforderungsquellen oder die zu ermittelnden Anforderungen.
2. Es unterstützt die Identifizierung geeigneter Anforderungsquellen (siehe Abschnitt 2.2).
3. Es unterstützt die Auswahl einer Ermittlungstechnik.
4. Es kann zur Messung des Erfolgs einer Aktivität herangezogen werden (wurde das Ziel am Ende der Aktivität erreicht?).
5. Die Ergebnisqualität des Ermittlungszieles ist ein wichtiger Indikator für das Maß an Verständnis, das für das zu entwickelnde System besteht (siehe Abschnitt 1.3.2.2).

Es gibt verschiedene Möglichkeiten, Ermittlungsziele zu formulieren, etwa durch:

1. die Formulierung realer Fragen zu den Anforderungen an das System¹.

Beispiel: Welche Schritte des Geschäftsprozesses, die das neue CRM (Customer Relationship Management)-System unterstützen muss, sind am wichtigsten und wie sieht diese Unterstützung aus?

2. die Formulierung von Ermittlungszielen als Hypothesen, die zu bestätigen oder zu entkräften sind.

Beispiel: Die Smartphone-App des CRM-Systems muss Funktionalitäten zum Hinzufügen neuer Kunden bieten. Es genügt nicht, wenn diese Funktionalität nur in der Desktop-Anwendung verfügbar ist.

3. die Formulierung von Ermittlungszielen basierend auf dem Kano-Modell mithilfe der Kategorien „Basisfaktoren“, „Leistungsfaktoren“ und „Begeisterungsfaktoren“ (siehe [IREB2020]).

Beispiel: Die Basisfaktoren des vorhandenen CRM-Systems verstehen.

4. die Verwendung von Standards oder Vorlagen für Spezifikationen in Ihrer Domäne.

Beispiel: Qualitätsanforderungen gemäß [ISO25010] identifizieren, z. B. Anforderungen an Benutzbarkeit, Zuverlässigkeit oder Sicherheit.

¹ [Mill2009] stellt einen umfangreichen Fragenkatalog im Zusammenhang mit nicht-funktionalen Anforderungen zur Verfügung. [Withall2007] stellt eine Liste mit Anforderungsmustern bereit, die zur Ableitung von Fragen verwendet werden können.

1.3.1.2 Ergebnisqualität

Die *Ergebnisqualität* beschreibt die angestrebte Qualität des Ergebnisses der Aktivität in Bezug auf das Maß an Gewissheit, Vollständigkeit und Einigung.

Gewissheit bezieht sich auf den Grad der Evidenz, der für die Korrektheit des Ergebnisses gegeben werden kann (d. h. die Anforderungsquellen oder Anforderungen).

Mit Vollständigkeit meinen wir, inwieweit das Ergebnis in Bezug auf den „theoretisch möglichen“ Informationsumfang, der im angestrebten Detaillierungsgrad hätte ermittelt werden können, abgedeckt ist.

Einigung bezieht sich auf den Grad, bis zu dem die Stakeholder mit dem Ergebnis einverstanden sein müssen. Dies deckt auch ab, welche Stakeholder sich einigen müssen. Bedenken Sie immer, dass das Erreichen einer Einigung der Stakeholder unter Umständen viel Kommunikation und daher einen erheblichen Aufwand erfordern kann.

Der Requirements Engineer sollte immer die Balance zwischen den Kosten zusätzlicher Ermittlungsaktivitäten und dem Nutzen für das aktuelle Projekt im Blick behalten. Das richtige Maß an Gewissheit und Vollständigkeit entsteht durch kontinuierliche Interaktion zwischen dem Requirements Engineer und den relevanten Stakeholdern.

1.3.1.3 Anforderungsquellen

Die *Anforderungsquelle* beschreibt die Quelle bzw. die Quellen, mit deren Hilfe die Anforderungen ermittelt werden sollen, oder die Quellen, die zur Identifizierung anderer Anforderungsquellen genutzt werden. Wir plädieren dafür, den Fokus der Ermittlungsaktivität auf genau eine Art von Quelle zu richten.

1.3.1.4 Ermittlungstechnik

Die *Ermittlungstechnik* ist die spezielle Technik, die zur Ermittlung der Anforderungen anhand einer Quelle verwendet wird.

Folgende Informationen sind u. a. für eine Ermittlungstechnik relevant:

- Der geschätzte Aufwand für die Vorbereitung/Durchführung/Nachbereitung: Vorbereitung, Durchführung und Nachbereitung einer Ermittlungstechnik können für den Requirements Engineer und/oder die Stakeholder mit erheblichem Aufwand verbunden sein. Für ein Interview ist beispielsweise die Vorbereitung der Fragen erforderlich. Das Apprenticing kann je nach Komplexität des Projekts sowohl für den Requirements Engineer als auch für die betroffenen Stakeholder mehrere Tage in Anspruch nehmen. Für einen Workshop ist möglicherweise die Analyse der Workshop-Ergebnisse erforderlich. Dieser Aufwand sollte geschätzt und dokumentiert werden, um die Planung von Ermittlungstechniken zu verbessern.
- Auf die Vorbereitung/Durchführung/Nachbereitung verwendeter Aufwand: Zusätzlich zum geschätzten Aufwand sollte der tatsächlich auf eine Ermittlungstechnik verwendete Aufwand dokumentiert werden. Eine signifikante Abweichung zwischen

dem geschätzten und dem aufgewendeten Aufwand sollte analysiert werden, um die Gründe für die Abweichung zu verstehen.

- Zeitplan für die Vorbereitung/Durchführung/Nachbereitung: Neben dem erforderlichen Aufwand ist der Zeitplan für die Ermittlungsaktivität für Requirements Engineers und Stakeholder wichtig, insbesondere bei Aktivitäten, für die ein sehr hoher Aufwand nötig ist. Durch den Zeitplan kann auch ein Termin vorgegeben sein.
- Verweis auf Vorbereitungsmaterialien: Wenn für eine Technik Vorbereitungsmaterialien benötigt werden (z. B. für einen Workshop zur Diskussion eines UI Mock-ups für die Benutzeroberfläche), sollte auf diese verwiesen werden.

Hier besteht die Aufgabe darin, die optimale(n) Technik(en) für das vorhandene Ermittlungsziel auszuwählen. Jede Ermittlungstechnik hat Vor- und Nachteile. Wie bei den Anforderungsquellen, sollten Sie sich nicht auf eine einzige Ermittlungstechnik verlassen. Interviews können Ihnen zwar dabei helfen, detaillierte und spezifische Informationen von mitteilensamen Stakeholdern zu erhalten, sie werden Ihnen aber wahrscheinlich keine große Hilfe sein, wenn Sie auf introvertierte Personen treffen oder jemanden, der Angst davor hat, Ihnen „falsche“ Antworten zu geben.

Abschnitt 3.4 soll Ihnen als Orientierungshilfe dafür dienen, wie Sie eine geeignete Auswahl an Ermittlungstechniken treffen können.

Denken Sie daran: Definieren Sie für jede Technik separate Ermittlungsaktivitäten!

1.3.1.5 Projektmanagement-Informationen

Eine Ermittlungsaktivität kann durch mehrere Projektmanagement-Attribute gekennzeichnet sein: Die konkreten Attribute hängen vom jeweiligen Projektkontext und von der angewandten Methode ab.

Nachfolgend sind einige praktische Beispiele für Projektmanagement-Attribute aufgeführt:

- Autor: die Person, welche die jeweilige Ermittlungsaktivität definiert hat
- Verantwortliche(r) Requirements Engineer(s): die für die Durchführung der Ermittlungsaktivität verantwortlichen Projektteammitglieder
- Priorität: Die Bedeutung einer Ermittlungsaktivität für das Gesamtprojekt
- Abhängigkeiten zu anderen Ermittlungsaktivitäten: Eine Ermittlungsaktivität kann vom Ergebnis anderer Ermittlungsaktivitäten abhängen. Für einen Anforderungsworkshop wird beispielsweise der Input aus Interviews mit Stakeholdern benötigt.
- Referenzen zu den dokumentierten Anforderungen: Sobald eine Aktivität abgeschlossen ist, können die daraus entstandenen Anforderungen Referenzen enthalten, anhand derer die Verfolgbarkeit zwischen den Anforderungen und der Ermittlungsaktivität gepflegt werden kann.

1.3.1.6 Nutzung der Beziehungen zwischen den Elementen einer Ermittlungsaktivität

Die Ermittlungstechnik ist das Werkzeug, anhand dessen die Ermittlung durchgeführt wird. Die Auswahl der geeigneten Technik ist für den Erfolg einer Ermittlungsaktivität entscheidend. Dennoch sollte die Entscheidung für eine bestimmte Technik den letzten Schritt bei der Definition einer Ermittlungsaktivität darstellen.

Der Hauptgrund dafür ist, dass jeder Aspekt einer Ermittlungsaktivität eine Beziehung zu den anderen Aspekten aufweist. Dies kann dafür genutzt werden, die Gesamtermittlungsaktivität zu validieren und zu verbessern. Abbildung 2 zeigt die vier Aspekte und die sechs untereinander bestehenden Beziehungen:

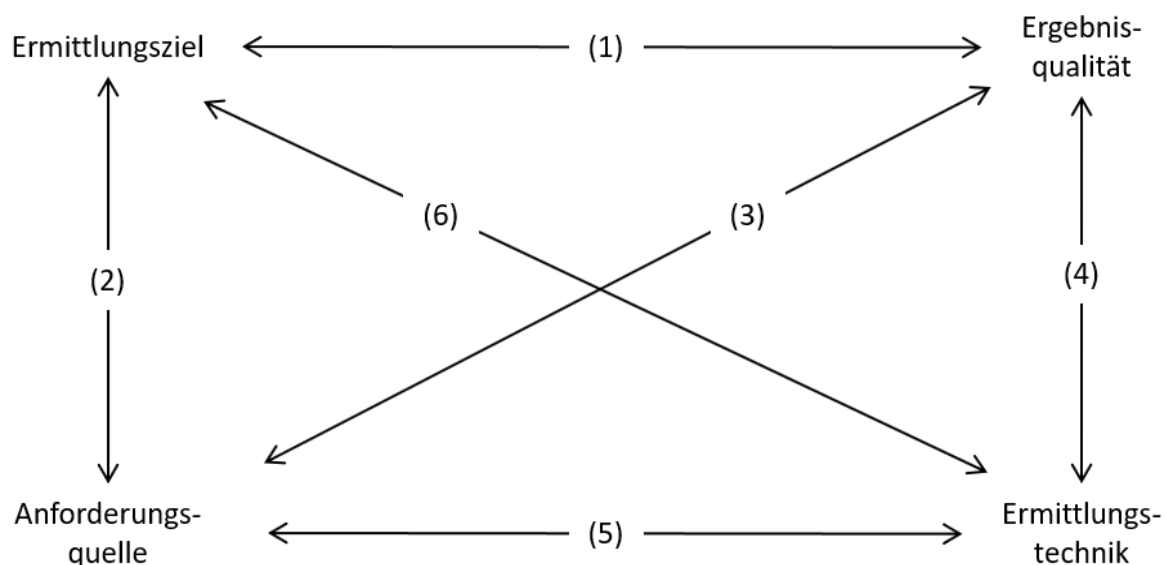


Abbildung 2: Beziehungen zwischen den Elementen von Ermittlungsaktivitäten

1. Ermittlungsziel – Ergebnisqualität: Das Ermittlungsziel muss so definiert werden, dass es möglich ist, eine (ausreichend) präzise Definition der für das Ergebnis angestrebten Gewissheit und Vollständigkeit zu formulieren.
2. Ermittlungsziel – Anforderungsquelle: Ist die ausgewählte Anforderungsquelle für das Erreichen des Ermittlungsziels nützlich? (Ein altes Prozessdokument ist für das Verständnis der realen Geschäftsprozesse eines Unternehmens möglicherweise ungeeignet.)
3. Ergebnisqualität – Anforderungsquelle: Kann die gewählte Anforderungsquelle die angestrebte Ergebnisqualität liefern? (Ist es möglich, bestimmte Anforderungen mit nur einem Stakeholder zu klären?)
4. Ergebnisqualität – Ermittlungstechnik: Kann die gewählte Technik die angestrebte Qualität liefern? (Ist es möglich, bestimmte Anforderungen in einer Brainstorming-Sitzung zu klären?)
5. Anforderungsquelle – Ermittlungstechnik: Technik und Quelle müssen miteinander kompatibel sein (ein Interview mit einem Dokument ergibt keinen Sinn!), und die

Quellen müssen mithilfe der Technik adressierbar sein. (Ist es möglich, alle sechs Vorstandsmitglieder eines Unternehmens in einem fünftägigen Workshop zusammenzubringen?)

6. Ermittlungsziel – Ermittlungstechnik: Ist die gewählte Technik zum Erreichen des definierten Ziels geeignet? (Ist es möglich, einen bestimmten Geschäftsprozess mithilfe einer Kreativitätstechnik zu verstehen?)

Diese sechs Beziehungen zeigen, dass eine geeignete Ermittlungstechnik nur dann ausgewählt werden kann, wenn die drei anderen Aspekte klar definiert sind.

1.3.2 Konfliktlösungsaktivität

Diese Art von Aktivität wird genutzt, um Konflikte zwischen Anforderungen aufzulösen. Anforderungen gelten als widersprüchlich, wenn sie nicht gleichzeitig im selben System implementiert werden können. Näheres finden Sie unter Kapitel 4.

Der Inhalt einer Konfliktlösungsaktivität lässt sich anhand von vier Elementen beschreiben: Beschreibung einbezogener Anforderungen, einbezogener Anforderungsquellen, der Konfliktlösungstechnik und des erreichten Ergebnisses (nach der Konfliktlösung).

Die Konfliktlösungsaktivität wird beschrieben, sobald der Konflikt identifiziert wurde. Zu diesem Zeitpunkt können in der Regel noch nicht alle Aspekte der Konfliktlösungsaktivität beschrieben werden. Die übrigen Aspekte werden während der Konfliktlösung dokumentiert.

ID	RS_CRA_3
Einbezogene Anforderungen	RS_REQ_37 und RS_REQ_221 sind nicht kompatibel. RS_REQ_37 erfordert eine WLAN-Verbindung, REQ_221 hingegen eine Kabelverbindung. Konfliktart: noch unklar
Einbezogene Anforderungsquellen	RS_REQ_37 war ein Beitrag von Frau Highmore, RS_REQ_221 stammt aus der Systemarchäologie des Altsystems.
Lösungstechnik	Bevorzugte Technik: Einigung Lösungsplan: Meeting mit Frau Highmore und Herrn Strong (System Owner des Altsystems) ansetzen, um eine passende Lösung zu finden.
Erreichtes Lösungsergebnis	noch zu erledigen (wird dokumentiert, wenn eine Lösung gefunden wurde)

Abbildung 3: Beispiel einer Konfliktlösungsaktivität („ID“ ist die Projektmanagement-Information) im Status der Identifizierung

1.3.2.1 Beschreibung der einbezogenen Anforderungen

Die Beschreibung benennt die im Widerspruch stehenden Anforderungen und die Gründe für die Widersprüche. Sie sollte die Notwendigkeit für die Lösung eindeutig und in adäquater Detailliertheit beschreiben, einschließlich einer Referenz auf die betroffenen Anforderungen. In der Beschreibung sollte näher auf die Konfliktart eingegangen werden (siehe Abschnitt 4.2).

1.3.2.2 Einbezogene Anforderungsquellen

Die einbezogenen Anforderungsquellen (z. B. Stakeholder) sind die Beteiligten, die im Konfliktlösungsprozess berücksichtigt werden müssen.

Eine Anforderung im Zusammenhang mit der Überwachung von Benutzeraktivitäten aus Sicherheitsgründen steht beispielsweise im Widerspruch zu einer Anforderung für den Schutz personenbezogener Daten. Ein Sicherheitsexperte hat die Anforderung für die Überwachung geäußert und die Datenschutzerfordernung ist auf ein domänenspezifisches Gesetz zurückzuführen. Bei diesem Konflikt ist der Sicherheitsexperte ein Stakeholder, und der Stakeholder für das domänenspezifische Gesetz könnte der Datenschutzbeauftragte des Unternehmens sein.

Die Identifizierung der involvierten Stakeholder ist ein wichtiger Teil der Konfliktidentifizierung (siehe Abschnitt 4.1).

1.3.2.3 Lösungstechnik

Im Lösungsansatz wird der für die Konfliktlösung geplante Ansatz festgelegt. Der Lösungsansatz besteht üblicherweise aus einer Beschreibung der ausgewählten Lösungstechnik und zusätzlichen vorbereitenden Aktivitäten.

Details zur Auswahl des geeigneten Lösungsansatzes werden in Abschnitt 4.3 dargelegt.

1.3.2.4 Erreichtes Lösungsergebnis

Nach der Lösung wird in einem letzten Schritt der Konfliktlösungsaktivität eine kurze Beschreibung des erreichten Ergebnisses dokumentiert. Diese kurze Beschreibung ist als Zusammenfassung der Konfliktlösung für Personen gedacht, die diese Aktivitäten in einer späteren Projektphase überprüfen.

1.3.2.5 Projektmanagement-Informationen

Eine Konfliktlösungsaktivität kann durch mehrere Projektmanagement-Attribute gekennzeichnet sein. Die genauen Informationen hängen vom jeweiligen Projektkontext ab. Nachfolgend sind einige zweckdienliche Aspekte für das Projektmanagement aufgeführt:

- Autor: die Person, die die spezielle Konfliktlösungsaktivität definiert hat

- Verantwortliche(r) Requirements Engineer(s): die für die Durchführung der Konfliktlösungsaktivität verantwortlichen Projektteammitglieder
- Priorität: Die Bedeutung einer Konfliktlösungsaktivität für das Gesamtprojekt
- Referenz zu detaillierter Konfliktdokumentation: Wenn ein Konflikt mit komplexen Problemen zusammenhängt, sollte auf weitere Dokumentation verwiesen werden, in der die Details zum Konflikt beschrieben sind.
- Verweis auf Vorbereitungsmaterialien: Wenn für die Konfliktlösung Vorbereitungsmaterialien benötigt werden (z. B. eine detaillierte Beschreibung für die Stakeholder), dann sollte auf diese Materialien verwiesen werden.
- Abhängigkeiten zu anderen Aktivitäten: Eine Konfliktlösungsaktivität kann vom Ergebnis anderer Konfliktlösungs- oder Ermittlungsaktivitäten abhängen. Für einen Konfliktlösungsvorschlag kann z. B. der Input aus Interviews mit Stakeholdern benötigt werden.
- Geschätzter Aufwand für die Vorbereitung/Durchführung/Nachbereitung: Vorbereitung, Durchführung und Nachbereitung einer Konfliktlösungstechnik können für den Requirements Engineer und/oder die Stakeholder mit erheblichem Aufwand verbunden sein.
- Letzter Zeitpunkt, zu dem der Konflikt gelöst werden muss: Es ist möglicherweise nicht notwendig, jeden Konflikt unmittelbar zu lösen, nachdem er erkannt wurde. Wenn dies verschoben werden kann, sollte der letzte vertretbare Zeitpunkt, bis zu dem die Lösung gefunden werden muss, dokumentiert werden.
- Zeitplan für die Vorbereitung/Durchführung/Nachbereitung: Neben dem Aufwand ist der Zeitplan für eine Lösungsaktivität für Requirements Engineers und Stakeholder wichtig, insbesondere bei Aktivitäten, für die ein sehr hoher Aufwand nötig ist.

1.3.3 Richtlinien für den Ermittlungsteil eines Projekts

Im vorangehenden Unterabschnitt haben wir Ermittlungsaktivitäten definiert; in diesem Unterabschnitt stellen wir einige Richtlinien für die Anwendung von Ermittlungsaktivitäten von der Planung bis zur Durchführung vor. Diese Richtlinien gelten unabhängig von einem bestimmten Projektmanagementansatz (beispielsweise der Wasserfall-Ansatz oder eine agile Herangehensweise). Der allgemeine Begriff „Projekt“ ist etwa auf ein reines RE-Projekt, ein Projekt zur Geschäftsanalyse/Business Analysis oder ein Softwareprojekt anwendbar.

Hinweis 1.3.1:

Wenn Sie ein physisches oder digitales Kanban-Board für das Aufgabenmanagement verwenden, eignen sich Ermittlungsaktivitäten sehr gut als einzelne Aufgaben. Gleiches gilt auch für Gantt-Diagramme.

1.3.3.1 Unterscheidung verschiedener Konstellationen von Ermittlungsaktivitäten

Wir empfehlen die Unterscheidung drei verschiedener Konstellationen von Ermittlungsaktivitäten:

- *Konstellation 1 – Ausgeführte Ermittlungsaktivitäten:* Diese Konstellation umfasst sämtliche Ermittlungsaktivitäten, die bis zum aktuellen Zeitpunkt im Projektverlauf ausgeführt wurden. Sie beschreibt den Verlauf Ihres Projekts aus der Ermittlungsperspektive und dient als Projektgedächtnis. Diese Konstellation ist zu Beginn natürlich leer.
- *Konstellation 2 – Kurzfristige Ermittlungsaktivitäten:* Dies umfasst sämtliche Ermittlungsaktivitäten, deren Ausführung in der nahen Zukunft geplant ist. Die Ermittlungsaktivitäten in dieser Konstellation müssen detailliert geplant werden, und ihre Durchführung sollte eingeplant und vorbereitet werden. Sie können diese Konstellation in Ihrem Projekt als To-do-Liste für die nähere Zukunft betrachten.
- *Konstellation 3 – Langfristige Ermittlungsaktivitäten:* Dies umfasst sämtliche Ermittlungsaktivitäten, die als wichtig erachtet werden, die aber noch nicht detailliert geplant und zeitlich eingeplant sind. Der Grund dafür, dass einige Aktivitäten als langfristig definiert werden, ist, dass der aktuelle Status des Projekts (vor allem vorhandenes Wissen und Annahmen) eine detaillierte Planung nicht immer erlaubt. Sie können diese Konstellation als ein Backlog für Ziele betrachten, die noch weiter ausgearbeitet werden müssen.

Mit dem Projektfortschritt wächst der Anteil der ausgeführten Ermittlungsaktivitäten, da kurzfristige Ermittlungsaktivitäten ausgeführt werden. Langfristige Aktivitäten werden detailliert aufgeführt und werden schließlich zu kurzfristigen Aktivitäten, oder sie werden durch mehrere kurzfristige Aktivitäten verfeinert oder gar völlig aufgegeben, da sie für das Projekt nicht mehr relevant sind.

Wir empfehlen die Unterscheidung zwischen der Vorbereitungs- und der Durchführungsphase der Ermittlung. In den folgenden Unterabschnitten stellen wir Richtlinien für beide Phasen bereit.

Zusätzlich zur Vorbereitungs- und Durchführungsphase ist es möglich, eine *Abschlussphase* durchzuführen, die zum Ziel hat, aus dem Projekt zu lernen und die Fähigkeiten der Projektteilnehmer weiterzuentwickeln. In dieser Phase wird die Versionshistorie des Ermittlungsplans mit den Ergebnissen (den ermittelten Anforderungen und den gelösten Konflikten) abgeglichen, damit aus den Erfolgen und Fehlern des abgeschlossenen Projekts Erkenntnisse gewonnen werden. In iterativen Projekten kann diese Phase nach jeder Iteration durchgeführt werden.

1.3.3.2 Richtlinien für die Vorbereitungsphase

In der *Vorbereitungsphase* wird die Anfangskonstellation der Ermittlungsaktivitäten festgelegt. Diese Anfangskonstellation beschreibt den gewünschten Ansatz für die Ermittlung von Anforderungen. Sie beruht auf den spezifischen Projektmerkmalen (z. B. dem

Entwicklungsansatz) und dem vorhandenen Wissen und den Annahmen der Projektteilnehmer. Lösungsaktivitäten werden in der Vorbereitungsphase normalerweise nicht definiert, da es in diesem Stadium keine Anforderungskonflikte gibt. Wenn es jedoch bekannte Anforderungskonflikte (beispielsweise aus früheren Projekten) oder Indikatoren für potenzielle Konflikte gibt, sollte sobald wie möglich ein Plan für deren Beseitigung eingearbeitet werden.

Der Begriff *Vorbereitungsphase* impliziert nicht, dass Sie mehrere Tage oder Wochen für die Erarbeitung dieser Anfangskonstellation von Aktivitäten aufwenden müssen. Dennoch ist die Vorbereitungsphase für einen erfolgreichen Projektstart und für das Ermöglichen einer effektiven Anforderungsermittlung von großer Bedeutung. Denken Sie daran: Die Ermittlung erfolgt ähnlich wie bei einem Forschungsprojekt – und für den effektiven Verlauf eines guten Forschungsprojekts ist die Definition geeigneter Forschungsziele nötig.

Ein Überblick über die Projektsituation und das Geschäftsszenario verschaffen

Jedes Projekt ist einzigartig. Was bei einem Projekt funktioniert hat, kann bei einem anderen unter Umständen völlig falsch sein. Es ist wichtig, jedes neue Projekt zu analysieren, um ein klares Bild davon zu erhalten, welche Ermittlungsaktivitäten geeignet sind. Zunächst müssen Sie die Art und den Kontext des Projekts verstehen. Die Domäne ist möglicherweise völlig neu für Sie, und/oder die Organisation oder Sie arbeiten vielleicht schon mehrere Jahre auf dem Gebiet. Auch wenn Letzteres der Fall ist, sind einige der Probleme, die im folgenden Abschnitt behandelt werden, nicht unbedingt weniger wichtig für Sie, nur weil Sie über viel Erfahrung verfügen.

Ein Entwicklungsprojekt wird auch stark durch seine **Domäne** beeinflusst. Wissen Sie genug über diese Domäne, um zu verstehen, wie sie das Projekt beeinflusst und was wichtig ist? Ist eine Domäne neu für Sie, dann werden Sie (zumindest einführende) Literatur und/oder Websites lesen müssen, um den Kontext und die Terminologie zu verstehen. Ein gängiges Ermittlungsziel, um ein Verständnis für die Domäne zu entwickeln, ist die Erstellung eines Glossars (siehe auch [IREB2020]). Weiterhin sollten Sie nach Personen suchen, die Ihnen dabei helfen können, die Domäne zu verstehen. Diese sollten Sie bitten, Ihnen zu erklären, was Sie im Kontext Ihres Projekts wissen müssen.

Es ist wichtig, den Verlauf eines Projekts zu verstehen. Kein Projekt entsteht aus dem Nichts; selbst ein brandneues Projekt hat eine Vorgeschichte. Sie müssen diese Vorgeschichte kennen, um die Ziele des Projekts zu verstehen und Stolpersteine zu vermeiden. Hier sind einige Fragen, die Sie in diesem Zusammenhang stellen sollten: Warum wurde dieses Projekt veranlasst? Wer hat es veranlasst? Gab es bereits fehlgeschlagene Versuche, das Projektziel zu erreichen? Wenn ja, wer war daran beteiligt? Weshalb sind diese Ansätze fehlgeschlagen? Wurde eine Voruntersuchung durchgeführt? Wenn ja, wer war in diese Untersuchung involviert und welche Ergebnisse hat sie hervorgebracht (potenzielle Anforderungsquellen!)? Wenn Sie die Vorgeschichte eines Projekts verstehen, hilft Ihnen das nicht nur beim Definieren des Ermittlungsziels und beim Identifizieren der richtigen Anforderungsquellen, sondern auch bei den Lösungsaktivitäten (Identifizieren eines Konflikts, Verstehen eines Konflikts, Kennen der Wurzeln eines Konflikts).

Sie sollten selbstverständlich auch mit den Projektinitiatoren, mit Projektleitern, Projektmitgliedern und allen Personen sprechen, die Ihnen gegebenenfalls Informationen zum Projekt liefern können (weitere Details zur Stakeholder-Analyse finden Sie in Kapitel 2). Wer sind die „wichtigen“ Personen in Ihrem Projekt? Wer weiß was über das Projekt, über die verwendeten Technologien, die Domäne usw.? Welche Ziele werden mit dem Projekt verfolgt? Wie sieht der Zeitrahmen aus? Wie ist das Projekt organisiert? Wer ist wofür verantwortlich? Dies sind nur einige Fragen, die Ihnen dabei helfen, ein Entwicklungsprojekt und dessen Kontext zu verstehen. Außerdem sollten Sie herausfinden, ob es bestimmte **Projektrandbedingungen** gibt, die den Ermittlungsansatz beeinflussen können.

Zudem sollten Sie versuchen, ein klares Bild davon zu erhalten, wie **komplex** das Projekt ist oder werden kann. Was beeinflusst die Komplexität des Projekts (Komplexitätstreiber)? Gibt es irgendwelche Anhaltspunkte dafür, dass das Projekt (viel) komplexer sein könnte als zunächst angenommen? Gibt es irgendwelche Vorschriften in Bezug auf Projektorganisation oder Arbeitsergebnisse, die erfüllt werden müssen (z. B. Automotive SPICE, GxP²)? Wie viele Personen oder Unternehmen sind in das Projekt involviert? Befindet sich das gesamte Projektteam am selben Standort, oder ist es (weltweit) verteilt? Wie sieht es mit potenziellen Stakeholdern aus? Zeichnet sich bereits ab, dass einige von ihnen schwer erreichbar sind?

Ermittlungsziel festlegen

Wir empfehlen, den Ermittlungsteil eines Projekts mit der Definition der Ermittlungsziele zu beginnen. Das Ermittlungsziel ist das zentrale Element der Ermittlungsaktivität, da es als Orientierung dafür dient, was wir über die Anforderungen oder deren Quellen erfahren möchten und welche Ermittlungstechnik wir anwenden sollten, um das Ermittlungsziel zu erreichen.

Zunächst sollte eine erste Liste erstellt werden, die Ihr Verständnis (oder das des Teams) des Projekts zeigt. Wenn sich die Projektteilnehmer auf eine Liste detaillierter Fragen einigen können, zeigt das, dass sie wissen, was sie über das Projekt erfahren möchten. Detaillierte Ermittlungsziele können in kurzfristige Ermittlungsaktivitäten überführt und nach den Projektprioritäten eingeplant werden.

Eine Liste von abstrakten oder vagen Ermittlungszielen kann ein Anzeichen für ein schwach ausgeprägtes Verständnis des angestrebten Projektergebnisses sein. Eine solche Situation ist nicht ungewöhnlich. Wählen Sie zwei oder drei Ziele aus, die Ihnen am wichtigsten erscheinen, und planen Sie eine kurzfristige Ermittlungsaktivität, um ein besseres Verständnis dieser Ziele zu erhalten. Verfeinern Sie anschließend das Ziel. Vage Ermittlungsziele, die für das Projekt sehr wichtig sind, sollten durch eine oder mehrere kurzfristige Ermittlungsaktivitäten detaillierter formuliert werden. Vage Ermittlungsziele, die für das Projekt weniger wichtig sind, sollten typischerweise in langfristige

² GxP ist eine allgemeine Abkürzung für Qualitätsrichtlinien und -vorgaben für eine gute Arbeitspraxis (engl. Good Practice). Das „x“ steht für den jeweiligen Anwendungsbereich, GLP bedeutet beispielsweise „Good Agricultural Practice“ (gute landwirtschaftliche Praxis).

Ermittlungsaktivitäten überführt und zu einem späteren Zeitpunkt im Projekt aufgegriffen werden.

Planen Sie die systematische Analyse des Systemkontextes ein.

Aus der Basisstufe wissen wir, dass der Systemkontext für die Identifizierung von Anforderungsquellen und für das Verständnis der Anforderungen essenziell ist (siehe [IREB2020]).

Deshalb ist das **Verständnis des Systemkontexts** ein äußerst empfehlenswertes Ermittlungsziel. Um ein Verständnis des Systemkontexts zu entwickeln, sind folgende Ermittlungsziele zu empfehlen:

- Personen identifizieren (Stakeholder oder Stakeholder-Gruppen), die mit dem System zusammenhängen
- Systeme in Betrieb (sonstige technische Systeme oder Hardware), die mit dem System zusammenhängen
- Dokumente identifizieren
- Prozesse (technische oder physikalische Prozesse, Geschäftsprozesse) identifizieren, in die das System involviert ist
- Ereignisse identifizieren

Systematische und pragmatische Identifizierung von (unterschiedlichen Arten von) Anforderungsquellen einplanen

Die richtigen Anforderungsquellen sind die entscheidende Ressource für eine erfolgreiche Anforderungsermittlung. Für die systematische und pragmatische Identifizierung von Anforderungsquellen ist das Verständnis der Domäne und der Organisation von essenzieller Bedeutung. Beachten Sie, dass die Genauigkeit, mit der eine Quelle definiert wird, für kurz- und langfristige Ermittlungsaktivitäten unterschiedlich ausfallen kann (siehe oben):

- Anforderungsquellen für kurzfristige Ermittlungsaktivitäten müssen benannt werden (z. B. Angabe der speziellen Stakeholder für ein Interview oder des zu analysierenden Standards). Dies ist erforderlich, damit die Ermittlungsaktivität durchführbar wird.
- Anforderungsquellen für langfristige Ermittlungsaktivitäten können auch als Arten oder Kategorien von Anforderungsquellen definiert werden.

Das Verständnis der **Organisation** und deren Kultur ist ebenfalls wichtig. Selbst zwei Unternehmen derselben Domäne können gänzlich unterschiedliche Kulturen und Arten der Herangehensweise haben. Je nach Organisation können auch zwischen Tochterunternehmen oder sogar Abteilungen einer Organisation enorme kulturelle Unterschiede bestehen. Stellen Sie sicher, dass Sie die ungeschriebenen Gesetze innerhalb der Organisation kennen. Dazu zählen Aspekte wie etwa die Art und Weise der Anrede (ist es beispielsweise üblich, jemanden mit Vornamen oder Nachnamen anzusprechen?). Sie sollten auch wissen, welche Rolle Hierarchien in einer Organisation spielen. Bis zu welcher Ebene können Sie Personen in Führungspositionen direkt einladen, und in welchen Fällen müssen Sie einen Termin mit deren Vorzimmer vereinbaren? Neben der Kultur einer Organisation sollten Sie auch deren Struktur kennen. Welche Abteilungen gibt es? Wie sind die Abteilungen miteinander verbunden? Und Sie sollten natürlich wissen, wie die Organisation

Cashflow generiert (d. h. Sie sollten wissen, was sie produziert/verkauft – selbst, wenn dies nichts mit dem Entwicklungsprojekt zu tun hat, für das Sie Anforderungen ermitteln!).

Bei der Definition von Aktivitäten relevante Prozessmuster einbeziehen

Das in diesem Abschnitt beschriebene Konzept mag auf den ersten Blick erdrückend und kompliziert erscheinen. Das ist nicht die Absicht dahinter. Das Konzept ist eine Abstraktion der besten Aspekte verschiedener Ansätze, die in der Literatur zu finden sind.

Die Literatur bietet verschiedene hoch entwickelte Methoden, welche die Ermittlung von Anforderungen unterstützen. Gängige Methoden sind beispielsweise das Human-Centered Design und Design Thinking. Bei der Anwendung unseres Konzepts können diese Methoden als eine Folge von mehreren Ermittlungstechniken betrachtet werden, und sie erfordern sehr viel Planung und Aufwand. Gleichzeitig enthält die Methodenbeschreibung in der Literatur mehrere Hinweise für die Definition von Ermittlungszielen und Anforderungsquellen.

Wenn Sie eine Methode verwenden, empfehlen wir dringend, dass Sie mehr als eine Ermittlungsaktivität planen. Unterschiedliche Aktivitäten können beispielsweise verschiedenen Techniken entsprechen, die zur selben Methode gehören.

Um Sie bei der Anwendung der Methoden aus der Literatur zu unterstützen, haben wir das Konzept von Prozessmustern in der Anforderungsermittlung entwickelt. In Abschnitt 1.4 wird dieses Konzept als separates Thema vorgestellt.

Zeit und Budget für Konfliktlösungsaktivitäten einrechnen

Auch wenn es zu Beginn eines Projekts möglicherweise keine Konflikte gibt, sollten Sie Zeit für Konfliktlösungsaktivitäten einplanen. Nicht jeder Konflikt ist ein Hindernis, aber es wird Konflikte geben, die Sie lösen müssen:

- Rechnen Sie Zeit ein, um aktiv nach Konflikten zu suchen; je früher Sie es tun, umso größer ist die Wahrscheinlichkeit, dass Sie die Zeit finden, die Konflikte zu lösen.
- Planen Sie Zeit für die Lösung neuer Konflikte ein, bevor Sie konkrete Anhaltspunkte für deren Existenz haben.
- Planen Sie die Konfliktanalyse so frühzeitig wie möglich ein, da Sie daraus viele nützliche Erkenntnisse ziehen können. Außerdem können Sie so gegebenenfalls verhindern, dass Sie sich später mit unkooperativen Stakeholdern beschäftigen müssen.

Entscheiden Sie sich für schnelle Erfolge. Wenn Sie vermeiden können, dass ein Konflikt entsteht, oder wenn Sie eine schnelle Lösung finden (selbst wenn etwas nicht explizit geplant war), sollten Sie diese Gelegenheit nutzen. Möglicherweise haben Sie all Ihre Stakeholder in absehbarer Zeit nicht mehr zusammen in einem Workshop.

1.3.3.3 Richtlinien für die Durchführungsphase

In der *Durchführungsphase* geht es um die Durchführung von Ermittlungsaktivitäten. In dieser Phase werden die Aktivitäten zur Anforderungsermittlung gemäß dem vorhandenen Plan durchgeführt.

Ermittlungsaktivitäten als Tätigkeiten betrachten, die in begrenzten Zeitfenstern (Timeboxes) stattfinden

Ein schwerwiegendes Risiko bei der Anforderungsermittlung ist, dass man Zeit und Ressourcen auf ineffiziente Tätigkeiten verschwendet, die nicht die erwarteten Ergebnisse erbringen. Daher empfehlen wir, für Ermittlungsaktivitäten begrenzte Zeitfenster, sogenannte Timeboxes, vorzusehen, wobei in jeder Timebox iterativ und inkrementell Informationen zur Anforderungsmenge hinzugefügt werden.

Wenn eine Aktivität die Erwartungen an den Aufwand und/oder Zeitplan nicht erfüllt, stoppen Sie diese und untersuchen Sie die erreichten Ergebnisse, um den Grund für den Misserfolg zu verstehen. Der potenzielle Misserfolg einer Ermittlungstechnik kann mehrere Gründe haben: Beispielsweise passt die Konfliktlöstechnik nicht zur Konfliktsituation, oder die genutzten Anforderungsquellen können keine adäquaten Informationen bieten. Die Ergebnisse dieser Untersuchung können verwendet werden, um neue Aktivitäten zu planen oder vorhandene Aktivitäten zu verfeinern.

Den Plan nach jeder durchgeführten Aktivität hinterfragen (und anpassen, sofern nötig)

Den vorhandenen Ermittlungsplan sollte man nicht als in Stein gemeißelt sehen. Der Plan wurde, einschließlich der Annahmen, mit den zum damaligen Zeitpunkt verfügbaren Informationen erstellt. Der Erhalt neuer Informationen, das Auftreten von Anforderungskonflikten und die Widerlegung von vorhandenen Annahmen gehören bei der Anforderungsermittlung zum Tagesgeschäft.

Nutzen Sie das bei den laufenden Aktivitäten erworbene Wissen über die Organisation, die Stakeholder und die Projektkomplexität, und passen Sie Ihren Plan kontinuierlich an. Konzentrieren Sie sich nicht ausschließlich auf die ermittelten Anforderungen, sondern beziehen Sie auch alle anderen Informationen mit ein. Die folgenden Fragen können Ihnen als Checkliste dienen, anhand der Sie Ihren Plan nach jeder Ermittlungsaktivität überprüfen können:

- Wirken sich die Ergebnisse einer Aktivität auf die definierten kurzfristigen Aktivitäten aus?
- Sind die Ziele der zugehörigen Aktivität immer noch gültig?
- Lassen sich die Ergebnisse einer Aktivität zur Verfeinerung vorhandener vager oder abstrakter Ziele verwenden?
- Können vorhandene langfristige Ziele überarbeitet werden?
- Führen die Ergebnisse einer Aktivität zu neuen Zielen?
- Deuten die Ergebnisse auf einen neuen Anforderungskonflikt hin?
- Sind die Ergebnisse für die Lösung vorhandener Anforderungskonflikte nützlich?

Defensiv planen und kurz- und langfristige Ermittlungsaktivitäten nutzen

Neben der präzisen Definition der Ermittlungsaktivitäten, ist die richtige Ausführungsfolge ein weiterer wichtiger Faktor für eine erfolgreiche Anforderungsermittlung. Vor allem in Projektsituationen mit vielen Unsicherheiten empfehlen wir einen defensiven Ansatz bei der

Einplanung von Ermittlungsaktivitäten, um eine Verschwendung von Ressourcen zu vermeiden.

Die zeitliche Planung der Ermittlungsaktivitäten ist von folgenden Faktoren abhängig:

- Verfügbarkeit von Stakeholdern/Anforderungsquellen: Eine Aktivität kann nur durchgeführt werden, wenn die erforderlichen Stakeholder (für Ermittlungsaktivitäten) oder Anforderungsquellen (für Ermittlungsaktivitäten) verfügbar sind.
- Verfügbarkeit von Requirements Engineers: Eine Aktivität kann nur durchgeführt werden, wenn für die Durchführung der Aktivität kompetente Requirements Engineers verfügbar sind.
- Wert für das Projekt: Aktivitäten, die einen hohen Wert für das Projekt haben, sollten priorisiert werden. Die Definition von Wert hängt von der spezifischen Situation ab. Es sind beispielsweise folgende Wertdefinitionen denkbar:
 - die Bedeutung der Anforderungen, die eine Ermittlungsaktivität erbringt (durch das Ermittlungsziel vorgegeben)
 - die Bedeutung des zu lösenden Konflikts
 - die Bedeutung der gelieferten Informationen für weitere Ermittlungsaktivitäten (z. B. eine Ermittlungsaktivität wurde definiert, um ein vages, aber wichtiges Ermittlungsziel präziser zu formulieren)

Puffer einbauen, um Zeit für Kreativität, die Konfliktlösung und unerwartete Ereignisse zu haben

Seien Sie sich der Tatsache bewusst, dass Sie nicht alles vorab planen können, da Sie mit vielen Unsicherheiten umgehen müssen. Seien Sie nicht überrascht, wenn im Projektverlauf dringende neue Aktivitäten auftauchen. Wenn Sie dafür keinen Puffer eingeplant haben, werden Ihre anderen geplanten Aktivitäten leiden und es wird kein Raum für Kreativität bleiben. In der frühen Phase eines Ermittlungsprojekts sind 25 % Puffer nicht ungewöhnlich. Dieser Prozentsatz kann später verringert werden, er wird jedoch niemals bei null liegen.

Voneinander unabhängige Aktivitäten parallelisieren

Durch das Parallelisieren voneinander unabhängiger Aktivitäten können Sie Ermittlungsaktivitäten effizienter gestalten. Zwei oder mehr Ermittlungsaktivitäten werden unabhängig voneinander betrachtet, wenn die Ermittlungsziele voneinander unabhängig und die Anforderungsquellen nicht identisch sind. Es ist nicht ratsam, Aktivitäten zu parallelisieren, die abhängige (oder gar dieselben) Ermittlungsziele haben, da sich das Ergebnis von einer Aktivität auf die andere Aktivität auswirken kann.

Zwei oder mehr Lösungsaktivitäten werden unabhängig voneinander betrachtet, wenn die widersprüchlichen Anforderungen voneinander unabhängig und die widersprüchlichen Beteiligten nicht identisch sind. Es ist nicht ratsam, Aktivitäten zu parallelisieren, die wechselseitig abhängige Konflikte aufweisen, da sich die Lösung von einem Konflikt auf den anderen auswirken könnte.

Stellen Sie jedoch sicher, dass Sie sämtliche parallelisierten Aktivitäten regelmäßig koordinieren. Anderenfalls entgehen Ihnen möglicherweise verborgene Abhängigkeiten oder Erkenntnisse, die für die anderen Aktivitäten nützlich sein könnten.

Ermittlungsaktivitäten kombinieren, die dieselbe Anforderungsquelle nutzen

Ermittlungsaktivitäten, die dieselbe Anforderungsquelle nutzen (z. B. denselben Stakeholder oder dieselbe Stakeholder-Gruppe), können zur Effizienzsteigerung kombiniert werden.

Für den Fall, dass wir für drei verschiedene Ermittlungsziele Interviews nutzen möchten und der Stakeholder für diese Ziele identisch ist, können wir ein Interview einplanen und zu diesem Termin alle drei Ziele bearbeiten.

Konflikte suchen und nach einem vereinbarten Vorgehen darauf reagieren

Das aktive Suchen nach Konflikten ist eine tägliche Aufgabe. Unabhängig davon, ob Sie Ermittlungsaktivitäten, Spezifikations-Reviews, Qualitätssicherungsmaßnahmen für ein Anforderungsmodell oder irgendeine andere Aufgabe durchführen: Sie sollten gute Fertigkeiten dafür entwickeln, diese jederzeit auf Konsistenz prüfen zu können.

Dokumentieren Sie Informationen zu potenziellen Konflikten, beispielsweise Anzeichen für Konflikte in Ihrer Stakeholder-Liste, für ein späteres Review. Achten Sie in Bezug auf soziale oder gemischte Konflikte auf die Indikatoren. Nicht jeder Indikator führt unbedingt zu einem realen Konflikt, aber es gibt für jeden Konflikt Indikatoren. Es ist immer hilfreich zu wissen, was man im Blick behalten sollte, bevor es zu spät ist zu reagieren.

Bleiben Sie in einer Konfliktsituation immer neutral, und versuchen Sie, eine nachhaltige Lösung für das gesamte Projekt und alle Stakeholder zu finden. Es ist auch hilfreich, wenn Sie in einer Konfliktsituation oder in einem Workshop die Rolle des Vermittlers übernehmen, so dass Sie als neutral wahrgenommen werden.

Nachfolgend finden Sie einige Beispiele dazu, wie Sie Konflikte in der Ausführungsphase aufspüren können:

- Durchsuchen Sie Ihr Glossar- oder Terminologiemodell nach inkonsistenten Begriffen.
- Zählen Sie: Wenn Ihre Benutzeroberfläche fünf Felder hat und Sie sechs Feldbezeichner haben, muss ein Fehler vorliegen!
- Wenn Stakeholder A gerne eine rote Schaltfläche hätte, kann dieselbe Schaltfläche für Stakeholder B nicht grün sein.
- Wenn Sie ein Verhalten für eine Bedingung definiert haben, haben Sie auch ein alternatives Verhalten festgelegt, falls die Bedingung nicht anwendbar ist?

1.4 Prozessmuster

Aus der Branchenpraxis wissen wir, dass jedes Projekt einzigartig ist. Daher ist es eine Herausforderung, für das Durchführen der Anforderungsermittlung konkrete Orientierungshilfen zu geben, da zahlreiche Faktoren den bestmöglichen Ansatz beeinflussen können. Einige dieser Faktoren sind die verfügbare Zeit und das Budget, die Art

des zu entwickelnden Systems, die Verfügbarkeit der Stakeholder und die Erfahrung der Requirements Engineers.

Dennoch sind in der Literatur und aus der Branchenpraxis bestimmte Methoden bekannt, deren Nützlichkeit sich in verschiedenen Situationen bewährt hat. Zusätzlich zur Beschreibung einiger dieser Methoden, stellen wir das Konzept der Prozessmuster vor. Das Konzept von Mustern wurde ursprünglich im Kontext der Architektur [ALLS1977] entwickelt, um wiederverwendbares Wissen für die Erstellung von Architekturen zu dokumentieren. Dies wurde später mit der Einführung von Entwurfsmustern für Software auf die Softwareentwicklung übertragen [GHJV1994].

Im Kontext der Anforderungsermittlung ermöglichen Prozessmuster eine Form der Dokumentation nützlicher und bewährter Möglichkeiten zum Durchführen der Anforderungsermittlung. Das Hauptziel dieses Abschnitts ist es, die zugrunde liegende Idee von Prozessmustern als Instrumentarium zu vermitteln, und nicht, bestimmte Ansätze zu lehren. Zunächst wird die Struktur von Prozessmustern für die Anforderungsermittlung beschrieben. Anschließend wird diese anhand einiger Beispiele von Prozessmustern verdeutlicht.

1.4.1 Struktur und Vorteile von Prozessmustern für die Anforderungsermittlung

Ein Prozessmuster besteht aus den folgenden Elementen:

- *Geltungsbereich*: In diesem Abschnitt wird die Projektsituation bzw. die Projektsituationen beschrieben, die sich für die Anwendung eines Musters eignen. Diese Beschreibung kann auch bestimmte Situationen enthalten, in denen die Anwendung des Musters nicht ratsam ist.
- *Erforderlicher Aufwand/Ressourcen*: In diesem Abschnitt werden der Aufwand und die Ressourcen beschrieben, die für die Anwendung bestimmter Muster nötig sind. Der Aufwand wird üblicherweise in Bezug auf die zeitlichen und personellen Ressourcen beschrieben. Zusätzliche Ressourcen können beispielsweise Workshop-Materialien, spezielle Software oder spezielle Standorte beinhalten.
- *Musterelemente*: In diesem Abschnitt werden die Details des Musters erläutert. Die Beschreibung beinhaltet Verweise auf angewendete Methoden und konkrete Beschreibungen von Ermittlungsaktivitäten, sodass der Leser das Muster in Bezug auf die Anforderungsermittlung versteht.
- *Instanziierung*: Dieser Abschnitt enthält spezifische Details zum Einstieg. Die Beschreibung beinhaltet Aktivitäten für die Vorbereitung des Musters und einen initialen Plan mit Ermittlungsaktivitäten (siehe Abschnitt 1.3.1.1), die für den Anfang als Richtlinie verwendet werden kann.
- *Weitere Lektüre*: In diesem Abschnitt erhalten Sie zusätzliche Verweise auf nützliche Literatur mit weiterführenden Details zum Muster.

Abhängig vom Ursprung der zugrunde liegenden Ideen des Musters, sind die beschriebenen Details nicht zwingend auf die Anforderungsermittlung beschränkt und können auch andere

Aktivitäten, wie Design oder Testen, umfassen. Solche nicht auf die Anforderungsermittlung bezogenen Bestandteile sollten kurz gehalten werden. Wenn möglich, enthält die Beschreibung die Ergebnisse solcher nicht auf die Anforderungsermittlung bezogenen Aktivitäten für die nachfolgende Verwendung in der Anforderungsermittlung.

Requirements Engineers sollten aktiv nach Prozessmustern suchen, die für ihre eigene Situation relevant sind. Beachten Sie Folgendes:

- Prozessmuster sind bewährte Verfahren aus Literatur und Praxis, die als Ausgangsbasis dafür dienen, Ermittlungsaktivitäten in vergleichbaren Situationen festzulegen.
- Prozessmuster bieten eine Metaperspektive auf die Arbeitsweise. Anstatt denselben Ansatz in allen Projektsituationen anzuwenden, erlauben Prozessmuster die Entwicklung eines Instrumentariums, das über bestimmte Techniken (z. B. Interviews) hinausgeht.
- Für gewöhnlich sind die verfügbaren Informationen nicht ausreichend, um das Prozessmuster direkt anwenden zu können. Daher muss analysiert werden, 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 in diesem Kapitel abgedeckten 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.

In diesem Abschnitt werden drei Beispiele für Prozessmuster dargestellt: der Wasserfall-Ansatz, Human-Centered Design und Design Thinking. Der Wasserfall wird als ein Muster ausgewählt, da das Wasserfall-Vorgehensmodell häufig mit dem Requirements Engineering als Methode verwechselt wird. Diese Beschreibung soll verdeutlichen, dass der Wasserfall-Ansatz nur eine Möglichkeit ist, mit der sich das Requirements Engineering/die Anforderungsermittlung durchführen lässt.

Human-Centered Design und Design Thinking werden hier behandelt, weil es sich dabei um zwei weit verbreitete Ansätze für die Ermittlung von Anforderungen handelt. Außerdem vereinen sie verschiedene Techniken (z. B. Workshops und Kreativitätstechniken) mit einem Vorgehensmodell, was eine einfache Anwendung in diversen Situationen ermöglicht.

Agile Entwicklung wurde hier bewusst nicht als Muster aufgenommen, da die Details zu Requirements Engineering in agilen Projekten im IREB CPRE RE@Agile abgedeckt werden.

1.4.2 Das Wasserfall-Modell

Das Wasserfall-Modell beschreibt ein linear-sequenzielles Lebenszyklusmodell der Softwareentwicklung, in dem die Entwicklung eines Systems eine Folge von verschiedenen Phasen durchläuft. Winston W. Royce [Royce70] beschrieb dieses Modell zuerst mit den Phasen „Anforderungen“, „Analyse“, „Architekturentwurf“, „Programmierung“, „Test“ und

„Betrieb“. Daneben wurden auch andere Phasendefinitionen veröffentlicht, z. B. der Standard DOD-STD-2167A des US-Verteidigungsministeriums [DoD88]. Das V-Modell [FoMo91] oder das Spiralmodell von Boehm [Boehm88] sind Varianten des Wasserfall-Lebenszyklus.

1.4.2.1 Geltungsbereich

Dieser Art von Entwicklung begegnet man überwiegend in großen, langfristigen Projekten, in denen (eingebettete) technische Systeme oder komplexe Systeme mit Schnittstellen zu anderen (Alt-)Systemen entwickelt werden und die einer Big-Bang-Einführungsstrategie folgen.

Die bekanntesten Merkmale dieser Art von Projekten sind:

- ein klar definiertes Geschäftsziel
- ein präzises Verständnis des zu entwickelnden Systems
- ein präzises Verständnis der Domäne
- ein präzises Verständnis der Technologie, die eingesetzt werden soll
- starke Anweisungen von einem einzigen Business Owner oder Lenkungsausschuss
- ein strikter Ansatz, der von Anfang an in einem formalen Plan mit Budgets und Zeitrahmen definiert und von einem starken Projektmanagement gesteuert wird
- aus unterschiedlichen Experten bestehende Teams, die an den einzelnen Phasen arbeiten
- Ergebnisse der einzelnen Phasen, die als Input für die nächste Phase dienen, wobei Informationen über formale Dokumentation weitergegeben werden
- Quality Gates zwischen den einzelnen Phasen, bei denen nach formalen Managemententscheidungen vorgegangen wird
- Nachdem ein Quality Gate erfolgreich durchlaufen wurde, werden die resultierenden Anforderungen in Form von Spezifikationen „eingefroren“ und wenn nachfolgend Änderungen erforderlich sind, werden diese nach einem formalen Änderungsverfahren ausgeführt.

1.4.2.2 Aufwand/Ressourcen

Wasserfall-Projekte laufen gewöhnlich über einen langen Zeitraum (mehrere Monate bis Jahre) und erfordern einen großen Aufwand (häufig mehrere hundert Personenmonate). Zu Beginn des Projekts wird ein detaillierter Plan für den Rest des Projekts erstellt und eine Einigung darüber erzielt. Dieser Plan umfasst eine anfängliche Anforderungsphase, die von einigen Wochen bis zu mehreren Monaten dauern kann und an der mehrere Experten beteiligt sein können.

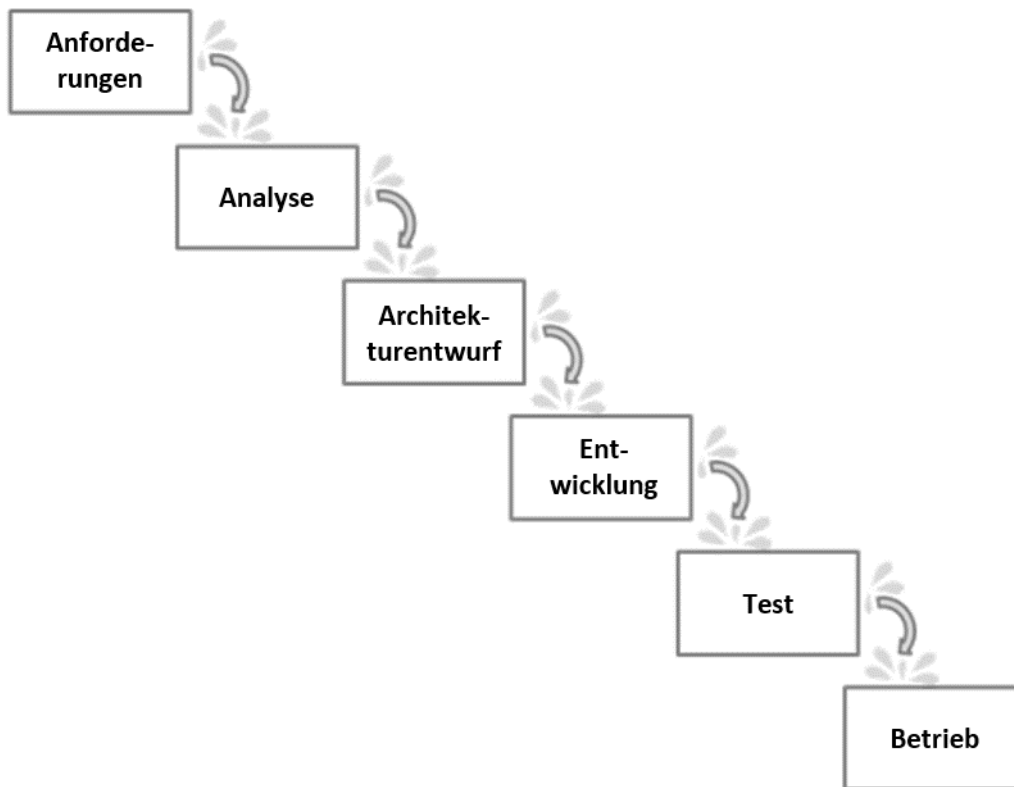


Abbildung 4: Der Wasserfall-Prozess

1.4.2.3 Musterelemente

Für das Wasserfall-Modell gibt es verschiedene Phasendefinitionen, bei denen es sich in der Regel um Varianten der ursprünglichen Definition von Royce handelt (siehe Abbildung 4). Aus Sicht des Requirements Engineering wird die Anforderungsermittlung als wichtigste Phase betrachtet. In dieser Phase werden die zentralen Anforderungen ermittelt und damit die Gesamtausrichtung für das restliche Projekt festgelegt. Das Requirements Engineering spielt in allen nachfolgenden Phasen eine Rolle, auch wenn es hier weniger ins Gewicht fällt.

In den folgenden Absätzen werden die einzelnen Phasen detaillierter beschrieben. Sie enthalten auch Richtlinien für Aktivitäten der Anforderungsermittlung.

Anforderungen

Ziel dieser Phase ist es, das vom Business Owner formulierte allgemeine Geschäftsziel in eine gültige Liste von wichtigen (geschäftlichen) Anforderungen umzusetzen. Zudem werden die wichtigsten Randbedingungen ermittelt, die den Lösungsraum für das System und das Projekt definieren.

Requirements Engineers ermitteln Anforderungen in der Regel, indem sie mit einer (gewöhnlich kleinen) Gruppe von Führungskräften und Managern qualitative Interviews durchführen, die Ergebnisse in Berichten zusammenfassen und Feedback verarbeiten. Es kann schwierig sein, ausreichend Zeit mit den Interviewpartnern zu erhalten, da diese meist

sehr volle Terminkalender haben. Neben Interviews kann eine Klärung der Anforderungen auch durch die Analyse von Dokumenten erreicht werden.

Die Ergebnisse werden in einem detaillierten Endbericht konsolidiert, der aus einer Reihe von wiederholt überarbeiteten Entwurfsversionen entsteht. Konflikte werden gelöst, indem das Feedback so lange verarbeitet wird, bis ein Konsens erreicht ist, oder indem das übergeordnete Management eine Entscheidung trifft. Am Ende der Phase dient eine formale Prüfung der neuesten Entwürfe als Quality Gate. Danach wird der Endbericht als Input für die nächste Phase freigegeben.

Analyse

Die Analysephase konzentriert sich auf die weitere Ausarbeitung der Informationsarchitektur, der Benutzungsschnittstelle und der Schnittstellen mit anderen Systemen zu einer Reihe von Systemanforderungen.

Eine wesentliche Aufgabe ist die Ermittlung der Randbedingungen, die aus der umgebenden IT-Landschaft entstehen. Dies geschieht hauptsächlich durch die Analyse von Dokumenten und durch Interviews mit Systemadministratoren und anderen IT-Experten. Auf der Benutzerseite verschiebt sich der Schwerpunkt zu den direkten Endbenutzern. Je nach Größe und Komplexität der Benutzer-Community können für die Anforderungsermittlung sowohl Interviews als auch Fragebogen verwendet werden. Auch Workshops mit Benutzergruppen können relevant sein. Ist die Benutzer-Community vielfältig, können Personas hilfreich sein. Das „Vergessen“ externer Benutzer und anderer indirekt betroffener Kunden, kann ein Stolperstein sein, vorausgesetzt, dass die internen Benutzer ihre Meinung ausreichend darstellen.

Es sei nochmals darauf hingewiesen, dass diese Phase in einem konsolidierten und validierten Endbericht resultiert, der vor der Freigabe ein Quality Gate durchläuft.

Design, Gestaltung

In der Designphase werden die Systemanforderungen zu Softwareanforderungen ausgestaltet, die als Plan für das zu entwickelnde System dienen.

In dieser Phase sind IT-Experten tonangebend: Functional und Technical Designer, Datenbank- und Systemadministratoren, Technical Operators. Schnittstellen werden detaillierter definiert und von Vertretern des Benutzerkreises wird Feedback eingeholt und verarbeitet. Hierbei können Low-Fidelity-Prototypen nützlich sein. Zur Ausrichtung des designten Systems auf die betrieblichen Prozesse des Endbenutzers können beobachtende Techniken angewendet werden. Die Abstimmung von Qualitätsanforderungen mit technischen Randbedingungen kann eine wichtige Aufgabe darstellen.

Am Ende dieser Phase wird die gesamte Konstellation aus Geschäfts-, System- und Softwareanforderungen zu einer umfangreichen und komplexen Sammlung angewachsen sein, die aktiv gemanagt werden muss (Änderungs- und Konfigurationsmanagement, Basislinien, Versionierung, Priorisierung, Verfolgbarkeit). Die Aufrechterhaltung der Konsistenz sämtlicher Anforderungen ist eine wichtige Angelegenheit.

Aus der Designphase gehen ein oder mehrere Dokumente hervor, die vor der Freigabe ebenfalls ein Quality Gate durchlaufen. Findet ein Outsourcing statt, bilden formale Design-Dokumente die Grundlage des Vertrages mit dem Outsourcing-Dienstleister. In solchen Fällen spielt die Qualität der Anforderungen in den Design-Dokumenten für den Projekterfolg eine entscheidende Rolle.

Codierung

Diese Phase soll zur Entwicklung einer technischen Lösung dienen, welche die Anforderungen erfüllt, die in den vorangehenden Phasen formuliert wurden. Darin sind IT-Entwickler wie etwa Technical Designer, Datenbankexperten und Programmierer involviert.

Theoretisch werden keine neuen Anforderungen entwickelt. In der Praxis können Entwickler auf Inkonsistenzen zwischen Anforderungen, Konflikte mit übersehenen Randbedingungen einer detaillierteren Ebene oder Fehlinterpretationen und Fehlberechnungen aus früheren Phasen stoßen. Gelegentlich werden frühzeitig in dieser Phase High-Fidelity Prototypen verwendet, um Feedback von Endbenutzern einzuholen.

Komponenten- und Integrationstests dienen als Quality Gates für die nächste Phase.

Test

In der Testphase versuchen unabhängige Tester, Endbenutzer, Systemadministratoren und Betreiber zu verifizieren, ob das System die definierten Anforderungen erfüllt, und zu validieren, ob das System die geplanten Geschäftsprozesse ohne größeres Risiko unterstützt. Kurz zur Wiederholung: Es werden keine neuen Anforderungen erwartet, aber für die Lösung von Defekten ist es eventuell erforderlich, dass vorherige Lösungen neu bewertet und angepasst werden.

Die Testphase bildet das finale Quality Gate. Sie stellt die Informationen bereit, welche die Geschäftsleitung benötigt, um gute Entscheidungen zu treffen.

Betrieb

In der Betriebsphase wird das System in einer geschäftlichen Livesituation verwendet.

Dabei können Vorfälle im operativen System auftreten und zu Änderungen in der Geschäftsumgebung führen, was wiederum neue oder aktualisierte Anforderungen zur Folge hat. Über Änderungsanträge wird anhand von Auswirkungsanalysen entschieden. Es ist wichtig, alle akzeptierten Änderungen zu berücksichtigen und sicherzustellen, dass während des gesamten Systemlebenszyklus nur eine konsistente, aktuelle Konstellation von Anforderungen gepflegt wird.

1.4.2.4 Instanziierung dieses Musters

Im Wasserfall-Muster bedarf es einer sehr hohen Ergebnisqualität. Niedrige Qualität in frühen Phasen führt unweigerlich zu Projekten mit Zeit- und Kostenüberschreitungen und Systemen, die den Kundenerwartungen nicht entsprechen.

Quality Gates am Ende jeder Phase dienen als Schutz vor niedriger Qualität. In den Anfangsphasen stellen Reviews und Inspektionen die zentralen Techniken dar, um die Qualität auf einem hohen Niveau zu halten. Es hat sich bewährt, nicht nur Teilnehmer aus der aktuellen Phase, sondern auch aus vorherigen und nachfolgenden Phasen miteinzubeziehen. Nachdem das Quality Gate erfolgreich durchlaufen wurde, werden die Anforderungen für die nachfolgenden Phasen in Form von Spezifikationen „eingefroren“. Das bedeutet Folgendes: Wenn jemand eine Anforderung später ändern möchte, muss ein formales Änderungsverfahren befolgt werden, bei dem die Auswirkung der Änderung analysiert und bewertet wird. In der Praxis bewirkt dies häufig, dass man von Änderungen absieht. Auf diese Weise fungieren Spezifikationen aus früheren Phasen als Randbedingungen für die nächsten Phasen. Das ist gleichzeitig eine Stärke und eine Schwäche des Wasserfall-Musters – da es einerseits der Stabilität dient und andererseits unflexibel macht. Daher eignet sich dieses Modell am besten für stabile Geschäftsumgebungen, in denen ein klares Ziel mit intensiver Kontrolle umgesetzt werden muss, das zudem auditierbar ist.

Wenn Sie in einem Wasserfall-Projekt mit dem Requirements Engineering beginnen, konzentrieren Sie sich auf die Anforderungsphase. Auch wenn dies scheinbar gegen den Charakter eines solchen Projekts spricht, sollten Sie einen iterativen Timebox-Ansatz in Betracht ziehen, bei dem das oberste Geschäftsziel schrittweise zu einer Reihe von Geschäftsanforderungen verfeinert wird. Die Timeboxes sollten in die Gesamtprojektplanung passen, damit man rechtzeitig einen Endbericht für den Beginn der Analysephase erreicht.

Die folgende Tabelle enthält eine beispielhafte Liste mit Zielen.

Nr.	Ermittlungsziel	Ermittlungstechnik	Anforderungsquelle	Ergebnisqualität
1	Relevante Stakeholder und Dokumentation identifizieren	Interview	Business Owner	Mittel
2	Auf Interviews vorbereiten	Dokumentanalyse	Relevante Domäne und IT-Dokumente	Mittel
3	Geschäftliche Anforderungen verstehen	Interview	Relevante Führungskräfte	Mittel
4	Feedback einholen	Verteilung, um Kommentare zu erhalten	Relevante Führungskräfte	Mittel – Hoch
Schritte 2 bis 4 wiederholen, bis das Feedback eine zufriedenstellende Qualität belegt				
5	Gesamtbild konsolidieren	Workshop	Business Owner, Interviewpartner, andere Stakeholder	Hoch

6	Ergebnisse validieren	Inspektion	Relevante Stakeholder und Vertreter der nächsten Phase	Hoch
---	-----------------------	------------	--	------

7	Abnahme erreichen	Endbericht	Business Owner und relevante Stakeholder	Hoch
---	-------------------	------------	--	------

In den nächsten Phasen kommt dem Requirements Engineer nicht unbedingt eine führende Rolle zu. Detaillierte System- und Softwareanforderungen werden gemeinsam mit anderen Experten entwickelt und verfeinert. Das wichtigste Anliegen des Requirements Engineers ist das Management der zunehmenden Menge von Anforderungen. Der Requirements Engineer fungiert oftmals als Berater und Coach bezüglich Problemen mit Anforderungen, der Beseitigung von Konflikten und der Wahrung der Konsistenz.

Stellen Sie sicher, dass in diesen Phasen genügend Requirements-Engineering-Kapazitäten vorhanden sind, vor allem während der (Abnahme-) Tests. Requirements Engineers sollten hierbei eine aktive Rolle übernehmen, um sicherzustellen, dass das System sich gemäß den festgelegten Anforderungen verhält.

1.4.2.5 Weiterführende Literatur

- [Royce70] W. Royce: Managing the Development of Large Software Systems. In Proceedings of IEEE WESCON 26 (August): 1-9, 1970
- [DoD88] United States Department of Defense: DOD-STD-2167A, MILITARY STANDARD: DEFENSE SYSTEM SOFTWARE DEVELOPMENT, 1988
- [FoMo91] K. Forsberg, H. Mooz: The Relationship of System Engineering to the Project Cycle. In: Proceedings of the First Annual Symposium of National Council on System Engineering, October 1991: 57-65.
- [Boehm88] B.W. Boehm: A Spiral Model of Software Development. In Tutorial: Software Engineering Project Management. Edited by R.H. Thayer, IEEE Computer Society Press, Washington D.C., 1988, pp. 128-142.

1.4.3 Human-Centered Design

Human-Centered Design (HCD), auch bekannt als User-Driven Development (UDD), ist ein Framework von Prozessen (nicht beschränkt auf Schnittstellen oder Technologien), in welchem dem Kontext der Verwendung, den Usability-Zielen, den Benutzereigenschaften, dem Umfeld, den Aufgaben und dem Workflow eines Produkts, Services oder Prozesses in jeder Phase des Designprozesses umfassende Aufmerksamkeit gewidmet werden [RiFi2014].

Die Benutzbarkeit eines Systems ist das Ausmaß, in welchem das System von bestimmten Benutzern verwendet werden kann, um festgelegte Ziele in einem bestimmten Verwendungskontext effektiv, effizient und zufriedenstellend zu erreichen [ISO9241.11].

Aus der vorgenannten Definition wird klar, dass „Benutzbarkeit“ nicht etwas ist, das man einfach in einem Interview erwähnen und dann als nicht-funktionale Anforderung definieren kann, z. B. „das System muss benutzbar sein“. Die Benutzbarkeit eines Systems hängt von den Aufgaben der Benutzer und dem Kontext der Verwendung ab. Human-Centered Design ist ein Ansatz zur Identifizierung dieser wechselseitig abhängigen Elemente:

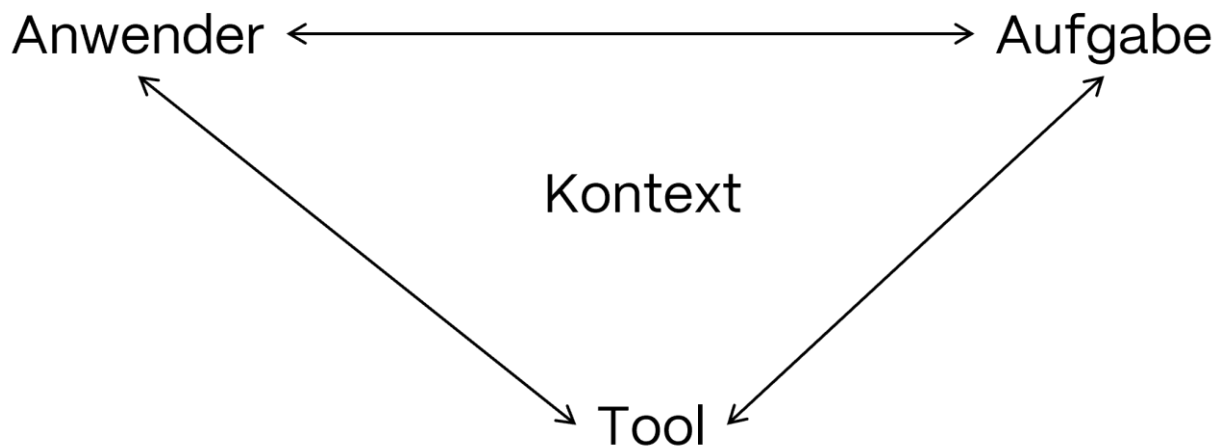


Abbildung 5: Wechselseitig abhängige Elemente, die im HCD zu berücksichtigen sind [Shackel1991]

User Experience (UX) bezieht sich auf das Konzept der Multi-Channel-Kontakte zwischen einem Kunden und einem Anbieter. Der Kunde erlebt eine sogenannte Customer Journey, die mit dem ersten Kontakt mit dem Anbieter beginnt (wahrscheinlich über Werbung oder einen Vertriebsmitarbeiter), die dann über verschiedene Interaktionen mit Online- oder Offline-Kontaktpunkten verläuft (z. B. die Suche nach einem Produkt im Onlineshop, der Produktkauf, der Kontakt mit dem Kundenservice und der Bezahlvorgang), bis der Kunde die Kundenbeziehung beendet. Daher ist die Aufgabe des Anbieters nicht, die Benutzbarkeit eines bestimmten Kanals zu optimieren, sondern stattdessen die gesamte User Experience/das Kundenerlebnis zu optimieren. Im Modell von Shackel (siehe Abbildung 5) ist dies ein wichtiger Aspekt des Kontexts eines Werkzeugs: Das Werkzeug ist lediglich eines von zahlreichen Kontaktpunkten des Benutzers.

1.4.3.1 Geltungsbereich

Das Muster des HCD ist auf Projekte anwendbar, die sich durch ein hohes Maß an Interaktionen mit Benutzern auszeichnen. Wenn überwiegend Interaktionen zu anderen Systemen bestehen, ist dies möglicherweise kein geeignetes Muster. Die meisten Systeme dienen jedoch letzten Endes Menschen. Daher lohnt es sich herauszufinden, wer diese Menschen sind (Benutzergruppen), was sie tun (Aufgaben) und in welchem Kontext sie das neue System betreiben werden.

1.4.3.2 Aufwand/Ressourcen

Wenn ein großes neues System entwickelt werden soll und das Projektteam mit der Domäne noch nicht vertraut ist und/oder das Team neu zusammengestellt wurde, dann sollte im Vorfeld eine Phase mit wichtigen Überlegungen eingeplant werden (bis zu drei Monate). So erhält das Team die Gelegenheit, die Domäne kennenzulernen, mehr über die Benutzer, deren Aufgaben und den Kontext herauszufinden und wichtige Risiken zu identifizieren und zu beheben. Auch die initiale Vorbereitung der Entwicklung, z. B. der Aufbau von Entwicklungsumgebungen, kontinuierliche Integration, eine erste Version der Lösungsarchitektur usw., sollten stattfinden bzw. erstellt werden. Dazu werden auch UX-Kenntnisse benötigt; es sind umso mehr Experten erforderlich, je mehr Benutzergruppen untersucht werden müssen.

Wenn das System weniger komplex ist, das Entwicklungsteam die Domäne gut kennt und auf eine lange, erfolgreiche Zusammenarbeit zurückblicken kann, oder wenn das Projekt nur eine Erweiterung oder Instandhaltungskorrektur für ein zuvor gemäß HCD entwickeltes System ist, kann der Zeitraum für die Vorabüberlegungen auf wenige Wochen oder gar Tage verkürzt werden. In diesem Fall werden weniger Experten mit UX-Kenntnissen benötigt, wenngleich es sich in der Praxis bewährt hat, dass sich alle Mitglieder des Projektteams der Bedeutung von UX bewusst sind und sich eine HCD-Denkweise angeeignet haben.

1.4.3.3 Musterelemente

HCD kann als ein Prozess ausgeführt werden, der die gesamte Systementwicklung abdeckt – siehe ISO 9241 210, Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme:

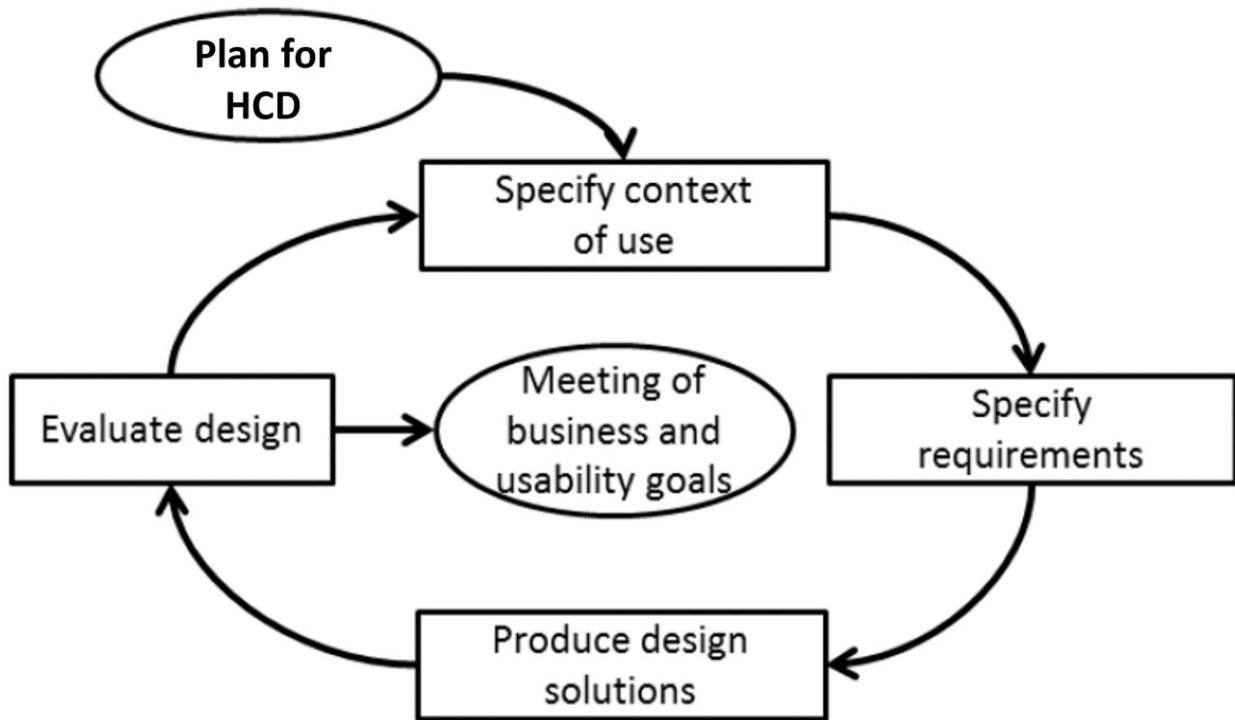


Abbildung 6: Iteratives Human-Centered Design (HCD) für interaktive Systeme [ISO9241.210]

HCD ist allerdings mehr ein Ansatz oder gar eine Einstellung als ein Vorgehensmodell. Es wird häufig in Engineering-Vorgehensmodelle integriert (z. B. agile Herangehensweisen oder Wasserfall-Methoden, siehe Abschnitt 1.4.2) oder es ist mitunter ein integraler Bestandteil des Designprozesses (z. B. Design Thinking, siehe Abschnitt 1.4.4), um Aspekte abzudecken, die den wichtigsten Stakeholder – den Benutzer – behandeln.

1. Plan für HCD („Plan for HCD“)

Planen Sie die notwendigen HCD-Aktivitäten, und stellen Sie sicher, dass sie im gesamten Projektplan etabliert sind (siehe auch nächster Abschnitt „Instanziierung“): Die Schritte werden iterativ ausgeführt, bis die Geschäfts- und Usability-Ziele erfüllt sind.

2. Nutzerforschung („Specify context of use“)

Der Requirements Engineer ermittelt und analysiert zunächst die Benutzer des zu entwickelnden Systems. Bei dieser Benutzeranalyse sollten Kontext, Merkmale, Ziele, Haltungen und Aufgaben einer angemessenen Anzahl von tatsächlichen Benutzern (nicht deren Vorgesetzten!) erfasst werden. Diese sollten anschließend in Benutzergruppen zusammengefasst und dokumentiert werden, z. B. in Form einer Persona pro Benutzergruppe.

Sobald die Benutzergruppen bekannt sind, kann eine bewusste Entscheidung getroffen werden, welche davon als primäre Benutzergruppe betrachtet wird, die durch die Lösung unterstützt wird. Das System, und besonders seine Benutzungsschnittstelle, wird für diese Benutzer optimiert. Anschließend kann entschieden werden, ob die anderen Nutzergruppen dieselbe Benutzungsschnittstelle verwenden sollen, oder ob separate Benutzeroberflächen bereitgestellt werden

müssen. Es kann auch beschlossen werden, dass einige Benutzergruppen überhaupt nicht mit dem System angesprochen werden sollen.

3. Anforderungen festlegen („Specify requirements“)

Der Requirements Engineer wählt Ermittlungstechniken aus, mit deren Hilfe Informationen über die von den Benutzern auszuführenden Aufgaben und die Arbeitsergebnisse, die sie verwenden, gesammelt werden sollen. Sind Benutzer, Aufgaben und Kontext vorhanden, können die Anforderungen für das System festgelegt werden.

4. Designlösungen erstellen („Produce design solutions“)

Bei der Validierung von Anforderungen mit Benutzern aus der Praxis sollten komplexe Modelle und aufwendige Spezifikationsdokumente, die diese nicht verstehen, vermieden werden. Es ist effizienter, Prototypen zu zeigen oder die Benutzer Szenarien oder Storyboards durchgehen zu lassen.

5. Design bewerten („Evaluate design“)

Verwenden Sie eine beliebige Form der Usability-Bewertung: Ein informeller Test (ein kurzer Systemdurchgang Ihres Prototyps mit einigen Mitarbeitern, die zufällig in der Nähe sind und 5 bis 10 Minuten Zeit erübrigen können) ist beispielsweise eine schnelle, kostengünstige und informelle Technik. Am anderen Ende der Skala gibt es formelle, Lab-basierte Usability-Tests mit einer größeren Anzahl von extern gewonnenen Benutzern.

1.4.3.4 Instanziierung

Nr.	Ermittlungsziel	Ermittlungstechnik	Anforderungsquelle	Ergebnisqualität
1	Etwas über die Aufgaben von Benutzern erfahren (zur Vorbereitung auf die Contextual Inquirys)	Interview	Spezielle Experten, die wissen, was Benutzer tun (oder tun sollten): Manager, Trainer, Verfahrenstechniker, Benutzervertreter usw.)	Hohe Sicherheit/geringes Maß an Vollständigkeit
2	Etwas über die Aufgaben von Benutzern erfahren (zur Vorbereitung auf die Contextual Inquirys)	Perspektivenbasiertes Lesen Schulungsmaterial studieren, an Kursen teilnehmen Die Systeme der Benutzer kennenlernen (Trainingssysteme, Testsysteme)	Schulungsmaterialien Trainingssysteme Testsysteme	Hohe Sicherheit/geringes Maß an Vollständigkeit
3	Zum Identifizieren von Benutzergruppen im Kontext	Contextual Inquiry (CI) Mehrere CIs	Einige bestimmte Benutzer	Mittlere Sicherheit/geringes Maß an Vollständigkeit
4	Etwas über Benutzer im öffentlichen Kontext erfahren (z. B. die Verwendung eines Informationssystems an einem öffentlichen Standort)	Feldbeobachtung	Viele Gelegenheitsnutzer	Mittlere Sicherheit/geringes Maß an Vollständigkeit
5	Quantitative Daten zu bestimmten Fragen sammeln	Befragung mit Fragebogen	Viele Benutzer an verschiedenen Orten	Hohe Sicherheit
6	Ein bestimmtes Geschäft verstehen	Apprenticing über einige Tage	Ein oder zwei bestimmte Benutzer	Hohe Sicherheit/geringes Maß an Vollständigkeit

Nr.	Ermittlungsziel	Ermittlungstechnik	Anforderungsquelle	Ergebnisqualität
7	Zum Validieren von Anforderungen und Ermitteln neuer Anforderungen	Ein User Walkthrough basierend auf: - Der Verwendung von Prototypen (Prototyping) - Anwendungsszenarien - Storyboards	Benutzer	Hohe Sicherheit/hohes Maß an Vollständigkeit

Hinweis 1.4.1:

Eine der ersten Lektionen in HCD für Entwicklungsprojektmitglieder ist die Erkenntnis: „Ich bin nicht der Benutzer“. Selbst wenn die Anwendung auf Nutzer wie mich abzielt, haben andere Benutzer eine ganz andere Einstellung, andere Erfahrungen, Ziele, Aufgaben usw., sodass es keinen Ersatz für ihr direktes Feedback gibt. Selbst ein repräsentativer Benutzer, der Teil des Projektteams wird, ist schon nach kurzer Zeit durch die Diskussionen im Entwicklungsteam „kontaminiert“ (z. B. kennt er oder sie das Konzept, das zum Design eines komplexen Bildschirmaufbau geführt hat), sodass er/sie nicht mehr wirklich repräsentativ ist.

1.4.3.5 Weiterführende Literatur

[RiFl2014] bieten einen kurzen, aber sehr praktischen Überblick über Human-centered Engineering, d. h. das Erstellen von Produkten für Menschen.

[HaPy2012] bieten einen umfassenden Überblick über UX.

[BeHo1998] prägten das Konzept von Contextual Inquiry im Contextual Design.

1.4.4 Design Thinking

Design Thinking ist ein formalisierter Prozess für die praktische und kreative Lösung von Problemen und für die Erstellung von Lösungen mit dem Ziel, in der Zukunft ein besseres Ergebnis zu erreichen. Es handelt sich dabei um eine Art lösungsbasierten oder lösungsorientierten Denkens, bei dem der Ausgangspunkt ein Ziel (eine bessere künftige Situation) anstelle der Lösung eines bestimmten Problems ist. Es gibt verschiedene Design-Thinking-Ansätze (siehe [A4qu2018]). Ein weit verbreiteter Ansatz ist d.school des Hasso Plattner Institute of Design an der Stanford University (siehe [Dsch2012]).

1.4.4.1 Geltungsbereich

Design Thinking ist nützlich für die Entwicklung von alternativen Lösungsideen und möglicherweise von Anforderungen, sei es für das Gesamtprojektziel oder für einige bestimmte Aspekte (z. B. bestimmte System-Features). Deshalb kann Design Thinking ganz

am Anfang eines Projekts eingesetzt werden (z. B. um innovative Ideen für einen Onlineshop zu entwickeln) oder auch während des Projekts bei der Ausarbeitung eines bestimmten Aspekts (z. B. ein Überblicksbildschirm für Kunden in einem System zur Richtlinienverwaltung). Eine Alternative für dieses Muster ist das Human-Centered Design, das ebenfalls in diesem Handbuch beschrieben wird.

1.4.4.2 Aufwand/Ressourcen

Design Thinking ist eine skalierbare Methode. Abhängig von den verfügbaren Ressourcen kann ein gesamter Design-Thinking-Prozess innerhalb von wenigen Stunden, Tagen oder Wochen durchgeführt werden. In der Literatur (z. B. [LiOg2011]) wird empfohlen, dass das Design-Thinking-Team in einem speziellen Raum arbeitet, da das Team verschiedene Ergebnisse produziert und je nach angewendeten Techniken Arbeitsraum benötigt (z. B. für die Entwicklung von Storyboards oder andere Arten von Canvas-Techniken).

1.4.4.3 Musterelemente

Im d.school-Ansatz besteht der Design-Thinking-Prozess aus fünf iterativen Phasen. Die Beteiligten sind Projekt-Stakeholder und Requirements Engineers. Ein Requirements Engineer ist auch für das Moderieren und Voranbringen des Prozesses verantwortlich. Beim Design Thinking wird Wert auf eine interdisziplinäre Aufstellung der Beteiligten gelegt, beispielsweise: Benutzer unterschiedlichen Alters, Vertreter mit betriebswirtschaftlichem sowie solche mit technischem Hintergrund, Personen aus anderen Gruppen, die mit dem Projekt in Zusammenhang stehen (z. B. NGOs). Aus Requirements-Engineering-Perspektive empfehlen wir die Durchführung von Ermittlungsaktivitäten zur systematischen Analyse des Projektkontexts, um eine optimale Gruppe von Stakeholdern für den Design-Thinking-Prozess zu identifizieren.

Auch wenn diese Phasen sequenziell dargestellt sind, wird beim Design-Thinking-Prozess Wert auf die Bedeutung von Iterationen gelegt. Wenn die Ergebnisse einer Phase zu einem besseren Verständnis oder zur Änderung von bereits entwickelten Ergebnissen führen, muss man im Prozess zur entsprechenden Phase zurückkehren und die Ergebnisse ändern. Wenn etwa aus der Prototyp-Phase Ergebnisse hervorgegangen sind, die eine bessere Definition des Projektziels erlauben, dann sollte man im Prozess zur Definitionsphase zurückkehren.

Nachfolgend werden die einzelnen Phasen des d.school-Ansatzes detaillierter beschrieben. Sie enthalten auch Richtlinien für das Definieren von Aktivitäten der Anforderungsermittlung.

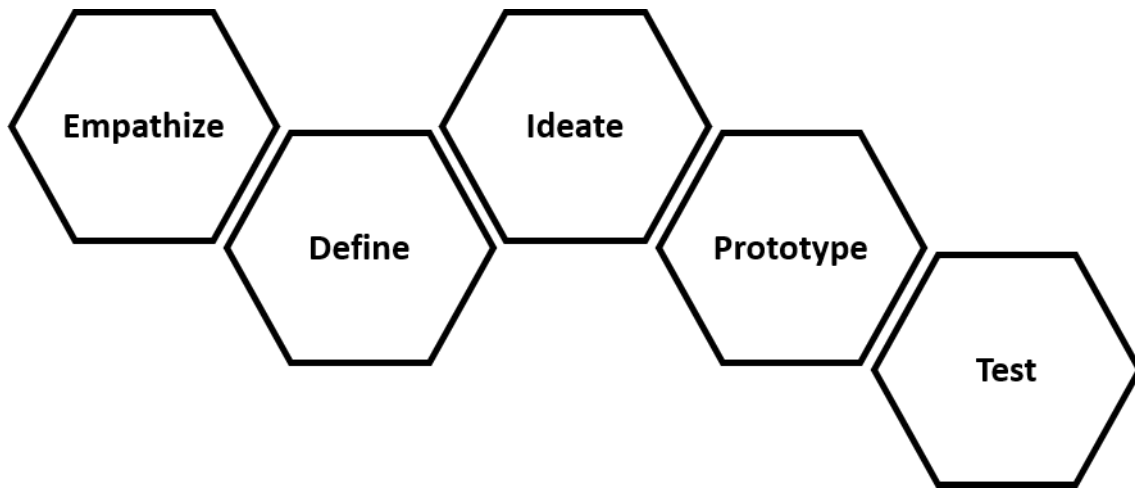


Abbildung 7: Design-Thinking-Ansatz „d.school“ [Dsch2012]

Hineinversetzen (Empathize)

Ziel dieser Phase ist es, die Menschen zu verstehen, auf die das Projektergebnis eine Auswirkung hat (z. B. die Benutzer der Software), um ein Verständnis für ihre Arbeitsweise, ihre physischen oder emotionalen Bedürfnisse oder das Umfeld, in dem sie arbeiten, zu entwickeln.

Aus RE-Perspektive liegt der Fokus in dieser Phase auf der Definition des Systemkontexts und der Identifizierung der Stakeholder (und anderer Anforderungsquellen). Um diese Phase durchzuführen, müssen Sie Ermittlungsaktivitäten mit Ermittlungszielen definieren, die darauf ausgerichtet sind, die Menschen innerhalb Ihres Systemkontexts zu verstehen. Die Anforderungsquellen sind Stakeholder, die nicht Teil des Design-Thinking-Teams sind. Die Ergebnisqualität hängt vom Gesamtprojektziel ab: Dennoch sind alle Informationen willkommen, die ein besseres Verständnis der Stakeholder unterstützen.

Mögliche Techniken für diese Phase sind qualitative Interviews (siehe Abschnitt 3.1.1.1), Anforderungswshops (siehe Abschnitt 3.1.3.1) und Beobachtungstechniken (siehe Abschnitt 3.1.2). Artefaktbasierte Techniken (siehe Abschnitt 3.1.3) sind ebenfalls nützlich, wenn Sie die Domäne oder die Technologie besser verstehen möchten. Zur Dokumentation Ihrer Erkenntnisse stellen Personas eine geeignete Möglichkeit dar (siehe Abschnitt 2.2.4)

Definieren (Define)

Verarbeiten Sie die Erkenntnisse aus Ihrer Empathie-Arbeit und fassen Sie diese zusammen, um eine Benutzerperspektive herauszuarbeiten, die Sie dann in Ihrem Design aufgreifen.

Aus Requirements-Engineering-Perspektive ist diese Phase auf die Dokumentation der erreichten Ergebnisse ausgerichtet. Ein wichtiges Ergebnis ist, dass in Ihrem Design-Thinking-Team eine konkrete Definition des Projektziels erarbeitet wird. Sie können Ermittlungsaktivitäten anhand von Ermittlungszielen definieren, die auf das Verständnis der Beziehungen und Abhängigkeiten ausgerichtet sind. Die Ergebnisqualität muss sehr hoch sein, da die Ergebnisse das gemeinsame Verständnis Ihres Projektteams abbilden müssen. Die Anforderungsquellen für diese Aktivitäten sind die am Design-Thinking-Prozess beteiligten Personen. Nützliche Techniken sind etwa das Mindmapping (siehe Abschnitt

3.3.5) und Storyboards (siehe Abschnitt 3.2.4). Diese helfen Ihnen, die Ergebnisse umzustrukturieren und zu visualisieren.

Ideen finden (Ideate)

Prüfen Sie die verschiedensten möglichen Lösungen, indem Sie eine große Menge an verschiedenartigen Lösungen generieren, die es Ihnen ermöglicht, über das Offensichtliche hinauszugehen und eine größere Bandbreite von Ideen zu erschließen.

Aus Requirements-Engineering-Perspektive handelt es sich bei dieser Phase um eine Kreativitätsphase, in der Sie Ermittlungsaktivitäten mit Ermittlungszielen definieren. Der Schwerpunkt hierbei liegt auf der Erzeugung einer großen Anzahl unterschiedlicher Ideen.

Die Anforderungsquellen für diese Aktivitäten sind die Beteiligten Ihres Design-Thinking-Teams. Es ist beim Design Thinking sehr wichtig, mehrere unterschiedliche Ideen für die Lösung eines vorliegenden Problems zu generieren. Die Ergebnisqualität ist flexibel; ausschlaggebend dafür sind die verfügbaren Ressourcen und zeitliche Einschränkungen. Eine nützliche Technik ist beispielsweise das Brainstorming (siehe Abschnitt 3.2.1).

Prototyp

Setzen Sie Ihre vielversprechendsten Ideen in eine physische Form um, damit Sie sie erfahrbar machen, mit ihnen interagieren können und dadurch hinzulernen und mehr Empathie entwickeln können.

Aus Requirements-Engineering-Perspektive geht es in dieser Phase darum, aus der Entwicklung von Prototypen zu lernen. Sie definieren Ermittlungsaktivitäten mit Ermittlungszielen, bei welchen der Schwerpunkt auf der Ausarbeitung von Ideen aus der Ideenfindungsphase liegt. Die Anforderungsquellen für diese Aktivitäten sind die Beteiligten Ihres Design-Thinking-Teams. Die Ergebnisqualität hängt vom Projekt und besonders von der verfügbaren Zeit und den Ressourcen ab. Stehen für Ihr Projekt nur ein geringes Budget und wenig Zeit zur Verfügung, sollten Sie kostengünstige und einfache Prototypen entwickeln. Im Design Thinking gibt es eine nützliche Faustregel für Prototypen: Verwenden Sie nicht zu viel Aufwand auf einen Prototypen, da Sie sonst vielleicht zu sehr an einer bestimmten Idee hängen und dadurch bei der Bewertung möglicherweise voreingenommen sind. Nützliche Techniken sind alle Arten von Prototyping (siehe 3.2.3), beispielsweise Prototypen, die mit Papier und Stift entwickelt wurden, Mock-ups oder Drahtgittermodelle.

Test

Versuchen Sie es mit hochauflösenden (High-Fidelity) Prototypen (siehe Abschnitt 3.2.3), und nutzen Sie Beobachtungen und Feedback, um sie zu verfeinern, mehr über den Nutzer zu erfahren und Ihren eigenen ursprünglichen Standpunkt zu verfeinern.

Aus Requirements-Engineering-Perspektive ist diese Phase eine Mischung aus Validierung und Ermittlung von Anforderungen. Sie definieren Ermittlungsaktivitäten mit Ermittlungszielen, wobei der Schwerpunkt auf dem Einholen von Feedback von Stakeholdern liegt. Die Anforderungsquellen für diese Aktivitäten sind Stakeholder, die nicht Teil Ihres Design-Thinking-Teams sind. Dieses neutrale Feedback ist wichtig, da die Stakeholder in

Ihrem Team an der Entwicklung von Ideen beteiligt waren und daher voreingenommen sein können.

Die Ergebnisqualität sollte hoch sein, da das Feedback von Stakeholdern sehr wichtig ist, damit Sie die entwickelten Ideen weiter verbessern können. Nützliche Techniken hierfür sind beispielsweise quantitative Interviews (siehe 3.1.1.1) und Workshops (siehe 3.1.3.1), in denen die Prototypen demonstriert werden.

1.4.4.4 Instanziierung

Um einen Design-Thinking-Prozess zu starten, müssen Sie zunächst Ihr Design-Thinking-Team identifizieren und die Phase des Hineinversetzens detailliert planen. Ihr Plan könnte zu Beginn so aussehen:

Nr.	Ermittlungsziel	Ermittlungstechnik	Anforderungsquelle	Ergebnisqualität
Kurzfristige Aktivitäten				
1	Mitglieder für die Design-Thinking-Gruppe ermitteln	Interview	Projektsponsor/Leiter der technischen Abteilung/...	Hohe Sicherheit/hohes Maß an Vollständigkeit
2	Menschen im Systemkontext verstehen: Benutzer (Phase des Hineinversetzens)	Interview	Systembenutzer	Hohe Sicherheit
3	Menschen im Systemkontext verstehen: Benutzer	Beobachtung	Systembenutzer	Hohe Sicherheit
4	Menschen im Systemkontext verstehen: Vorgesetzte der Benutzer	Workshop	Vorgesetzte der Benutzer	Hohe Sicherheit
Langfristige Aktivitäten				
5	Definitionsphase	Workshops	Design-Thinking-Team	noch zu klären*
6	Ideenfindungsphase	Kreativitätstechniken	Design-Thinking-Team	noch zu klären*
7	Phase der Prototypenentwicklung	Prototyping	Design-Thinking-Team	noch zu klären*

8	Testphase	noch zu klären*	noch zu klären*	noch zu klären*
---	-----------	-----------------	-----------------	-----------------

*Die Einträge „noch zu klären“ in dieser Tabelle bedeuten, dass Sie diese Aspekte zu Beginn des Design-Thinking-Prozesses nicht entscheiden können oder sollten, da sie in hohem Maß von den Ergebnissen der anderen Phasen abhängen. Denken Sie auch an den iterativen Charakter des Design-Thinking-Prozesses und daran, dass die langfristigen Aktivitäten zunächst abstrakt definiert und erst im Laufe des Prozesses detaillierter formuliert werden.

1.4.4.5 Weiterführende Literatur

[Brown2009] bietet einen allgemeinen Überblick über die dem Design Thinking zugrunde liegenden Ideen und dessen Potenzial.

[Design Council 2007] erläutert das Double-Diamond-Modell, das 2005 vom British Design Council veröffentlicht wurde.

[KnZK2016] stellen eine kompakte Version des Design Thinking für einen Zeitraum von fünf Tagen dar. Das Buch orientiert sich an der Struktur der fünf Tage und bietet konkrete methodologische Orientierungshilfen dazu, was an den einzelnen Tagen zu tun ist.

[LeLL2018] stellt eine nützliche Sammlung von Hinweisen, Beispielen und Techniken für Design-Thinking-Experten bereit.

2 Anforderungsquellen

In diesem Kapitel werden die drei Arten von Anforderungsquellen behandelt sowie deren Identifizierung, Klassifizierung und Management. In Abschnitt 2.1 wird der pragmatische und systematische Ansatz für die Identifizierung von Anforderungsquellen erläutert. Abschnitt 2.2 deckt Identifizierung, Klassifizierung und Management von Stakeholdern ab und die Abschnitte 2.3 und 2.4 dasselbe für Dokumente und Systeme.



Tools, die auf Künstlicher Intelligenz (KI) basieren, können die systematische und pragmatische Identifikation von Anforderungsquellen wirksam unterstützen. Insbesondere Technologien wie generative KI und Large Language Models (LLM) erlauben es, große Mengen an unstrukturierten Daten effizient auszuwerten. Solche Daten umfassen beispielsweise interne E-Mails, Inhalte aus dem Intranet, umfangreiche elektronische Dokumentensammlungen oder auch Beiträge aus sozialen Medien. Durch diese automatisierte Analyse können relevante Stakeholder, Dokumente und Systeme schneller und präziser identifiziert werden. Die Verwendung solcher KI-gestützter Verfahren ergänzt bewährte Methoden der Quellenarbeit sinnvoll, ermöglicht eine fundierte Validierung der ermittelten Anforderungsquellen und hilft dem Requirements Engineer, die vorgesehenen Dokumentationsschemata transparent und zielgerichtet anzuwenden.

2.1 Grundlagen zu Anforderungsquellen

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 und einem höheren Risiko für Ihr Vorhaben. Wurden beispielsweise Benutzer nicht einbezogen, steigt das Risiko einer geringen Akzeptanz des neuen Systems. Wird es versäumt, einen wichtigen Stakeholder zu involvieren, kann das Projekt in einer kritischen Phase unter Umständen blockiert werden. Wenn zudem vorhandene Funktionalitäten eines Altsystems unberücksichtigt bleiben, kann dies zur Folge haben, dass grundlegende Anforderungen übersehen werden. Aus diesem Grund ist es die Aufgabe des Requirements Engineers während eines Entwicklungsprojekts alle relevanten Anforderungsquellen zu identifizieren und zurate zu ziehen.

Die drei wichtigsten Arten von Anforderungsquellen sind [IREB2020]:

- Stakeholder
- Dokumente
- Systeme

Da es keine Anforderung ohne Quelle gibt, ist eine der ersten Aktivitäten bei der Anforderungsermittlung naturgemäß das Identifizieren potenzieller Anforderungsquellen. Dies ist ein iterativer Prozess: Es reicht nicht aus, diese Quellen lediglich zu Beginn eines Projekts oder einer Produktentwicklung zu ermitteln. Es handelt sich vielmehr um einen wiederkehrenden Prozess, der kontinuierlich wiederholt werden muss. Mit jeder neuen Ermittlungsaktivität und dem zunehmenden Wissen über das herzustellende Produkt können

neue potenzielle Anforderungsquellen identifiziert werden, die in einem rekursiven Prozess zu weiteren Ermittlungsaktivitäten und zur Identifizierung weiterer Quellen führen können.

Beispiel

Sie führen ein Interview mit einem potenziellen Benutzer (einer Zeitreisenden) der Zeitmaschine durch, die Sie gerade entwickeln. Bei diesem Interview erwähnt die Zeitreisende, dass ihr Kollege in der Regel die Aktion durchführt, nach der Sie gerade gefragt haben, und dass sie die Details dazu in der Reiseprozessdokumentation nachschlagen müsste.

Die Zeitreisende hat gerade zwei potenzielle Anforderungsquellen genannt, die Sie berücksichtigen sollten.

Daher erkundigen Sie sich nach dem Namen des Kollegen und nach den Details zur Reiseprozessdokumentation, die sie erwähnt hat.

Nach dem Interview prüfen Sie in der Dokumentation Ihrer Anforderungsquellen (siehe 2.3.2), ob der Kollege der Interviewpartnerin und die Zeitreise-Prozessdokumentation bereits identifiziert wurden oder ob es sich dabei um neue potenzielle Anforderungsquellen handelt.

Durch das Interagieren mit einer zuvor identifizierten Anforderungsquelle haben Sie eine weitere Quelle identifiziert. So entsteht ein rekursiver Prozess.

Seien Sie selbst in späten Projektphasen auf die Identifizierung neuer Anforderungsquellen vorbereitet, wenn bei den Projektbeteiligten gewöhnlich bereits ein Gefühl von „wir wissen, wer sie sind und was sie möchten“ vorherrscht.

Sie sollten stets wieder auf die Dokumentation Ihrer Anforderungsquellen zurückgreifen und neu bewerten, ob sie immer noch aktuell ist, oder ob etwa neue Anforderungsquellen relevant sind oder sich einige zuvor identifizierte Quellen als hinfällig erwiesen haben.

Die pragmatische und die systematische Identifizierung von Anforderungsquellen

Wir unterscheiden zwei grundlegende Herangehensweisen zur Identifizierung von Anforderungsquellen:

- Pragmatische Identifizierung
- Systematische Identifizierung

Pragmatische Identifizierung von Anforderungsquellen

Die „pragmatische Identifizierung“ ist ein Modewort und bedeutet, die eigene Intuition und Erfahrung zu nutzen. Wenn Sie bereits an einem Projekt in derselben Abteilung beteiligt waren oder in einem ähnlichen Projekt in einem anderen geschäftlichen Kontext, dann werden Sie ohne langes Nachdenken eine Liste mit potenziellen Stakeholdern, Dokumenten und Systemen nennen können.

Das ist ein wichtiges Mittel der Identifizierung potenzieller Anforderungsquellen; mit geringem Zeitaufwand erhalten Sie eine Liste von potenziellen Anforderungsquellen, mit der Sie beginnen können. Möglicherweise fallen Ihnen später im Produktentwicklungsprozess noch weitere potenzielle Anforderungsquellen ein.

Es ist jedoch gefährlich, sich auf die pragmatische Identifizierung von Anforderungsquellen als einzigen Ansatz zu verlassen, da Ihnen so entscheidende Quellen entgehen könnten. Sie

sollten sich immer absichern und Ihre Ergebnisse aus der pragmatischen Identifizierung durch die Nutzung der systematischen Identifizierung erweitern.

Systematische Identifizierung von Anforderungsquellen

Die systematische Identifizierung von Anforderungsquellen besteht aus zwei Schritten:

1. Ermittlung von Kriterien, welche die Anforderungsquellen charakterisieren können und dadurch zu deren Identifizierung beitragen
2. Systematische Suche auf Basis der Kriterien (Anwendung des Schneeballprinzips³)

Für die systematische Identifizierung müssen Sie Ermittlungsaktivitäten festlegen (siehe 1.3) mit Ermittlungszielen, deren Fokus auf der Identifizierung von Anforderungsquellen liegt. Auf diese Weise unterteilen Sie das abstrakte Ziel, alle relevanten Anforderungsquellen zu finden, in spezifische, umsetzbare Aufgaben.

In Abbildung 8 bis Abbildung 10 sind Beispiele für Ermittlungsaktivitäten zur Identifizierung von Stakeholdern für den Bau einer Zeitmaschine enthalten.

ID	RS_EA_01
Ermittlungsziel	Mindestens zehn potenzielle Zeitreisende (Benutzer) finden
Ergebnisqualität	Benutzername, Rolle und Kontaktdaten
Anforderungsquelle(n)	Organigramm
Ermittlungstechnik	Perspektivenbasiertes Lesen

Abbildung 8: Beispiel für eine Ermittlungsaktivität zur Identifizierung von Benutzern als Stakeholdern

ID	RS_EA_02
Ermittlungsziel	Mindestens fünf rechtsverbindliche Dokumente finden, die für eine Zeitmaschine potenziell relevant sind
Ergebnisqualität	Bezeichnung des Dokuments, aktuelle Version, Speicher-/Aufbewahrungsort
Anforderungsquelle(n)	Evelyn Hall, Bob Miller und der Rechtsberater/Anwalt des Entwicklungsunternehmens

³ Wenn Sie einen Stakeholder identifiziert haben, fragen Sie ihn nach weiteren Anforderungsquellen.

Ermittlungstechnik	Interview
--------------------	-----------

Abbildung 9: Beispiel für eine Ermittlungsaktivität zur Identifizierung von rechtsverbindlichen Dokumenten

ID	RS_EA_03
Ermittlungsziel	Herausfinden, wer der Rechtsberater/Anwalt des Entwicklungsunternehmens ist
Ergebnisqualität	Name, Kontaktdaten, Verfügbarkeit
Anforderungsquelle(n)	Rechtsabteilung
Ermittlungstechnik	Telefoninterview

Abbildung 10: Beispiel für eine Ermittlungsaktivität zur Identifizierung eines Rechtsberaters/Anwalts

In diesem Beispiel wurde dem Requirements Engineer bei der Planung der Anforderungsquellen (Requirements Sources – RS) für die Ermittlungsaktivität (Elicitation Activity – EA) RS_EA_02 bewusst, dass er noch nicht identifiziert hatte, wer der Rechtsberater des Unternehmens für das Entwicklungsprojekt war. Daher fügte er eine weitere Ermittlungsaktivität hinzu (RS_EA_03), um diese Frage aufzugreifen.

Sie sollten immer sowohl die pragmatische als auch die systematische Identifizierung nutzen. Die pragmatische Identifizierung spart Zeit und Ressourcen, birgt jedoch das Risiko der subjektiven Voreingenommenheit. Die pragmatische Identifizierung allein wird in den meisten Fällen nicht ausreichen. Die systematische Identifizierung erfordert dagegen zwar mehr Zeit und Ressourcen als die pragmatische, sie führt jedoch auch zu einer objektiveren und potenziell vollständigeren Liste von Anforderungsquellen.

Wir schlagen vor, dass Sie mit der pragmatischen Identifizierung beginnen und Ihre Ergebnisse dann zur Sicherheit mithilfe der systematischen Identifizierung erweitern.

2.2 Stakeholder identifizieren, klassifizieren und verwalten

In diesem Abschnitt wird die wichtigste Anforderungsquelle für die meisten Projekte behandelt: die Stakeholder. Stakeholder sind Menschen oder juristische Personen im Zusammenhang mit Ihrem Projekt, die das Projekt beeinflussen und die davon betroffen sind. Als Requirements Engineer können Sie sich mit diesen treffen, sie beobachten, mit ihnen sprechen und ihnen Fragen stellen. Juristische Personen werden durch natürliche Personen repräsentiert (z. B. wird das Unternehmen WeAreTheBest Ltd durch seine Geschäftsführung oder Vertreter der Öffentlichkeitsarbeit repräsentiert).

Stakeholder können in Gruppen oder Rollen zusammengefasst werden, beispielsweise Benutzer, Instandhaltungsmitarbeiter, Tester usw. Dieses Konzept der Stakeholder-Gruppen oder -rollen ist für zahlreiche Requirements-Engineering-Aufgaben sehr nützlich. Denken Sie jedoch daran, dass Sie nicht mit abstrakten Rollen interagieren können, sondern schließlich immer einen echten Menschen benötigen, mit dem Sie kommunizieren können.

Außerdem geraten Sie mit abstrakten Gruppen in aller Regel nicht in Schwierigkeiten, was mit Einzelpersonen sehr wohl möglich ist (z. B. mit Herrn Fudge, Leiter für Reisegenehmigungen, der nicht eingeladen wurde, um seinen Input zu den Anforderungen für das Zeitreisensystem abzugeben).

Hinweis 2.2.1:

Es hat sich in der Praxis bewährt, für jede Stakeholder-Gruppe einzelne Stakeholder zu benennen, vor allem für wichtige interne Gruppen wie die Rechtsabteilung, Security Officers, Manager, Vertreter von verbundenen Abteilungen usw.

In Abschnitt 2.2.1 wird erläutert, wie Stakeholder systematisch und pragmatisch identifiziert werden können; in Abschnitt 2.2.2 wird das Stakeholder-Relationship-Management (einschließlich Klassifizierung) behandelt;

in Abschnitt 2.2.3 erfahren Sie mehr über die Stakeholder-Dokumentation und

in Abschnitt 2.2.4 liegt das Hauptaugenmerk auf dem Benutzer als einer speziellen Stakeholder-Rolle.

2.2.1 Stakeholder als Anforderungsquelle identifizieren und auswählen

Stakeholder spielen in den meisten Entwicklungsprojekten eine entscheidende Rolle. Daher ist deren Identifizierung und Verwaltung eine wichtige Aufgabe im Rahmen der Anforderungsermittlung.

2.2.1.1 Pragmatischer Ansatz zur Identifizierung von Stakeholdern

Es ist üblich, Stakeholder pragmatisch zu identifizieren. Aufgrund seiner Erfahrung im Projektkontext (z. B. aus früheren Projekten in derselben Organisation) und durch Wiederverwendung von vorhandenen Stakeholder-Listen kann der Requirements Engineer schnell eine initiale Liste von Stakeholder-Gruppen und -Rollen zusammenstellen (darin sind häufig bereits einzelne Vertreter enthalten).

2.2.1.2 Systematischer Ansatz zur Identifizierung von Stakeholder-Gruppen und -Rollen

Wenngleich die pragmatische Identifizierung sehr hilfreich ist, um mit der Identifizierung von Stakeholdern zu beginnen, sollte diese immer durch die systematische Identifizierung untermauert werden.

Für die systematische Identifizierung von Stakeholdern sind folgende Quellen und Techniken hilfreich:

- **Checklisten mit typischen Stakeholder-Gruppen und -Rollen**

Listen, wie die metaphorische Länderkarte für Stakeholder aus [Rupp et al.2014] – siehe unten – oder die folgende einfache Liste können für alle Projekte als Orientierungshilfe dienen:

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

Sehen Sie sich jede Stakeholder-Rolle in der Liste an und klären Sie, ob sie für Ihr Projekt eine relevante Rolle ist. Zudem ist es sinnvoll, im Laufe der Zeit eine eigene Checkliste zu erstellen, vor allem wenn Sie auf einem bestimmten Gebiet tätig sind (z. B. Versicherungsbranche), auf dem die Stakeholder-Rollen für die einzelnen Projekte sehr ähnlich sein können.

- **Organisationsstrukturen**

(z. B.: Organigramme des Unternehmens, welches das zu bauende System dann nutzen wird)

Die meisten Unternehmen verfügen über Organigramme. Diese können nützlich sein, um die unterschiedlichen Abteilungen in einem Unternehmen und die Stakeholder dieser Abteilungen herauszufinden.

- **Geschäftsprozessdokumentation**

Viele Unternehmen haben ihre Geschäftsprozesse auf irgendeine Weise dokumentiert, auch wenn diese Dokumentation vielleicht nicht immer ganz aktuell ist. Dies könnte ein Geschäftsprozessmodell (z. B. ein BPMN-Modell) oder eine Beschreibung der Funktionsweise eines Prozesses in natürlicher Sprache sein. Mithilfe einer solchen Dokumentation könnten Sie herausfinden, welche Rolle für eine bestimmte Aufgabe verantwortlich ist. Eine Dokumentation in dieser Form ist bereits eine potenzielle Anforderungsquelle (siehe 2.3).

- **Stakeholder-Kategorisierungsschemata**

Vorhandene Stakeholder-Kategorisierungsschemata sind ebenfalls wertvolle Quellen für die Identifizierung von potenziell relevanten Stakeholder-Rollen. Sie bezeichnen typische Kategorien und Rollen von Stakeholdern.

Einige Beispiele hierfür: das Zwiebschalenmodell nach Alexander [Alexander2005] – Näheres dazu im nachfolgenden Abschnitt – oder der generische Stakeholder-Plan der Robertsons [RoRo2013].

- **Informationsfokus**

Versuchen Sie, relevante Informationen oder Wissensgebiete zu identifizieren und Stakeholder-Gruppen abzuleiten, die diese Informationen bereitstellen können oder die Experten auf diesen Gebieten sind.

Beispiel:

ID	RS_EA_04
Ermittlungsziel	Finden Sie mindestens zwei Stakeholder, die Ihnen potenziell Informationen über die physikalischen Randbedingungen von Zeitreisen bereitstellen können.
Ergebnisqualität	Name, Rolle und Kontaktdaten des Stakeholders
Anforderungsquelle(n)	Organigramm
Ermittlungstechnik	Perspektivenbasiertes Lesen

Abbildung 11: Beispiel für eine informationsbasierte Ermittlungsaktivität

- **Produktlebenszyklus-Analyse**

Unternehmen Sie einen virtuellen Spaziergang durch den Lebenszyklus des zu entwickelnden Produkts. Wer wird in irgendeiner Weise mit dem Produkt oder dessen Dokumentation, angefangen mit den Anforderungen über die Entwicklung und Verwendung bis hin zu Deinstallation oder Rückbau, interagieren? Dabei werden Sie auf andere Abteilungen (z. B. Produktion und Instandhaltung) oder Organisationen (z. B. Secondhandbenutzer, Lieferanten oder Recyclingunternehmen) als potenzielle Stakeholder stoßen.

Hinweis 2.2.2:

Es hat sich als nützlich erwiesen, für die Identifizierung von Stakeholdern einen Workshop abzuhalten, bei dem alle potenziellen Rollen, Namen und Wissensgebiete von Stakeholdern gesammelt werden, die den Teilnehmern in den Sinn kommen. In einem nächsten Schritt werden die Dreiergruppe aus Namen, Stakeholder-Rollen und Wissensgebieten sowie potenzielle Lücken identifiziert.

2.2.1.3 Systematischer Ansatz zur Identifizierung von individuellen Stakeholdern (Personen)

Wie zuvor angemerkt, ist es wichtig, Individuen zu identifizieren, die erreichbar und greifbar sind. Diese Stakeholder haben Namen, Kontaktdaten und Geburtstage. Letztere sind für Requirements-Engineering-Aktivitäten vielleicht nicht von primärer Bedeutung, aber im Hinblick auf das Stakeholder-Relationship-Management eine wertvolle Information.

Während der systematischen Identifizierung von einzelnen Stakeholdern legt der Requirements Engineer entsprechende Ermittlungsaktivitäten fest.

Beispiel:

ID	RS_EA_05
Ermittlungsziel	Mindestens eine Person für die Stakeholder-Rolle „Training für Zeitreisende“ suchen
Ergebnisqualität	Name und Kontaktdaten des Stakeholders
Anforderungsquelle(n)	Dr. Emmet Brown (erfahrener Zeitreisender, könnte geeignete Kandidaten kennen)
Ermittlungstechnik	Interview

Abbildung 12: Beispiel für eine Ermittlungsaktivität zur Identifizierung von einzelnen Stakeholdern

2.2.2 Stakeholder-Relationship-Management

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. Um Stakeholder in den Ermittlungsprozess einzubinden, müssen wir sicherstellen, dass sie wissen, worum es bei dem Projekt geht und welche Rolle sie in dem Projekt einnehmen.

Der Stakeholder-Kreis [Bourne2015] ist ein nützliches Framework für ein erfolgreiches Stakeholder-Relationship-Management. Er besteht aus den folgenden 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

Das Framework von Bourne zielt auf das Projekt–Stakeholder–Management ab. Wir haben die Details der einzelnen Schritte leicht angepasst, damit sie in den Kontext des Anforderungs–Stakeholder–Managements und der IREB–Lehrpläne passen.

Schritt 1: Identifizierung aller Stakeholder

Der Stakeholder–Kreis beginnt mit der Identifizierung sämtlicher Stakeholder. Dieser Schritt ist in Abschnitt 2.2.1 beschrieben.

Schritt 2: Priorisierung, um festzulegen, wer wichtig ist

Nicht alle Stakeholder sind gleich. Einige sind wichtiger als andere, und einige sind zu Beginn eines Projekts wichtiger als am Ende.

Da uns für das Stakeholder–Management und die Anforderungsermittlung begrenzte Ressourcen zur Verfügung stehen, ist es entscheidend, die identifizierten Stakeholder zu priorisieren. In Abhängigkeit vom Projekt sind gegebenenfalls unterschiedliche Priorisierungsschemata sinnvoll. Bevor Sie ein Priorisierungsschema auswählen, sollten Sie sich folgende Fragen stellen:

- Welche Vorzüge hat dieses Schema?
- Inwiefern sind die Informationen, die es liefern kann, nützlich für das Projekt?
- Welchen Vorteil hat dieses Schema gegenüber anderen Schemata, die wir verwenden könnten?
- Wie unterstützt es uns dabei herauszufinden, wer für unser Projekt wirklich wichtig ist?

Der erste Schritt vor der Priorisierung ist die Klassifizierung. Werden alle Stakeholder nach einem bestimmten Schema klassifiziert, dann können diese Klassen in Bezug zueinander priorisiert werden. Häufig ist es auch sinnvoll, innerhalb einer identifizierten Klasse weitere Prioritäten zu setzen.

Ein Beispiel für ein Klassifizierungsschema ist das Zwiebelschalenmodell von Ian Alexander [Alexander2005]. Es weist drei Klassen auf:

- **Stakeholder des Systems:** Diese Stakeholder sind direkt von dem neuen oder geänderten System betroffen. Typische Beispiele für diese Klasse sind Benutzer, Wartungsmitarbeiter und Systemadministratoren.
- **Stakeholder des Umgebungssystems:** Diese Stakeholder sind indirekt von dem neuen oder geänderten System betroffen. Typische Beispiele dafür sind Manager von Benutzern, Project Owner oder Sponsoren.
- **Stakeholder aus dem weiteren Umfeld:** Diese Stakeholder haben eine indirekte Beziehung zu dem neuen oder geänderten System. Typische Beispiele sind Gesetzgeber, Standardisierungsinstitute, Nichtregierungsorganisationen (NGOs, z. B. Gewerkschaften, Umweltschutzverbände), Wettbewerber und auch Projektmitglieder, die zwar an der Entwicklung des Systems beteiligt sind, das System aber nicht produktiv nutzen.

Stakeholder können auch nach ihrem Einfluss auf das Projekt klassifiziert werden (hoher vs. geringer Einfluss) und nach ihrer Motivation in Bezug auf das Projekt (hohe vs. geringe

Motivation). Stakeholder mit hohem Einfluss können ein Projekt beispielsweise behindern oder voranbringen. Stakeholder mit höherer Motivation sind wertvoll für das Projekt, da sie selbst ein Interesse daran haben, das Projekt voranzubringen [Rupp et al.2014].

Weitere Beispiele für Klassifizierungsattribute von Stakeholdern:

- Nähe
- Verfügbarkeit
- Interesse
- Einfluss
- Erfahrung in ähnlichen Projekten
- Kommunikationsfähigkeiten

Schritt 3: Visualisierung, um die Gesamtheit der Stakeholder zu verstehen

Alle, über die Stakeholder gesammelten Informationen müssen dokumentiert (d. h. visualisiert) und diese Dokumentation auf dem aktuellen Stand gehalten werden. Eine solche visuelle Darstellung der Stakeholder fördert das Verständnis von den Stakeholdern insgesamt, und sie trägt dazu bei, dass keine wichtigen Stakeholder übersehen werden.

Näheres zur Visualisierung/Dokumentation von Stakeholdern finden Sie in Abschnitt 2.2.3.

Schritt 4: Aktives Einbringen durch effektive Kommunikation

Nun ist es wichtig, ein Verständnis davon zu entwickeln, was die einzelnen Stakeholder hinsichtlich Kommunikation erwarten, und wie deren Haltung gegenüber dem Projekt ist. Die folgenden Elemente sollten in Betracht gezogen werden [Bourne2015]:

- Kultur (der Organisation, des Teams oder von Individuen)
- Identifizierung mit der (Entwicklungs-)Aktivität und deren Ergebnissen
- Wahrgenommene Bedeutung der Aktivität und deren Ergebnisse
- Persönliche Attribute, wie Persönlichkeit und Rolle

Nach der Beurteilung der Haltungen und Einstellungen der Stakeholder müssen Sie eine „realistische angestrebte Haltung“ für jeden Stakeholder definieren. Diese angestrebte Haltung sollte sowohl dem Stakeholder als auch dem Entwicklungsprojekt dienen.

Gibt es eine Lücke zwischen der aktuellen und der angestrebten Haltung? Wenn ja, sollten Sie Aktivitäten definieren, die auf das Schließen dieser Lücke abzielen (Kommunikationsplan).

Die folgenden Fragen können für den Schritt „aktives Einbringen“ nützlich sein:

- Wie oft sollte ich bestimmte Stakeholder kontaktieren, um sie über die Ermittlung der Anforderungen auf dem aktuellen Stand zu halten?
- Welche Informationen sind für welchen Stakeholder relevant?
- Auf welche Weise halte ich einen bestimmten Stakeholder am besten auf dem Laufenden? (z. B. per Telefonanruf, E-Mail, Newsletter, Mittagessen)
- Auf welche Weise setze ich mich am besten mit diesem Stakeholder in Verbindung, wenn ich Informationen benötige?

- Wie behalte ich den Überblick darüber, wann ich diesen Stakeholder zuletzt kontaktiert habe?
- Gibt es irgendwelche kulturellen Aspekte, die für die Kommunikation relevant sind?

Hinweis 2.2.3:

Beim Management von Stakeholdern für Anforderungen und Stakeholdern für Projekte kann es leicht zu Überschneidungen kommen. Stellen Sie sicher, dass Sie sich mit der verantwortlichen Person für das Projekt-Stakeholder-Management abstimmen, wie Sie mit einer solchen Überschneidung umgehen.

Das Stakeholder-Relationship-Management für Anforderungen ist auf das Management der Beziehung zu einem Stakeholder ausgerichtet, um die notwendigen Anforderungen von ihm zu erfahren. Beim Stakeholder-Relationship-Management für Projekte besteht in der Regel eine breitere Perspektive auf das Stakeholder-Relationship-Management.

Schritt 5: Prüfung der Wirksamkeit der Kommunikation

Wiederholen Sie die Beurteilung von Schritt 4 regelmäßig, um zu festzustellen, wo weitere Stakeholder-Relationship-Managementaktivitäten erforderlich sind und, um zu sehen, ob die unternommenen Aktivitäten effektiv waren. Passen Sie den Kommunikationsplan nach Bedarf an, um die Lücke, die zwischen aktuellen und angestrebten Haltungen von Stakeholdern stets geschlossen zu halten.

2.2.3 Dokumentationsschema für einbezogene Stakeholder

Alle Informationen, die während der Identifizierung von Stakeholdern und des Stakeholder-Relationship-Managements gesammelt wurden, müssen ausreichend dokumentiert werden. Eine solche Dokumentation sollte für jeden Stakeholder mindestens Folgendes beinhalten:

- Name
- Funktion (Rolle)
- Kontaktdaten
- Verfügbarkeit (räumlich und zeitlich)
- Relevanz
- Fachgebiet und Umfang des Fachwissens
- Ziele und Interessen bezogen auf das Projekt

Darüber hinaus sollten die Klassifizierungsinformationen (z. B. nach Ian Alexander) und die Prioritäten aus Schritt 2 sowie die in Schritt 4 gewonnenen Informationen (einschließlich des Kommunikationsplans) dokumentiert werden.

Abhängig vom Projekt können auch weitere Informationen relevant sein. Folgende Einflussfaktoren sind denkbar:

- *Öffentliche Relevanz:* In einem Kontext von größerer 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 striktem Zeitrahmen können die Erreichbarkeit oder Antwortzeit eines Stakeholders sehr wichtige Informationen sein, wenn kritische Entscheidungen nötig sind.

Stellen Sie sicher, dass Sie eine Form der Dokumentation wählen, welche die Bedürfnisse an die Dokumentation und das Stakeholder-Relationship-Management Ihres Projektumfeldes erfüllt.

Übliche Dokumentationsformen sind:

- Stakeholder-Tabelle
- Stakeholder-Datenbank (häufig in das Requirements-Management-Werkzeug integriert)
- Stakeholder-Mindmap

Zudem können Diagramme oder andere grafische Darstellungen verwendet werden, um Änderungen in der Haltung oder bei Prioritäten von Stakeholdern zu verfolgen.

Die Dokumentation zu Stakeholdern muss so lange auf dem aktuellen Stand gehalten werden, wie diese Informationen benötigt werden könnten (d. h. gewöhnlich mindestens bis zum Ende des Entwicklungsprojekts oder sogar bis zum Ende des Produktlebenszyklus).

2.2.4 Der Nutzer als besondere Stakeholder-Gruppe

Bei interaktiven Systemen mit einer Benutzeroberfläche sind alle direkten Anwender des Systems für den Requirements Engineer von besonderem Interesse.

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

Die Anzahl potenzieller Anwender erlaubt es gewöhnlich 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 [Cooper2004]. Personas sind fiktive Individuen, die typische Nutzergruppen eines Systems repräsentieren, welche ähnliche Bedürfnisse, Ziele, Verhaltensweisen oder Einstellungen haben. 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, oder Proto-Personas [Gothelf2013] entwickelt werden.

Personas können auch auf Basis von Rohdaten erstellt werden, die durch Interviews, Contextual Inquiries, Fragebögen oder Apprenticing (siehe Beschreibung dieser Methoden in Abschnitt 2.2.2) ermittelt wurden, wie von [Goodwin2009] beschrieben. Das zentrale Konzept der Erstellung von Personas ist die Identifizierung von bipolaren Variablen, die unterschiedliche Personas kennzeichnen und differenzieren.

Die resultierenden Nutzergruppen oder Personas sollten als primär oder sekundär priorisiert werden. Das System, besonders seine Benutzungsoberfläche, wird für die primäre Nutzergruppe optimiert. Sekundäre Benutzergruppen werden nur insoweit unterstützt, wie die Erfüllung deren Bedürfnisse die User-Experience der primären Benutzergruppen nicht beeinträchtigt.

Wenn Ihre primären Stakeholder Benutzer sind, dann sollten Sie das Muster des Human-Centered Design anwenden, um deren Erwartungen zu erfüllen (siehe Abschnitt 1.4.3).

Hinweis 2.2.4:

Bei der agilen Herangehensweise an die Softwareentwicklung liegt das Hauptaugenmerk auf der Werthaltigkeit, welche die Lösung den Benutzern bieten soll („User Story“). Daher wird der Benutzer eines interaktiven Systems als primärer Stakeholder betrachtet. Dennoch besteht, vor allem in großen Organisationen, das Risiko, dass agile Teams oder Product Owner nicht mit den echten Nutzern in Kontakt sind. Die Personas werden hier möglicherweise von jemandem erdacht, der nur vorgibt, die Benutzer zu kennen! In einem solchen Kontext sind die Stakeholder der Organisation (Manager, Fachbereiche, Rechtsabteilungen, Marketing usw.) wesentlich präsenter und dominanter als der externe Endbenutzer. Seien Sie sich über diese Falle im Klaren: Bestehen Sie darauf, dass ein direkter Zugang zu Endbenutzern besteht, um eine angemessene Nutzerforschung durchführen und nach der Sprint-Lieferung direktes Feedback einholen zu können. Nur so können Sie Ihre Benutzer und deren Bedürfnisse wirklich kennenlernen.

2.3 Dokumente identifizieren, klassifizieren und verwalten

Dokumente werden verwendet, um Informationen zwischen Menschen über Zeit und Raum hinweg zu übermitteln. Aus Dokumenten lassen sich häufig Anforderungen ableiten, was sie zu einer wertvollen Quelle macht.

In Abschnitt 2.3.1 behandeln wir die Identifizierung und Auswahl solcher Dokumente und in Abschnitt 2.3.2 besprechen wir die Dokumentation von Dokumenten als Anforderungsquelle.

2.3.1 Dokumente als Anforderungsquelle identifizieren und auswählen

Anforderungen können aus vielen verschiedenen Arten von Dokumenten abgeleitet werden. Je nach Charakter eines Entwicklungsprojekts, können Dokumente entweder eine hohe, mittlere oder niedrige Bedeutung als Anforderungsquelle einnehmen. Engineering-Projekte für technische Systeme (z. B. Doppelkupplungsgetriebe) haben für gewöhnlich zahlreiche Dokumente als Anforderungsquellen (z. B. technische Normen, Patente), während es bei Human-Centered-Projekten (z. B. eine App für Mobile-Shopping) weniger Dokumente gibt.

So können beispielsweise folgende Dokumentenarten als Anforderungsquellen dienen (unvollständige Aufzählung):

- **Technische Standards, Gesetze, interne Vorschriften**

Alle Entwicklungsprojekte unterliegen gesetzlichen Randbedingungen, sowohl zivil- als auch strafgesetzlichen Randbedingungen (national und potenziell auch

international). Datenschutzgesetze sind typischerweise für viele Entwicklungsprojekte relevant. Für die Medizin- und Nahrungsmittelbranche in den USA sind etwa die Bestimmungen der US-amerikanischen Food and Drug Administration (FDA) von großer Bedeutung, und in der Automobilindustrie gelten branchenspezifische internationale Normen wie die ISO 26262, in der funktionale Sicherheitsaspekte geregelt sind. Dies sind nur einige Beispiele. So gibt es für jede Branche spezielle Normen und Gesetze. Zudem hat jedes Unternehmen potenziell interne Vorschriften, die unter Umständen Anforderungen für das Entwicklungsprojekt enthalten (z. B. Styleguides).

- **Anforderungsdokumente**

Nicht alle Systeme werden auf der grünen Wiese entwickelt. Häufig sind Vorgängersysteme zu ersetzen oder es wird eine Variante eines bereits vorhandenen Systems entwickelt (z. B. ein neues Vertragsmanagementsystem für ein Versicherungsunternehmen, das kürzlich mit einem Wettbewerber gemergt wurde). In solchen Fällen können Anforderungsdokumente aus Vorgängersystemen wertvolle Quellen für neue Anforderungen darstellen, z. B. Geschäftsregeln, Glossare, Geschäftsobjektmodelle, Use Cases, Workflowspezifikationen, Authentifizierungs- und Berechtigungsdokumente usw.)

Auch Anforderungsdokumente von angrenzenden Systemen können relevant sein.

- **Benutzerhandbücher**

Wenn Vorgänger- oder Wettbewerbersysteme verfügbar sind, gibt es wahrscheinlich auch bereits Benutzerhandbücher. Diese bieten einen guten Überblick über die Systemfunktionalität, und sie können aktueller⁴ sein als die Anforderungsdokumentation dieser Systeme.

- **Strategiepapiere**

Wenn das zu entwickelnde System Teil einer übergreifenden Unternehmensstrategie ist, sind gegebenenfalls Strategiepapiere oder Präsentationen vorhanden, die relevante Anforderungen für das System enthalten können.

- **Zieldokumentation**

Bevor mit einer Systementwicklung begonnen wird, sind gewöhnlich verschiedene Dokumente vorhanden, in denen die Notwendigkeit und die potenziellen Vorzüge des Projekts bewertet werden. Solche Dokumentationen beinhalten in der Regel die Ziele des Projekts, die eine wesentliche Anforderungsquelle darstellen.

- **Geschäftsprozessdokumentation**

Wenn das zu entwickelnde System im Zusammenhang mit einem oder mehreren Schritten eines Geschäftsprozesses steht, stellt jegliche Geschäftsprozessdokumentation eine wertvolle Quelle für Anforderungen dar. Eine solche Dokumentation ist in unterschiedlicher Form anzutreffen: z. B. Dokumente in

⁴ Bei einem guten Anforderungsmanagement sollte dieses Szenario nicht auftreten. Leider haben noch nicht alle Unternehmen und Projekte einen solchen Reifegrad in Bezug auf das Anforderungsmanagement erreicht.

reiner Textform, Präsentationsfolien, Intranetseiten, Wiki-Seiten, BPMN oder andere Modelle.

- **Schnittstellendokumentation**

Die meisten Systeme interagieren mit anderen Systemen. Jegliche Schnittstellendokumentation für solche angrenzenden Systeme kann relevante Informationen für das zu entwickelnde System enthalten.

- **Durch den Geschäftsprozess generierte Dokumente**

In vielen Geschäftsprozessen werden Dokumente wie Versicherungsverträge, Rechnungen oder Kundenlisten erzeugt. Solche Dokumente enthalten oftmals wertvolle Informationen für jedes entwickelte System, um diesen Prozess zu unterstützen.

- **In technischen Analysen erzeugte Dokumente**

Die Entwicklung von Systemen, einschließlich Hardware- oder mechanischen Komponenten, umfasst häufig die Analyse der Ergebnisse von Simulationen, Sicherheitsbewertungen oder etablierten Methoden wie Quality Function Deployment (QFD) oder Fehlermöglichkeits- und -einfluss-Analyse (FMEA).

Ebenso wie Stakeholder, können auch Dokumente pragmatisch und systematisch identifiziert werden.

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.

Bei der *systematischen Identifizierung von Dokumenten* kann der Requirements Engineer Folgendes:

- Nach Repräsentanten typischer Dokumentenarten suchen (z. B.: Welche technischen Normen gelten für eine Zeitmaschine?)
- Nach Verweisen auf andere möglicherweise relevante Dokumente in zuvor identifizierten Dokumenten suchen (z. B. werden in ISO 26262 andere Normen zitiert, die für unsere Zeitmaschine gelten könnten?)
- Bereits identifizierte Stakeholder nach relevanten Dokumenten fragen (z. B. Dr. Emmet Brown fragen, ob für den Fluxkompensator bereits eine Dokumentation vorliegt?)
- Nach Dokumentation zu bereits als relevant identifizierten Systemen suchen (vgl. 2.4) (Gibt es z. B. ein Benutzerhandbuch für TARDIS oder die DeLorean-Zeitmaschine?)

Bei der *systematischen Identifizierung von Dokumenten* legt der Requirements Engineer Ermittlungsaktivitäten fest, die auf die Identifizierung von Dokumenten ausgerichtet sind. 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

ID	RS_EA_06
Ermittlungsziel	Mindestens ein Dokument zur Physik des Zeitreisens finden
Ergebnisqualität	Bezeichnung des Dokuments, Autor, Veröffentlichungsdatum, Speicher-/Aufbewahrungsort
Anforderungsquelle(n)	Bibliothek, das Internet
Ermittlungstechnik	Perspektivenbasiertes Lesen

Abbildung 13: Beispiel für eine informationsbasierte Ermittlungsaktivität

ID	RS_EA_07
Ermittlungsziel	Herausfinden, welche rechtsverbindlichen Dokumente für Zeitreisen relevant sind
Ergebnisqualität	Bezeichnung des Dokuments; ausstellende Behörde; Veröffentlichungsdatum; Region, in der es anwendbar ist; Speicher-/Aufbewahrungsort
Anforderungsquelle(n)	Doctor Who, der Rechtsberater/Anwalt des Entwicklungsunternehmens
Ermittlungstechnik	Interview

Abbildung 14: Beispiel für eine dokumentbasierte Ermittlungsaktivität

Der Requirements Engineer muss entscheiden, welche der zusammengetragenen Dokumente welchen potenziellen Wert als Anforderungsquellen haben. Daher müssen die Dokumente durchgesehen und auf ihre potenzielle Nützlichkeit hin bewertet werden.

In Abhängigkeit vom Kontext können auch andere Kriterien relevant sein. Neben dem Inhalt beeinflussen unter Umständen auch andere Aspekte den Wert (und damit die Priorität) eines Dokuments als Anforderungsquelle:

- **Verfügbarkeit:** Dokumente können vertraulich sein und erfordern, dass eine bestimmte Sicherheitsüberprüfung durchgeführt wird, bevor eine Person Zugriff erhält.
- Verhältnis von **Größe**, oder noch besser, geschätzter Größe und Inhalt: Gilt für ein Projekt ein straffer Zeitraum, können kleine Dokumente, die wichtige Informationen enthalten, wertvoller sein als große Dokumente mit vielen irrelevanten Details.
- **Alter:** Je älter ein Dokument ist, umso höher ist die Wahrscheinlichkeit, dass sein Inhalt veraltet ist.
- **Relevanz:** Das Anforderungsdokument des Vorgängersystems kann relevanter sein als das Anforderungsdokument eines angrenzenden Systems.

2.3.2 Dokumentationsschema für Dokumente

Sämtliche, zu Dokumenten gesammelten Informationen, müssen ausreichend dokumentiert werden. Eine solche Dokumentation sollte mindestens Folgendes beinhalten:

- Titel
- Ort, wo das Dokument aufbewahrt wird (z. B. physischer Ordnername, Link zu digitalem Dokument)
- Version des Dokuments
- Kurze Beschreibung (welche Art von Information das Dokument mutmaßlich enthält)
- Relevanz

Abhängig vom Kontext können auch weitere Informationen relevant sein. Beispiele:

- Für das Dokument verantwortliche Person (Dokument-Owner)
- Person, die das Dokument zur Liste hinzugefügt hat (relevant, wenn mehr als eine Person die Liste aktualisiert)
- Datum, zu dem das Dokument zur Liste hinzugefügt wurde
- Datum, zu dem das Dokument zuletzt durchgesehen wurde (Review in Bezug auf Anforderungen oder auf Verfügbarkeit einer neuen Version)
- Größe des Dokuments

Dokumente weisen immer bestimmte Beziehungen zu Stakeholdern auf, die ebenfalls festgehalten werden sollten. Daher ist es unter Umständen sinnvoll, die Dokumenten- mit der Stakeholder-Dokumentation zu verknüpfen.

Nachfolgend sehen Sie Beispiele für Beziehungen zwischen Dokumenten und Stakeholdern.

- Stakeholder, die auf die Relevanz des Dokuments hingewiesen haben
- Autor, 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 dokumentenbezogenen Informationen aktuell halten. Dies beinhaltet erneute Überprüfungen, 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

Im Kontext des zu entwickelnden Systems können auch andere Systeme vorhanden sein, die Anforderungsquellen darstellen. Sobald ein System als potenzielle Anforderungsquelle identifiziert wurde, wird sie Teil des Systemkontexts (siehe [IREB2020]).

In Abschnitt 2.4.1 behandeln wir die Identifizierung und Auswahl solcher Systeme und in Abschnitt 0 besprechen wir die Dokumentation von Systemen als Anforderungsquelle.

2.4.1 Systeme als Anforderungsquelle identifizieren und auswählen

Anforderungen können aus vielen verschiedenen Arten von Systemen abgeleitet werden. Je nach Charakter eines Entwicklungsprojekts, können andere Systeme eine hohe, mittlere oder niedrige Bedeutung als Anforderungsquelle haben. Bei Systemen, die viele Schnittstellen aufweisen, gibt es wahrscheinlich zahlreiche Systeme, die als Anforderungsquellen relevant sind. Wird ein neues System als Ersatz für ein oder mehrere vorhandene Systeme entwickelt, dann werden diese Systeme eine hohe Relevanz als Anforderungsquellen aufweisen.

Selbst für neue Systeme können in anderen Anwendungsgebieten vorhandene Systeme relevant sein (siehe auch Abschnitt 3.2.2 Analogietechnik).

Folgende Systemarten können als Anforderungsquellen relevant sein (unvollständige Aufzählung):

- **Angrenzende Systeme, einschließlich Altsystemen**
Damit das neue System mit den angrenzenden Systemen interagieren kann, ist es wesentlich, diese Systeme und vor allem deren Schnittstellenspezifikationen/-anforderungen zu kennen.
- **Systeme, die auf derselben Plattform oder in derselben Umgebung/Umwelt betrieben werden**
Wenn das zu entwickelnde System in eine bestehende Plattform, eine vorhandene Umgebung/Umwelt integriert werden soll, dann werden aus diesen bereits vorhandenen Elementen Anforderungen (überwiegend Randbedingungen) resultieren. Das können etwa ein Usability-Konzept oder technische Anforderungen sein.
- **Konkurrenzsysteme**
Wettbewerber sind stets Antriebskräfte für Innovationen. Daher ist es sehr wichtig, die Produkte von Wettbewerbern zu kennen, seien es serienmäßig hergestellte Produkte oder solche, die innerhalb eines Unternehmens entwickelt werden, z. B. von

anderen Abteilungen oder Büros. Ist Letzteres der Fall, sollten wir sowohl nach potenziellen Synergien Ausschau halten, aber auch darauf achten, dass sich Lösungen zu internen Wettbewerbern entwickeln könnten.

- **Systeme, die ähnliche Daten verarbeiten, ähnliche Funktionalitäten haben oder ähnliche Benutzungsschnittstellen bereitstellen**

Es gibt kaum ein System, das einzigartig ist. Tatsächlich ist es nicht unbedingt wünschenswert, ein einzigartiges System zu entwickeln.

Systeme, die Ihrem zu entwickelnden System ähnlich sind, können wertvolle Quellen für Anforderungen darstellen. Sei es, weil sie ähnliche Daten verarbeiten (z. B. Münzautomaten vs. Registrierkassen vs. Onlineshops), ähnliche Funktionalitäten aufweisen (z. B. Verkaufsautomaten für Zugtickets vs. Getränkeautomaten vs. Geldautomaten) oder eine ähnliche Benutzungsoberfläche bereitstellen (z. B. Fernbedienungen vs. Taschenrechner).

- **Vorgängersysteme, die ersetzt werden sollen**

Soll ein Vorgängersystem ersetzt werden, dann stellt das alte System in der Regel die wichtigste Anforderungsquelle dar. Es ist jedoch wichtig, dass der Blickwinkel durch das alte System nicht zu sehr eingeschränkt wird und man offen für Verbesserungen und Ergänzungen ist.

- **Zukünftige Systeme (in Erstellung)**

Es ist wichtig, im Kontext des zu entwickelnden Systems auf künftige Systeme zu achten. Es könnten sich bereits Systeme in der Entwicklungsphase befinden. Solche zukünftigen Systeme können unter Umständen das Umfeld des zu entwickelnden Systems und damit dessen Anforderungen verändern.

Ebenso wie Stakeholder und Dokumente, können auch Systeme pragmatisch und systematisch identifiziert werden.

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.

Bei der *systematischen Identifizierung* kann der Requirements Engineer:

- die Dokumentation zum Systemkontext verwenden
- bereits identifizierte Stakeholder nach relevanten Systemen fragen (z. B. : Welche Systeme sind geplant oder werden derzeit entwickelt, die für das zu entwickelnde System relevant sein könnten?)
- zuvor identifizierte Dokumente nach Informationen über relevante Systeme durchsuchen (z. B. die Dokumentation der Systemarchitektur eines angrenzenden Systems nach Informationen über dessen Schnittstellen zu anderen Systemen durchsuchen)
- Techniken zur Ideenfindung verwenden, um potenziell analoge Systeme zu identifizieren (z. B. Brainstorming)
- eine Marktanalyse durchführen, um Konkurrenzsysteme zu ermitteln

(d. h.: Welche anderen Systeme sind verfügbar oder werden entwickelt, die demselben Zweck dienen?)

- Altsysteme in Betracht ziehen
(z. B.: Welche Altsysteme dienen einem ähnlichen Zweck?)

Bei der *systematischen Identifizierung von Systemen* legt der Requirements Engineer Ermittlungsaktivitäten fest, die auf die Identifizierung von Systemen ausgerichtet sind. Dabei müssen zwei verschiedene Arten von Ermittlungszielen 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

Abbildung 15 und Abbildung 16 enthalten Beispiele für informationsbasierte und systembasierte Ermittlungsaktivitäten.

ID	RS_EA_08
Ermittlungsziel	Mindestens fünf Zeitreisemaschinen von Wettbewerbern finden
Ergebnisqualität	Systemname; Ersteller; Ort, an dem das System verfügbar ist (oder Zeitpunkt, zu dem es verfügbar ist). Im Idealfall sollten die Zeitmaschinen funktionieren, andernfalls wären auch nicht funktionierende Zeitmaschinen akzeptabel.
Anforderungsquelle(n)	Bibliothek, das Internet, Konferenz zu Zeitreisen
Ermittlungstechnik	Perspektivenbasiertes Lesen

Abbildung 15: Beispiel für eine systembasierte Ermittlungsaktivität

ID	RS_EA_09
Ermittlungsziel	Mindestens fünf Systeme finden, die Platz für zwei Erwachsene bieten
Ergebnisqualität	Systemname; Ersteller; Ort, wo es verfügbar ist
Anforderungsquelle(n)	Sämtliche Entwickler
Ermittlungstechnik	Workshop

Abbildung 16: Beispiel für eine informationsbasierte Ermittlungsaktivität

2.4.2 Dokumentationsschema für Systeme

Sämtliche, zu Systemen gesammelten Informationen müssen ausreichend dokumentiert werden. Diese Dokumentation sollte für jedes System mindestens Folgendes beinhalten:

- Name
- Art (z. B. Konkurrenzsystem, Vorgängersystem, Nachbarsystem usw.)
- Daten, Funktionalitäten, Prozesse, Nutzergruppen (kurze Beschreibung)
- Version
- Relevanz

Abhängig vom Kontext können auch weitere Informationen relevant sein. Beispiele:

- Für das System verantwortliche Person (System-Owner)
- Person, die das System zur Liste hinzugefügt hat (relevant, wenn mehr als eine Person die Liste aktualisiert)
- Datum, zu dem das System zur Liste hinzugefügt wurde
- Datum, zu dem das System zuletzt durchgesehen wurde (Review in Bezug auf Anforderungen oder auf Verfügbarkeit einer neuen Version)
- Anzahl der Benutzer

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

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

Systeme weisen immer bestimmte Beziehungen zu den Stakeholdern auf, die ebenfalls festgehalten werden sollten. Daher ist es unter Umständen sinnvoll, die System- und die Stakeholder-Dokumentation zu verknüpfen.

Nachfolgend sehen Sie Beispiele für Beziehungen zwischen Systemen und Stakeholdern:

- 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)
- Stakeholder, die auf die Relevanz des Systems hingewiesen haben
- System-Owner
- Organisationen, die daran beteiligt sind, die Einhaltung der geltenden Gesetze und Normen für das System zu prüfen

Der Requirements Engineer muss die systembezogenen Informationen aktuell halten. Dies beinhaltet die Überprüfung, ob zwischenzeitlich weitere Systeme relevant geworden sind, oder ob zuvor identifizierte Systeme inzwischen möglicherweise an Relevanz verloren haben. Änderungen, Aktualisierungen und Versionsnummern sollten dabei im Blick behalten werden.

Informationen über Systeme finden sich für gewöhnlich in Dokumenten. Solche Dokumente sollten separat als Anforderungsquellen verwaltet werden (siehe 2.3). Die Beziehung zwischen Dokument und System sollte ebenfalls dokumentiert werden.

3 Anforderungsermittlung

Für die Ermittlung von Anforderungen wurde eine Fülle von Techniken entwickelt. Für den Lehrplan Requirements Elicitation und damit für dieses Handbuch haben wir einige Techniken ausgewählt, die wir hier strukturiert darstellen.

Da wir dem Requirements Elicitation Modul neue Techniken hinzugefügt haben, haben wir auch die für das Foundation Level verwendete Kategorisierung angepasst. Durch diese Kategorisierung wird die Präsentation unserer 19 Techniken und Denkwerkzeuge für die Anforderungsermittlung verständlicher. Diese Differenzierung ist selbstverständlich künstlich: In der Praxis gibt es keine klare Trennung zwischen diesen Techniken. Zur Darstellung und Lehre ist diese Unterscheidung jedoch wichtig für die Bereitstellung einer Struktur und zur Beschreibung des primären Schwerpunkts der einzelnen Techniken.

Alle in diesem Kapitel beschriebenen Ermittlungstechniken sind wie folgt strukturiert:

- **Was ist das?**
beschreibt kurz die wichtigsten Faktoren der Ermittlungstechnik.
- **Die Rolle der Teilnehmer (sofern zutreffend)**
beschreibt, welche Rollen bei der Anwendung diese Technik beteiligt sind.
- **Vorbereitung**
beschreibt, welche vorbereitenden Maßnahmen für die Anwendung der Ermittlungstechnik unternommen werden müssen.
- **Anwendung**
beschreibt, wie die Ermittlungstechnik angewendet wird und was bei der Anwendung berücksichtigt werden muss.
- **Ergebnisverarbeitung**
beschreibt, welche Maßnahmen erforderlich sind, um die Ergebnisse zu verarbeiten (siehe nachfolgenden Hinweis).
- **Typische Arbeitsprodukte**
benennt die bei der Anwendung der Ermittlungstechnik üblicherweise verwendeten oder hergestellten Arbeitsergebnisse (siehe nachfolgenden Hinweis).
- **Möglichkeiten**
beschreibt die möglichen Vorteile der Ermittlungstechnik.
- **Herausforderungen**
beschreibt die möglichen Stolpersteine der Ermittlungstechnik.
- **Varianten (sofern zutreffend)**
beschreibt die Varianten der Ermittlungstechnik und stellt eine kurze Beschreibung bereit.

Figure 17

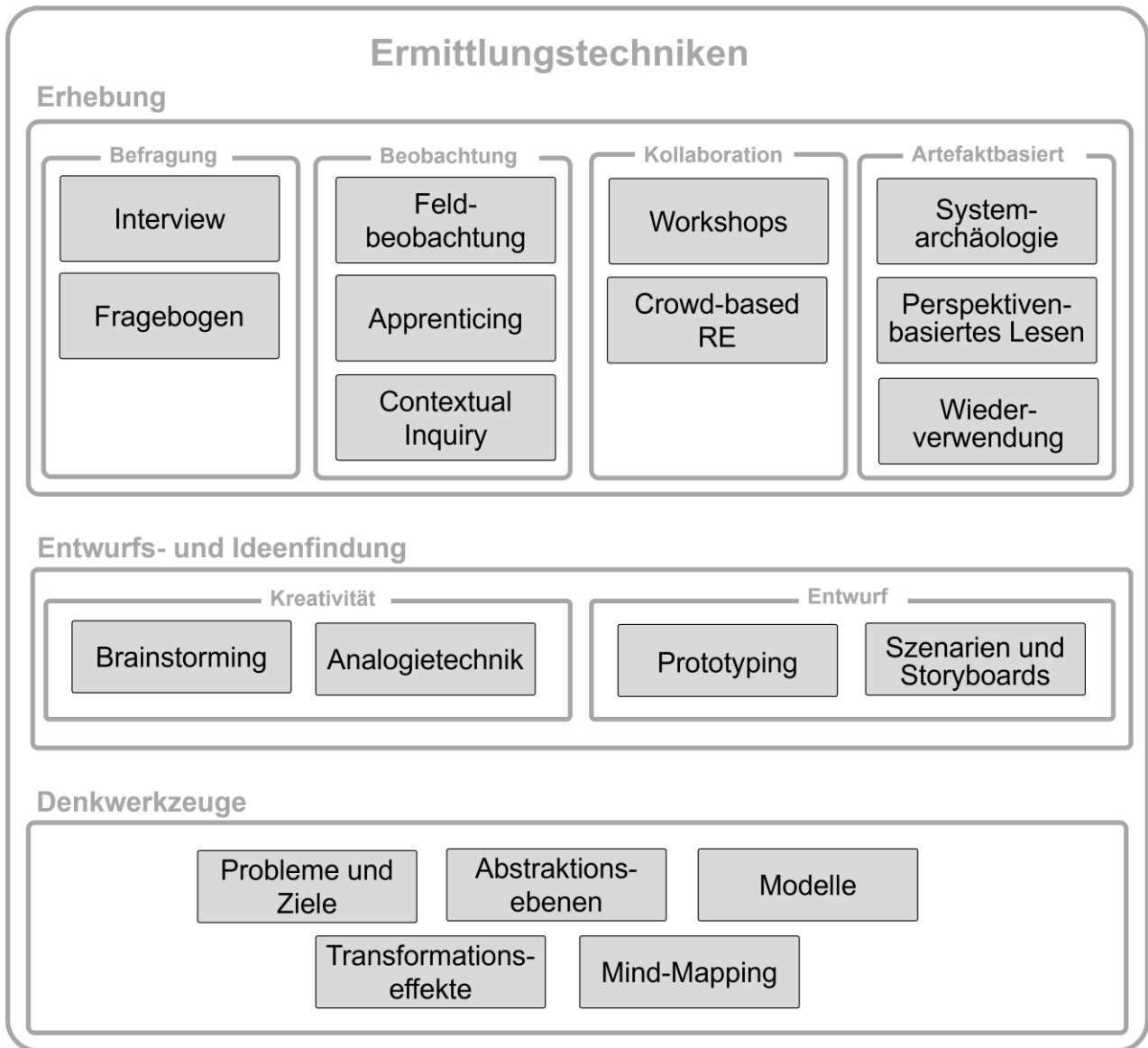


Abbildung 17: Überblick über Ermittlungstechniken

Hinweis:

Die bei Anwendung einer Ermittlungstechnik erzielten Ergebnisse sind Rohinformationen, die verarbeitet und in Anforderungen formuliert werden müssen. Letzteres ist Teil der Anforderungsdokumentation und wird als solches nicht in diesem Handbuch behandelt.

Nach jeder Ermittlungsaktivität muss die Dokumentation mit den gesammelten Informationen aktualisiert werden, z. B. Anforderungsdokumentation, Systemkontextdokumentation, Anforderungsressourcen-Dokumentation. Dann sollte entschieden werden, ob weitere Ermittlungsaktivitäten nötig sind.

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



Der Einsatz Künstlicher Intelligenz (KI), insbesondere generativer KI und sogenannter "Large Language Models" (LLM), kann die Planung und Durchführung bewährter Ermittlungstechniken gezielt unterstützen und effizienter gestalten. So können KI-gestützte Werkzeuge dem Requirements Engineer helfen, Interviews systematisch vorzubereiten, beispielsweise durch automatische Generierung relevanter Fragestellungen oder Themenbereiche. Bei Beobachtungstechniken kann KI dazu beitragen, sinnvolle Beobachtungsziele und -kriterien anhand früherer Erfahrungen oder vorhandener Projektdaten vorzuschlagen. Auch in der Workshop-Planung bietet KI Unterstützung, indem sie Themen und Inhalte identifiziert, die für bestimmte Stakeholder-Gruppen besonders relevant sind. Durch die automatisierte Auswertung bereits vorhandener Daten und Dokumente erkennt KI zudem Muster und Zusammenhänge, die bei der Auswahl geeigneter Ermittlungstechniken und deren zielgerichteter Anwendung hilfreich sind. Dieser technologiegestützte Ansatz verbessert somit die Qualität und Effizienz bekannter Methoden, während gleichzeitig die menschliche Expertise und Verantwortung des Requirements Engineers weiterhin unverzichtbar bleibt. Insbesondere bei Anforderungen, die von der KI ergänzt oder erstellt werden, muss der Requirements Engineer stets kritisch hinterfragen, ob diese tatsächlich gewünscht sind und ob die Traceability (z.B. Anforderungsquelle, Zurechenbarkeit, Notwendigkeit) dokumentiert ist.

3.1 Erhebungstechniken

Erhebungstechniken sind etablierte Methoden zur Anforderungsermittlung. Sie helfen, Leistungs- und Basisfaktoren zu ermitteln. Wir unterteilen Erhebungstechniken in Fragetechniken (Interview und Fragebogen), Beobachtungstechniken (Feldbeobachtung, Apprenticing, Contextual Inquiry) Kollaborationstechniken (Anforderungs-Workshop und Crowd-based Requirements Engineering) und artefaktbasierte Techniken (perspektivenbasiertes Lesen, Systemarchäologie und Wiederverwendung von Anforderungen), Abbildung 17: Überblick über Ermittlungstechniken.

3.1.1 Fragetechniken

Fragetechniken zielen darauf ab, Stakeholdern die passenden Fragen zu stellen, um Antworten zu erhalten, aus denen Anforderungen abgeleitet werden können.

Offene und geschlossene Fragen

Es ist für sämtliche Fragetechniken, die in diesem Abschnitt behandelt werden, wichtig, den Unterschied zwischen offenen und geschlossenen Fragen zu kennen.

Bei geschlossenen Fragen werden die Antworten entweder durch die Frage selbst oder durch ausdrücklichen Verweis auf die verfügbaren Antworten gegeben oder festgelegt (z. B.:

Wie alt sind Sie? Essen Sie gerne Fisch?). Geschlossene Fragen führen zu quantitativen Daten, d. h. zu Daten, die ohne weitere Anpassung statistisch verarbeitet werden können.

Offene Fragen hingegen ermöglichen eine freie Antwort. So können narratives oder argumentatives Wissen abgefragt und qualitative Daten ermittelt werden (z. B. Was ist Ihre Lieblingspeise? Wie können Flugzeuge fliegen?). Qualitative Daten können nur durch kognitive Analysen ausgewertet werden.

Im Kontext der Anforderungsermittlung werden vorwiegend qualitative Daten benötigt. Aus diesem Grund wird meist auf offene Fragen zurückgegriffen. In manchen Situationen werden aber auch quantitative Daten benötigt (z. B. zur Beurteilung, ob die Aussage eines repräsentativen Nutzers von anderen Nutzern bestätigt wird). Hierfür sollten geschlossene Fragen verwendet werden.

Bei der Nutzung von Fragetechniken in einer Ermittlungsaktivität sollte die Definition der Ergebnisqualität beinhalten, ob qualitative oder quantitative Daten ermittelt werden sollten.

Nachfolgend werden die zwei relevantesten Fragetechniken beschrieben: Interview, Fragebogen.

3.1.1.1 Interview

Aufgrund ihrer Flexibilität sind Interviews wahrscheinlich eine der am häufigsten verwendeten Ermittlungstechniken. Sie erfordern keine große Vorbereitung oder spezielle Werkzeuge und können verwendet werden, um allgemeinere sowie sehr spezifische Anforderungen in Erfahrung zu bringen. Anforderungen, die über ein Interview ermittelt werden, sind in der Regel Leistungsfaktoren, da der Befragte bewusste Informationen äußert. Sie können jedoch auch, durch die Nutzung der richtigen Fragetechnik und die Beobachtung der Reaktionen des Benutzers, Basisfaktoren oder Begeisterungsfaktoren identifizieren.

Wenngleich ein Interview nicht sehr kompliziert ist und die meisten Personen ein gutes Verständnis davon haben, was das ist, sind klare Ziele und eine detaillierte Vorbereitung nötig, damit die Interviewzeit gut genutzt wird und nützliche und nachhaltige Ergebnisse erreicht werden.

Was ist ein Interview?

In einem Interview stellt der Requirements Engineer einem oder mehreren Stakeholdern Fragen, um neue Anforderungen zu ermitteln oder bestehende weiter zu verfeinern. Die Stakeholder beantworten diese Fragen und der Requirements Engineer (oder ein Assistent) zeichnet die Informationen auf, die er erhält. Ein Interview unterscheidet sich von einer reinen Konversation hauptsächlich in drei Faktoren: Rolle der Teilnehmer, Vorbereitung und Ergebnisverarbeitung.

Die Rolle der Teilnehmer

In einem Interview sind drei Rollen obligatorisch: *Interviewer*, *Interviewter* und *Protokollant*.

Der Requirements Engineer nimmt die Rolle des *Interviewers* ein. Es liegt in seiner Verantwortung, das Interview vorzubereiten (siehe Abschnitt Vorbereitung) und durchzuführen (siehe Abschnitt Ergebnisverarbeitung) und die gesammelten Informationen zu verarbeiten (siehe Abschnitt Ergebnisverarbeitung). Ein Interviewer sollte folgende Eigenschaften mitbringen: *fachkundig, strukturiert, klar, rücksichtsvoll, verständnisvoll, offen, steuernd, kritisch, erinnerungs- und interpretationsfähig* (basierend auf [Kvale2008], detailliert erläutert im Abschnitt zur Anwendung).

Der Stakeholder ist der *Interviewte*. Er beantwortet die Fragen, die der Interviewer stellt und äußert auf diese Weise seine Anforderungen.

Die Aufgabe des *Protokollanten* besteht darin, sämtliche vom Interviewten bereitgestellten Informationen aufzuzeichnen. Er muss vorab wissen, welches Ermittlungsziel mit dem Interview verfolgt wird, und er muss den Interviewleitfaden kennen und verstehen. Nur wenn der Protokollant den Kontext des Interviews und des Themas (einschließlich Terminologie) ausreichend gut kennt, ist er in der Lage, dem Interview zu folgen und zu entscheiden, welche Informationen festzuhalten sind.

Der Interviewer kann ebenfalls die Rolle des Protokollanten übernehmen. Dies sollte allerdings nach Möglichkeit vermieden werden, da der Interviewer in diesem Fall zwischen zwei verschiedenen kognitiven Modi hin- und herspringen muss, was zusätzliche Energie erfordert und sich in der Regel negativ auf die Leistung in beiden Rollen auswirkt. Die Rolle des Protokollanten kann durch ein Ton- oder Videoaufzeichnungsgerät ersetzt werden. Die Vor- und Nachteile werden im Abschnitt über Vorbereitung behandelt.

Vorbereitung

Definieren Sie das bzw. die Ermittlungsziele und die erforderliche Ergebnisqualität im Rahmen der Definition der Ermittlungsaktivität (siehe 1.3.1).

Wählen Sie einen oder mehrere geeignete Stakeholder für das Interview aus, von denen Sie erwarten, dass sie Ihre Fragen beantworten können, damit Sie Ihr(e) Ermittlungsziel(e) und die erforderliche Ergebnisqualität erreichen. Sie werden im Voraus niemals genau wissen, ob Sie die richtigen Stakeholder ausgewählt haben. Mit einem guten Stakeholder-Management und sorgfältig recherchierter Stakeholder-Dokumentation werden Sie jedoch die Chance erheblich steigern, eine gute Wahl zu treffen.

Bereiten Sie einen Interviewleitfaden vor. Dafür stehen Ihnen viele unterschiedliche Möglichkeiten zur Verfügung: Manche bevorzugen eine Liste mit Aufzählungspunkten, aus der die gewünschte Reihenfolge und Hierarchie der Fragen hervorgeht; andere bevorzugen einen Leitfaden in der Art von Mindmaps, bei welchen die Fragen im Uhrzeigersinn rund um das Interviewziel angeordnet sind. Letzteres kann es erleichtern, den Gedankengängen des Interviewten zu folgen, wenn er bei der Beantwortung der ursprünglich gestellten Frage von einem Thema zum anderen springt.

Der Interviewer entwickelt eine Gesamtstruktur für das Interview, die im Interviewleitfaden abgebildet wird. Nachdem entschieden wurde, welche Fragen gestellt werden sollen und welche Fragen welche Priorität haben, sollte man darüber nachdenken, wie man die Einführung gestaltet, welchen Zweck das Interview hat, welche Frage zu Beginn und in

welcher Reihenfolge die übrigen Fragen gestellt werden sollten und wie das Interview enden sollte. Der Leitfaden muss nicht den exakten Wortlaut der zu stellenden Fragen enthalten (zumindest nicht in einem qualitativen Interview). Dennoch sollten Interviewer darüber nachdenken, wie sie ihre Fragen formulieren, um mehr *Klarheit* in ihren Fragestellungen zu erreichen.

Zeit und Ort des Interviews sind eine weitere organisatorische Angelegenheit, die es zu klären gilt. Der Interviewer muss einen geeigneten Zeitraum festlegen, der in den Zeitplan des Befragten passt. Es ist ratsam, zusätzlich zur geplanten Interviewzeit 15 Minuten als Aufwärmphase und als generellen Puffer einzuplanen. Das Interview sollte 60 Minuten nicht überschreiten. Für die meisten Interviewer und Interviewten sind 20 bis 40 Minuten ein guter Zeitrahmen.

Das Interview sollte in einem separaten Raum stattfinden, um eine konzentrierte und vertrauliche Atmosphäre zu ermöglichen. Im Idealfall sollte der Ort des Interviews für den Interviewten nah und komfortabel sein.

Der Interviewer muss im Voraus entscheiden, ob das Interview aufgezeichnet werden soll. Es bestehen drei Optionen: Der Interviewer erstellt selbst ein Protokoll, ein Protokollant erstellt das Protokoll oder das Gespräch wird per Audio- oder Videoaufnahme aufgezeichnet.

Das sind die Vor- und Nachteile der drei Optionen:

	Interviewer erstellt Protokoll	Protokollant erstellt Protokoll	Audio-/Videoaufzeichnung
Vorteile	Keine zusätzliche Person/kein Gerät erforderlich Während Notizen gemacht werden, haben Interviewer und Interviewter Zeit zum Nachdenken.	Der Interviewer kann sich voll auf das Fragenstellen und das Zuhören/Beobachten des Interviewten konzentrieren. Interviewer und Protokollant haben zwei Sichtweisen auf dasselbe Ereignis (bietet die Möglichkeit, kognitive Voreingenommenheit aufzudecken).	Der Interviewer kann sich voll auf das Fragenstellen und das Zuhören/Beobachten des Interviewten konzentrieren. Nach dem Interview stehen alle Informationen ungefiltert zur Verfügung.
Nachteile	Einige Informationen können verloren gehen oder verzerrt werden. Der Interviewer muss zwischen zwei kognitiven Modi wechseln (erfordert Energie). Während Notizen gemacht werden, können unangenehme Pausen entstehen. Notizen sind gewöhnlich sehr kurz und werden unter Umständen nach dem Interview nicht mehr verstanden.	Einige Informationen können verloren gehen oder verzerrt werden. Wenn der Protokollant nicht vollständig mit dem Interviewthema vertraut ist, versteht er manche Informationen möglicherweise nicht richtig und macht falsche oder unvollständige Notizen.	Das Filtern von Informationen beginnt erst nach dem Interview; das gesamte Interview muss nochmals angehört werden. Abhängig von der nationalen Gesetzgebung oder von Unternehmensvorschriften können rechtliche Einschränkungen gelten. Der Interviewte kann die Aufzeichnung untersagen.

Hinweis 3.1.1:
Häufig werden die Optionen 1 und 3 kombiniert, d. h. das Interview wird aufgezeichnet und der Interviewer macht sich dennoch einige Notizen (er kann sich jedoch auf das Stellen der Fragen und Zuhören konzentrieren).

Wenn eine Aufzeichnung des Interviews geplant ist, ist es ratsam, vorab eine Genehmigung vom Interviewten und seiner Organisation einzuholen.

Anwendung

Während des Interviews führt der Interviewer den Interviewten, indem er ihm Fragen stellt. Der Interviewer setzt dabei folgende Qualifikationen [Kvale2008] in die Praxis um:

- *Fachkundig*: hat umfassende Kenntnis des Themas; weiß, welche Aspekte wichtig sind und verfolgt werden sollten
- *Strukturiert*: bereitet eine Struktur für das Interview vor und folgt dieser Kommuniziert die Struktur während des Interviews an den Interviewten
- *Klar*: verwendet einfache Sprache, stellt klare, einfache, leicht verständliche und kurze Fragen
- *Rücksichtsvoll*: lässt jeden aussprechen; gibt Zeit zum Nachdenken; toleriert Pausen
- *Verständnisvoll*: schenkt dem Gesagten und wie es geäußert wird volle Aufmerksamkeit; zeigt sich im Umgang mit dem Interviewten empathisch
- *Offen*: reagiert auf wichtige Aspekte, die der Interviewte anspricht; ist offen für neue Aspekte, die der Interviewte einbringt und verfolgt diese weiter
- *Steuern*: Der Interviewer ist sich darüber im Klaren, was er herausfinden möchte. Er steuert den Verlauf des Interviews und hat keine Angst vor Unterbrechungen durch Exkurse.
- *Kritisch*: ist darauf vorbereitet, das Gesagte auf den Prüfstand zu stellen, beispielsweise indem er Inkonsistenzen in den Antworten des Interviewten anspricht
- *Erinnerungsfähig*: kann sich frühere Aussagen ins Gedächtnis rufen und damit in Bezug setzen, was zu einem früheren Zeitpunkt im Interview gesagt wurde
- *Interpretationsfähig*: klärt und erweitert die Bedeutungen der Aussagen des Interviewten; äußert Interpretationen des Gesagten, die vom Interviewten bestätigt oder negiert werden können

Die Stakeholder (Interviewten) beantworten diese Fragen und der Requirements Engineer (Interviewer) hört aufmerksam zu und prüft Verschiedenes, z. B.:

- ob der Stakeholder die Frage verstanden hat und die gewünschten Informationen liefert
- ob der Requirements Engineer versteht, was der Stakeholder sagt
- ob eine Frage vollständig beantwortet wird
- ob der Stakeholder die relevanten nonverbalen Informationen sendet
- ob der Protokollant die erforderlichen Informationen notiert

Der Protokollant hört dem Interviewer und dem Interviewten zu, filtert die relevanten Informationen aus dem Gespräch und notiert sie. Er beobachtet den Interviewer genau, um nonverbale oder verbale Signale darüber zu erfassen, was notierenswert ist.

[Port2013] und [BaCC2015] stellen noch weitere Erkenntnisse über die Kunst der Interviewführung aus. Den Schwerpunkt legen sie dabei auf Interviews mit Nutzern.

Nach dem Interview bereiten der *Interviewer* und/oder der *Protokollant* das Protokoll des Interviews vor und senden dies zum Review an den/die Interviewten. Dies dient zweierlei Zwecken: Zum einen stellen Sie sicher, dass Sie sämtliche Informationen aus dem Interview richtig verstanden und keine wichtigen Aspekte vergessen haben. Zum anderen zeigen Sie Ihre Wertschätzung für die Zeit und den Input des Interviewten.

Hinweis 3.1.2:

Legen Sie ein Datum fest, bis zu dem Sie das Feedback des Interviewten benötigen. Es ist außerdem ratsam, dem Interviewten einige Tage vor Ablauf dieses Termins eine kurze, freundliche Erinnerung zu senden.

Ergebnisverarbeitung

Nach den Interviews müssen die gesammelten (Roh-)Daten analysiert und zu nützlichen Informationen zusammengefasst werden. Dies könnten Anforderungen, Bedürfnisse, Ziele, Probleme, Benutzergruppen, Szenarien, Prozesse, Arbeitsergebnisse usw. sein.

Bei der Verarbeitung der gesammelten Daten kann ein Affinitätsdiagramm [BaCC2015] nützlich sein. Sehen Sie sich das folgende Beispiel an, das die beiden Hauptaspekte eines Affinitätsdiagramms verdeutlicht:

1. Extrahieren Sie die Erkenntnisse aus Ihren gesammelten Daten, und notieren Sie jede Erkenntnis auf einer (grünen) Karte.
2. Clustering: Gruppieren Sie Ihre Erkenntnisse/Karten in Clustern, und benennen Sie die einzelnen Gruppen (gelbe Karten).



Abbildung 18: Affinitätsdiagramm

Typische Artefakte

Bei der Vorbereitung, Durchführung und Nachbereitung eines Interviews werden üblicherweise die folgenden Arbeitsergebnisse erzeugt oder aktualisiert:

- Interviewleitfaden
- Notizen und/oder Audio-/Videoaufzeichnungen, die während des Interviews gemacht wurden
- Abschließende Notizen des Interviews (an alle Teilnehmer zu senden)

Möglichkeiten

Der große Vorteil eines Interviews ist das direkte Feedback – nicht nur verbal, sondern auch nonverbal. Indem Sie den Interviewten genau beobachten, können Sie deutlich mehr Informationen aufnehmen als lediglich über das Gesagte. Sie können außerdem unmittelbar auf alle erhaltenen Informationen reagieren.

Ein Interview kann auch kurzfristig realisiert werden. Es sollte niemals unvorbereitet stattfinden, obgleich ein erfahrener Interviewer vielleicht nur rund eine Stunde vorab für eine ausreichende Vorbereitung benötigt.

Herausforderungen

Die Durchführung und besonders die Auswertung von Interviews sind sehr zeitaufwendig. Fehlen dem Requirements Engineer einige der genannten Qualifikationen, kann ein Interview leicht aus dem Ruder laufen und nicht die angestrebten Ergebnisse erbringen. Werden die falschen Fragen gestellt oder aber die richtigen Fragen auf die falsche Weise, dann kann dies die Nützlichkeit des Interviews drastisch verringern.

Varianten

Es gibt verschiedene Interviewvarianten, z. B.:

- **Offenes/qualitatives Interview:**
Das Interview folgt keiner rigiden Struktur. Der Interviewer kann während des Interviews von den ursprünglich vorbereiteten Themen oder Fragen abweichen. Wenn mehrere Interviews zum selben Thema durchgeführt werden, hat jedes Interview einen individuellen Verlauf.
Diese Variante ist in der Anforderungsermittlung recht üblich.
- **Teilweise standardisiertes Interview:**
Einige Fragen sind standardisiert und müssen in jedem Interview auf exakt dieselbe Weise gestellt werden. Im restlichen Verlauf ist es ein offenes Interview.
Der standardisierte Teil wird zum Sammeln quantitativer Daten verwendet.
- **Vollständig standardisiertes/quantitatives Interview:**
Die Reihenfolge der Fragen, ihr exakter Wortlaut und deren mögliche Antworten sind vorgegeben. Ziel dabei ist es, vergleichbare Ergebnisse bereitzustellen. Diese können direkt mit statistischen Methoden verarbeitet werden.
Diese Variante ist im Requirements Engineering nicht sehr häufig anzutreffen. Sie wird hauptsächlich in der Marktforschung verwendet.
- **Gruppeninterview:**

Mehrere Interviewpartner werden im Rahmen einer Interviewsitzung von einem Interviewer befragt.

Diese Variante hat den Vorteil, dass Meinungsverschiedenheiten zwischen den Interviewten direkt aufgegriffen werden können. Diese Art von Interview ist jedoch nicht zu empfehlen, wenn zwischen den Interviewpartnern persönliche Differenzen bestehen oder eine Person eine höherrangige Position innehat als andere Beteiligte. Beide Faktoren beeinflussen das Ergebnis des Interviews wahrscheinlich negativ.

3.1.1.2 Fragebogen

Was ist ein Fragebogen?

Bei einem Fragebogen sind mehrere Personen aufgefordert, dieselben Fragen, die in strukturierter Weise präsentiert werden, schriftlich zu beantworten. Man unterscheidet zwei Hauptarten: *quantitative* und *qualitative* Fragebögen.

Quantitative Fragebögen werden dazu verwendet, Hypothesen oder zuvor ermittelte Anforderungen zu bestätigen. Sie können schneller ausgewertet werden und liefern statistische Informationen.

In quantitativen Fragebögen verwendet man geschlossene Fragen (siehe Abschnitt 0).

Qualitative Fragebögen sind dazu geeignet, neue Anforderungen zu ermitteln. Sie liefern tendenziell komplexe Ergebnisse und sind daher in der Regel weitaus zeitaufwendiger in der Vorbereitung und auch in der Auswertung.

In qualitativen Fragebögen werden offene Fragen verwendet (siehe Abschnitt 3.1.1).

Quantitative Fragebögen können auch einige offene Fragen enthalten und qualitative Fragebögen geschlossene Fragen.

Die Rolle der Teilnehmer

In einer Umfrage, die anhand von Fragebögen⁵ durchgeführt wird, sind zwei Rollen erforderlich: Verfasser und Beantworter des Fragebogens.

Der Requirements Engineer nimmt die Rolle des *Verfassers des Fragebogens* ein. Gestaltung und Verteilung des Fragebogens liegen in seiner Verantwortung, ebenso wie die Auswertung der Ergebnisse.

Die *Beantworter*, welche gleichzeitig Stakeholder sind, erhalten den Fragebogen und sind dazu aufgefordert, diesen bis zu einem bestimmten Termin zu beantworten.

Abhängig davon, wie der Fragebogen ausgegeben wird, können auch andere Rollen involviert sein, beispielsweise eine Person für die Datenverarbeitung, die Daten von den Papierfragebögen in ein Datenverarbeitungssystem überträgt oder die Daten manuell bearbeitet.

⁵ Der Begriff „Fragebogen“ beschreibt das Werkzeug, anhand dessen anschließend eine Befragung per Fragebogen durchgeführt wird.

Vorbereitung

Da eine Umfrage anhand von Fragebögen nicht mehr korrigiert werden kann, nachdem die Fragebögen einmal verteilt worden sind, sollten Sie ausreichend Zeit und Überlegungen in die Vorbereitungsphase investieren.

Zunächst einmal sollten Sie klären, ob für Ihr Projekt in Bezug auf die Art und Weise der Durchführung einer Befragung irgendwelche gesetzlichen oder unternehmensspezifischen Einschränkungen gelten. In manchen Unternehmen müssen Befragungen von einem bestimmten Gremium genehmigt werden.

Definieren Sie das bzw. die Ermittlungsziele und die erforderliche Ergebnisqualität im Rahmen der Definition der Ermittlungsaktivität (siehe 1.3.1).

Wählen Sie für die Befragung geeignete Stakeholder aus. Wie viele Teilnehmer werden für ein werthaltiges oder repräsentatives Ergebnis benötigt? Ein quantitativer Fragebogen muss repräsentative Ergebnisse liefern. Bei einem qualitativen Fragebogen liegt der Schwerpunkt dagegen nicht unbedingt auf repräsentativen, sondern eher auf werthaltigen Ergebnissen, die neue Anforderungen liefern.

Bei der Auswahl der Teilnehmer sollten Sie die Liste der Stakeholder berücksichtigen. Stehen die Teilnehmer nicht auf der Liste oder fehlen Kontaktinformationen, dann sollten Sie die Stakeholder-Liste aktualisieren.

Legen Sie die maximale Länge des Fragebogens fest: Je länger der Fragebogen, desto weniger Antworten. Wie viele Fragen sind für Ihre Teilnehmer akzeptabel?

Wählen Sie die Form der Darstellung. Können Sie ein Umfragewerkzeug nutzen oder müssen Sie die Papierform verwenden?

Formulieren Sie die Fragen im Hinblick auf das Ermittlungsziel und die angestrebten Stakeholder. Verwenden Sie offene Fragen, um neue Aspekte und Anforderungen zu ermitteln; verwenden Sie geschlossene Fragen, um Anforderungen zu verifizieren, die Sie bereits ermittelt haben, oder um eine Hypothese auf Basis der bis dato erhobenen Anforderungen zu bestätigen oder zu entkräften.

Wenn Sie mit geschlossenen Fragen arbeiten, sollten Sie sich für einen Skalentyp entscheiden. Stellen Sie außerdem sicher, dass Sie in Ihrem Fragebogen nicht so viele unterschiedliche Skalentypen verwenden. Prüfen Sie bei allen geschlossenen Fragen, ob Sie die Option „nicht zutreffend“ bereitstellen müssen. Fehlt eine solche Option – vor allem bei obligatorischen Fragen –, kann dies das Ergebnis möglicherweise falsifizieren, da der Teilnehmer zur Beantwortung der Frage gezwungen ist, unabhängig davon, ob er sie beantworten kann oder nicht.

Legen Sie die Reihenfolge der Fragen fest und stellen Sie sicher, dass diese logisch aufgebaut ist. Achten Sie besonders darauf, dass die Antworten auf manche Fragen andere Fragen nicht ausschließen (z. B. „Trinken Sie Wein?“ und „Bevorzugen Sie Rotwein oder Weißwein?“, wenn der Teilnehmer die erste Frage mit „nein“ beantwortet, ist die andere Frage nicht mehr sinnvoll).

Planen Sie den Zeitrahmen für Ihre Befragung. Wann versenden Sie den Fragebogen? Wie viel Zeit sollten Sie für die Antworten einräumen? Wann sollten Sie eine freundliche Erinnerung versenden? Wie viel Zeit werden Sie für die Auswertung der Ergebnisse benötigen?

Testen Sie Ihren Fragebogen vor der Verteilung. Lassen Sie ihn von anderen durchsehen und von einigen Testkandidaten beantworten, um unklare Fragen zu identifizieren. Sobald der Fragebogen verteilt ist, können Sie keine Fehler mehr korrigieren. Führen Sie außerdem eine Versuchsauswertung der aus Ihren Testfragebögen erhaltenen Daten durch. Häufig fällt erst bei der Auswertung auf, dass man einen wichtigen Aspekt vergessen hat, oder dass die Fragen nicht genau die Daten liefern, die Sie untersuchen wollten. Ein weiterer Vorteil: Wenn Sie die Auswertung einmal durchgeführt haben, können Sie diese effizienter durchführen, wenn Ihnen die Daten aus den echten Fragebögen vorliegen.

Anwendung

Verteilen Sie die Fragen an die Teilnehmer. Berücksichtigen Sie beim Versenden des Fragebogens per E-Mail, ob Sie die Empfänger öffentlich oder vertraulich behandeln sollten. In der Regel werden die Empfänger in E-Mails vertraulich behandelt.

Kommunizieren Sie in der Ankündigung mit der Verteilung auch den Kontext des Fragebogens: Wer sind Sie und weshalb versenden Sie einen Fragebogen? Welchen Zweck verfolgen Sie mit der Befragung? Wie viel Zeit wird die Beantwortung des Fragebogens beanspruchen? Was geschieht mit den Ergebnissen?

Werden die Teilnehmer über die Ergebnisse informiert? Wenn ja, bis wann? Um die Anzahl der potenziellen Antworten zu steigern, ist ein entsprechender Anreiz möglicherweise hilfreich.

Ergebnisverarbeitung

Die Verarbeitung der Ergebnisse qualitativer Fragebögen wird gewöhnlich eine gewisse Zeit in Anspruch nehmen. Je mehr Antworten Sie erhalten, umso mehr Zeit werden Sie für deren Auswertung benötigen. Dabei werden Sie gewöhnlich mit widersprüchlichen oder potenziell widersprüchlichen Aussagen und unterschiedlichen Ebenen von Anforderungen umgehen müssen.

Hinweis 3.1.3:

Ein Affinitätsdiagramm [BaCC2015] kann bei der Verarbeitung der gesammelten Daten hilfreich sein (sehen Sie sich hierzu das Beispiel eines Affinitätsdiagramms in Kapitel 3.1.1.1 zur Verarbeitung von aus Interviews erhaltenen Daten an).

Bei quantitativen Fragebögen kann die Ergebnisverarbeitung automatisch erfolgen (sofern die Befragung elektronisch durchgeführt wird). Bei Verwendung eines webbasierten Werkzeugs wirkt sich die Anzahl der Antworten nicht auf die für die Aggregation der Daten benötigte Zeit aus. Für die intellektuelle Interpretation der Statistiken kann aber dennoch einige Zeit benötigt werden.

Typische Arbeitsprodukte

Bei der Vorbereitung, Durchführung und Ergebnisverarbeitung einer Befragung anhand von Fragebögen werden üblicherweise die folgenden Arbeitsergebnisse erzeugt oder aktualisiert:

- Fragebogen
- Verarbeitete Ergebnisse (z. B. Statistiken, Diagramme, Berichte)

Möglichkeiten

Eine Befragung anhand von Fragebögen ist eine asynchrone Ermittlungstechnik, das heißt der Requirements Engineer und der Stakeholder müssen nicht zur selben Zeit am selben Ort sein, um diese Technik zu nutzen.

Durch die Verwendung von Fragebögen können viele Stakeholder auf einmal involviert werden; entweder, um neue Anforderungen von zahlreichen Stakeholdern gleichzeitig zu sammeln oder um vorhandene Anforderungen mithilfe einer großen Menge von Stakeholdern auszuwerten und auf diese Weise eine Hypothese zu bestätigen oder zu entkräften (z. B. „Die Benutzer sind mit dem System, so wie es ist, zufrieden.“).

Herausforderungen

Der Einsatz eines Fragebogens kann sehr zeitaufwendig sein: „Eines der größten Missverständnisse in Bezug auf eine Befragung ist die Geschwindigkeit, mit der Sie die Ergebnisse vorbereiten, sammeln und analysieren können. Eine Befragung kann eine äußerst nützliche Methode sein, aber es braucht Zeit, sie korrekt durchzuführen.“ [BaCC2015]

Vor allem bei qualitativen Fragebögen kann die Auswertung extrem zeitaufwendig ausfallen.

Ein wesentlicher Nachteil von Fragebögen ist die begrenzte (oder gar nicht vorhandene!) Möglichkeit von Feedback in Zweifelsfällen. Dies gilt sowohl für die Antwortenden als auch für den Requirements Engineer. Daher könnten Fragen sowie Antworten falsch interpretiert werden.

[BaCC2015] beschreibt außerdem folgende Punkte, die zu beachten sind:

- *Selektionseffekt* (engl. Selection Bias): Häufig werden aus Bequemlichkeit nur Teilnehmer ausgewählt, an die man leicht herankommt (z. B. aus der eigenen Abteilung, Freunde, Familie). Eine solche Gruppe ist allerdings nicht repräsentativ, was zu unpräzisen Daten führen kann.
- *Schweigeverzerrung* (engl. Non-Response Bias): Selbst, wenn die Stakeholder, an welche der Fragebogen verteilt wird, repräsentativ ausgewählt werden, kann ein Missverhältnis auftreten, da einige Teilnehmer antworten, andere aber nicht. Die Rücklaufquote kann zwischen 20 und 60 Prozent liegen.
- *Satisficing*: Wenn der Fragebogen zu viel kognitiven Aufwand erfordert, kann Satisficing angewendet werden, eine Strategie, um zufriedenstellende, aber nicht ideale Ergebnisse zu erzielen. Teilnehmer können beispielsweise für alle Fragen dieselbe Wahl treffen.

Varianten

Abgesehen von den zuvor besprochenen Varianten von qualitativen und quantitativen Fragebögen, gibt es noch die folgenden Varianten:

- **Schriftlich auszufüllender Fragebogen:**
Der Fragebogen wird auf Papier beantwortet. Dies hat einen hohen Verarbeitungsaufwand zur Folge, da die Daten vor der Auswertung computergestützt verarbeitet werden müssen. Für diese Art von Fragebogen steigt der Aufwand für Bewertungen mit der Anzahl der Antworten. Dies gilt für qualitative und quantitative Fragebögen gleichermaßen.
- **Computergestützter/webbasierter Fragebogen:**
Der Fragebogen wird online beantwortet. Bei quantitativen Fragebögen können die Antworten automatisch aggregiert werden.

3.1.2 Beobachtungstechniken

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 stark zu vereinfachen. Daher wird sehr empfohlen, sich vor Anwendung von Beobachtungstechniken Wissen über das Thema anzueignen (z. B. mit einem Fachexperten sprechen), um diese Voreingenommenheit zu minimieren. Es ist auch ratsam, Fachexperten die Beobachtungsnotizen im Anschluss reviewen zu lassen.

In der Praxis können sich die drei Techniken, die wir in diesem Handbuch vorstellen, überschneiden: Feldbeobachtung, Apprenticing und Contextual Inquiry. Daher ist es wichtig, den Übergang von einer Technik in die andere vorzunehmen oder eine bewusste Mischung von Techniken anzuwenden. Es sollte jedoch vermieden werden, sich von einer Technik in eine andere treiben zu lassen, z. B. von einem Contextual Inquiry zu einer reinen Feldbeobachtung oder umgekehrt. Wenn es Gründe für eine Mischung der Techniken gibt, dann tun Sie es, aber beachten Sie dabei, was für die einzelnen Elemente wichtig ist.

3.1.2.1 Feldbeobachtung

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. Bei der Feldbeobachtung, die auch als Job-Shading oder Beobachtung bekannt ist, begleitet man jemanden während einer Tätigkeit.

Was ist Feldbeobachtung?

Bei der Feldbeobachtung beobachtet der Requirements Engineer die Stakeholder (in der Regel Endbenutzer) in ihrer Umgebung, während diese die Aufgaben ausführen, für die ein System entwickelt oder verbessert werden soll. Der wichtigste Unterschied zwischen der Feldbeobachtung und Apprenticing oder Contextual Inquiry ist, dass zwischen dem Beobachter und dem oder den beobachteten Subjekten keine Interaktion stattfindet.

Die Feldbeobachtung wird üblicherweise in Situationen angewendet, in denen eine Interaktion mit Benutzern nicht möglich ist (z. B. wenn es ablenken würde) oder wenn dies den eigentlichen Prozess stören und die Ergebnisse potenziell falsifizieren würde. Sie kann auch an öffentlichen Orten angewendet werden, ohne dass die beobachteten Subjekte darüber informiert werden (z. B. mit anderen Patienten im Wartezimmer einer Arztpraxis sitzen und diese während der Wartezeit beobachten).

Die Rolle der Teilnehmer

Der Requirements Engineer übernimmt die Rolle des Beobachters. Der Stakeholder ist das *beobachtete Subjekt*. Es kann mehr als einen Beobachter und auch mehr als ein beobachtetes Subjekt geben.

Vorbereitung

Definieren Sie das oder die Ermittlungsziele und die erforderliche Ergebnisqualität, wenn Sie die Ermittlungsaktivität definieren, und wählen Sie für die Beobachtung einen oder mehrere geeignete Stakeholder aus.

Hier ist das Ermittlungsziel von besonderer Bedeutung, da es Ihnen hilft, sich während der Beobachtung auf die relevanten Aspekte zu konzentrieren. Da unser Gehirn nicht dazu in der Lage ist, die ganze Zeit über alles zu beobachten, ist es äußerst wichtig, genau zu wissen wonach man sucht.

Bei der Feldbeobachtung haben Zeit und Ort einen wesentlichen Einfluss auf das Ergebnis. Unter Umständen sind weitere Nachforschungen notwendig, um die richtige Zeit und den passenden Ort herauszufinden.

Für Zeit und Ort kann es etwa folgende Einflussfaktoren geben:

- Der zu beobachtende Prozess findet nur zu bestimmten Zeiten statt (z. B. Menschen, die ihren Arbeitstag beginnen; Essensausgabe in einer Kantine).
- Der Prozess findet an verschiedenen Orten und zu unterschiedlichen Zeiten statt (z. B. Menschen, die öffentliche Verkehrsmittel nutzen und ein- oder aussteigen).
- Teile des zu beobachtenden Gesamtprozesses finden zu verschiedenen Zeiten und/oder an unterschiedlichen Orten statt (z. B. ein Teil wird in einer Produktionslinie hergestellt und dann an eine andere Produktionslinie übergeben, wo es nicht umgehend bearbeitet wird).

Entscheiden Sie im Voraus, ob die beobachteten Subjekte darüber informiert werden, dass man sie beobachtet. In allen Situationen, in denen es offensichtlich ist, dass sie als Beobachter dabei sind (z. B. wenn Sie jemanden beobachten, der ansonsten gewöhnlich alleine in einem Raum arbeitet), sollten Sie die beobachteten Subjekte im Voraus informieren.

In anderen Fällen ist es unter Umständen nicht einmal möglich, die beobachteten Subjekte zu informieren (z. B. bei einer Beobachtung an einem öffentlichen Ort wie in einem Supermarkt).

Anwendung

Wird das beobachtete Subjekt über die Beobachtung informiert, ist es eine gute Idee, kurz zu erklären, weshalb Sie dies tun und dass die Person Ihre Anwesenheit möglichst ausblenden und ihre Arbeit wie gewohnt durchführen soll.

Bei der Feldbeobachtung bezieht der Beobachter alle Sinne mit ein (Sehen, Hören, Riechen, Fühlen), um Informationen über die beobachtete Situation/die beobachteten Prozesse zu sammeln, die in Bezug auf das Ermittlungsziel relevant sein können. Je nach Situation ist es vielleicht möglich, Fotos und/oder Videos aufzunehmen.

Wie im Abschnitt über die Vorbereitung bereits erwähnt, ist es sehr wichtig, dass Ihnen Ihr Ermittlungsziel während der Beobachtung präsent ist. Es besteht aber auch das Risiko, dass man sich zu sehr darauf konzentriert und daher während der Beobachtung voreingenommen ist. Bildlich gesprochen ist das Ermittlungsziel Ihr Leuchtturm: Es hilft Ihnen dabei, durch die Beobachtung zu navigieren und zu entscheiden, ob etwas für das Erreichen des Ermittlungsziels relevant ist.

Beteiligen Sie sich nicht an den Situationen unter Beobachtung, und greifen Sie nicht ein.

In manchen Fällen ist es möglich, die Feldbeobachtung per Kamera durchzuführen (z. B. mithilfe zuvor installierter Sicherheitskameras) und dadurch ein Eingreifen in den Prozess durch Ihre physische Präsenz zu vermeiden.

[BaCC2015] schlägt vor, bei der Feldbeobachtung folgende Aspekte zu berücksichtigen:

- Welche Sprache und Terminologie nutzen die Personen?
- Wenn Sie die Nutzung eines vorhandenen Systems beobachten: Wie viel von diesem System/dieser Software/den Features nutzen die Anwender tatsächlich?
- Auf welche Hindernisse oder Haltepunkte stoßen die Nutzer?
- Wenn das Objekt Ihres Interesses aufgabenorientiert ist:
 - Wie viel Zeit widmen die Nutzer der Erledigung einer Aufgabe?
 - Welche Fragen müssen die Nutzer stellen, um eine Aufgabe zu bewerkstelligen?
 - Mit welchen Werkzeugen interagieren Benutzer, wenn sie versuchen, eine Aufgabe durchzuführen?

Ergebnisverarbeitung

Anmerkung: Die Ergebnisverarbeitung für Feldbeobachtung und Contextual Inquiry ist im Allgemeinen identisch.

Jede Feldbeobachtungssitzung führt zu einer großen Menge an gesammelten Rohdaten. Diese müssen analysiert und aggregiert werden. Für die Ergebnisverarbeitung einer Feldbeobachtung könnten folgende unterstützende Techniken genutzt werden:

- Die Analyse des Benutzer–Aufgabe–System–Kontext–Problems, d. h. die Anwendung von perspektivenbasiertem Lesen, um die Informationen zu diesen fünf Aspekten aus

den gesammelten Rohdaten zu extrahieren. Das Ergebnis ist ein intermediäres Arbeitsergebnis, auf dessen Basis die nachfolgenden Analysen vorgenommen werden können. Diese Analysen sollten im Team, bevorzugt unter Beteiligung von Stakeholdern und technischen Teammitgliedern durchgeführt werden.

- Alternativ kann das gesammelte Material mit einem Affinitätsdiagramm verarbeitet werden. [BaCC2015]
- Anhand der bei der Feldbeobachtung gesammelten Rohdaten oder basierend auf den intermediären Arbeitsergebnissen, die gemäß der beiden vorherigen Punkte erstellt wurden, können Sie Folgendes identifizieren und dokumentieren:
 - reale Benutzergruppen oder Personas
 - Prozesse und Reibungspunkte innerhalb dieser Prozesse
 - kulturelles Modell, physisches Modell und Flussmodell [BeHo1998]
 - Liste mit wichtigen Arbeitsergebnissen, die bei der Fertigstellung der Aufgabe verwendet wurden
 - Liste der Probleme
 - Ziele, Probleme, Bedürfnisse und Anforderungen

Typische Arbeitsprodukte

Bei der Vorbereitung, Durchführung und Nachbereitung einer Feldbeobachtungsaktivität werden üblicherweise die folgenden Arbeitsergebnisse erzeugt oder aktualisiert:

- Hinweise von der Beobachtung (z. B. Texte, Zeichnungen)
- Arbeitsergebnisse, die während der Beobachtung gesammelt wurden (z. B. aufgenommene Bilder oder Videos)
- Verarbeitete Ergebnisse, z. B. wie von [BeHo1998] vorgeschlagen:
 - Skizzen von interessanten sachlichen Zusammenhängen
 - physische Modelle, die den Kontext der Arbeit darstellen
 - ein kulturelles Modell, das die Zusammenarbeit zwischen den teilnehmenden Benutzern verdeutlicht
 - ein Flussmodell, das Prozessunterbrechungen aufzeigt

Möglichkeiten

Die Feldbeobachtung ist eine sehr nützliche Technik, wenn die Interaktion mit Benutzern nicht möglich oder beabsichtigt ist [BaCC2015]. Sie hilft dabei, Informationen zu sammeln, die von den Stakeholdern nicht verbal ausgedrückt werden können (z. B. implizites Wissen), und sie hilft dabei, die Situation von Stakeholdern zu verstehen und dadurch in Folgeinteraktionen empathischer zu sein.

Die Feldbeobachtung hat den großen Vorteil, dass keine zusätzlichen Stakeholder-Ressourcen erforderlich sind: Die Stakeholder führen ihre Aufgaben einfach wie gewohnt aus.

Nach [Koelsch2016] können Feldbeobachtungen auch zu folgendem beitragen:

- Arbeits- oder Prozessflüsse zu identifizieren
- Aspekte zu identifizieren, die Nutzer stören

- umständliche Schritte festzustellen, denen sie begegnen
- Raum für Verbesserungen zu identifizieren

Herausforderungen

Die Voreingenommenheit des Beobachters (Observer's Bias) beeinflusst, was er oder sie tatsächlich wahrnimmt. Ein Beobachter kann davon überzeugt sein, zu verstehen, worum es bei einer bestimmten Maßnahme geht, aber er kann damit völlig falsch liegen. Die Feldbeobachtung sollte niemals für sich allein genutzt werden; es sollten immer zusätzliche Ermittlungstechniken (z. B. Interviews) daran anschließen.

Abhängig davon, was und wo etwas beobachtet wird, kann die Feldbeobachtung sehr zeitaufwendig sein, vor allem wenn lange Perioden auftreten, in denen nichts geschieht.

Varianten

[BaCC2015] beschreiben das intensive Verbringen von Zeit mit etwas („deep hanging out“) als eine strukturiertere Form der Beobachtung. Der Beobachter wird selbst zu einem Benutzer (z. B. durch die Nutzung öffentlicher Verkehrsmittel), und es ist eine formale Struktur nötig, um den Beobachtungsprozess zu organisieren, beispielsweise durch die Konzentration auf diese zehn Schwerpunkte: Familie und Kinder, Essen und Trinken, Gebäudeumgebung, Besitz, Medienkonsum, Werkzeuge und Technologie, demografische Daten, Verkehr, Zugang zu Informationen und Kommunikation sowie die Gesamterfahrung.

3.1.2.2 Apprenticing

Wir nehmen hier eine strikte Trennung von Feldbeobachtung und Apprenticing vor: Die Feldbeobachtung erfolgt ohne Beobachter, das Apprenticing hingegen unter Beteiligung des Beobachters. In der Literatur finden sich unterschiedliche Definitionen, wonach beide Konzepte, oder sogar alle drei Konzepte, einschließlich Contextual Inquiry, als Apprenticing bezeichnet werden (siehe [RoRo2013]).

Was ist Apprenticing?

Apprenticing folgt der Idee eines Meisters (engl. master) und eines Auszubildenden (engl. apprentice) [RoRo2013]. Der Requirements Engineer macht ein kurzes Praktikum in dem Umfeld, in dem das zu entwickelnde/zu verbessernde System später eingesetzt wird (oder bereits im Einsatz ist).

Ein erfahrener Experte („Meister“) leitet dabei den Requirements Engineer („Auszubildenden“) an, um ihn zu befähigen, das Fachgebiet besser zu verstehen und dadurch Anforderungen besser zu ermitteln. „Um welche Arbeit es auch immer geht, scheint es vernünftig zu sein, wenn man ein angemessenes Verständnis davon hat, bevor man versucht, etwas daran zu ändern.“ [RoRo2013].

Die optimale Dauer des Praktikums hängt von vielen verschiedenen Faktoren ab, z. B. von der Komplexität des Prozesses, einer hohen gegenüber einer geringen Wiederholungshäufigkeit und der verfügbaren Zeit von „Meister“ und „Auszubildendem“.

Ein wichtiger Unterschied zwischen Apprenticing und Contextual Inquiry ist, dass der Requirements Engineer beim Apprenticing die untersuchte Arbeit tatsächlich ausführt und etwas über den Kontext und die Aufgaben lernt, und zwar nicht nur durch Beobachten und Fragenstellen, sondern in erster Linie durch das tatsächliche Ausüben der Arbeit – was naturgemäß mehr Zeit in Anspruch nimmt als die Durchführung eines Contextual Inquiry. Bei Letzterem geht es lediglich um die Beobachtung des Experten, während dieser eine Aufgabe in seinem Kontext ausführt, sowie um die verbale kognitive Aufgabenanalyse gemeinsam mit dem Experten (keine praktische Ausübung durch den Requirements Engineer).

Die Rolle der Teilnehmer

Der Requirements Engineer wird zum „Auszubildenden“. Seine Aufgabe ist es, vom Stakeholder, der zum „Meister“ wird, zu lernen.

Vorbereitung

Definieren Sie das oder die Ermittlungsziele und die erforderliche Ergebnisqualität, wenn Sie die Ermittlungsaktivität definieren, und wählen Sie für einen oder mehrere geeignete Stakeholder als Meister für das Apprenticing aus (siehe unten).

Wie bei der Feldbeobachtung ist das Ermittlungsziel für das Apprenticing von besonderer Bedeutung, da es Ihnen hilft, sich während des Apprenticing auf die relevanten Aspekte zu konzentrieren.

Der Requirements Engineer sollte alle Informationen im Voraus sammeln, die ihm dabei helfen könnten, in der Apprenticing-Situation zu einem besseren Schüler zu werden, wodurch das Stellen grundlegender Fragen während des Apprenticing vermieden wird. Auf diese Weise wird die vom Stakeholder aufgewendete Zeit effizient genutzt.

Aus organisatorischer Perspektive muss ein geeigneter „Meister“ gefunden werden, der bereit ist, sich am Apprenticing zu beteiligen. Hat der „Meister“ keine Motivation, das Apprenticing durchzuführen, dann ist es empfehlenswert zu einer anderen Ermittlungstechnik zu wechseln. Der „Meister“ spielt bei dieser Technik eine aktive und grundlegende Rolle. Wir empfehlen ein Vorbereitungsgespräch mit dem „Meister“, um zu klären, wie und wann das Apprenticing stattfinden soll.

Es wäre gut, den „Meister“ dahingehend zu instruieren, dass er vorsichtig wird, wenn der „Auszubildende“ zu lange schweigt. Dann sollte er beispielsweise folgende Fragen stellen: „Was denken Sie darüber?“ oder „Erzählen Sie mir, was in Ihnen vorgeht.“

Anwendung

Der „Meister“ bringt dem „Auszubildenden“ die relevanten Prozesse und/oder Maßnahmen bei. Das Hauptziel des Apprenticing ist es, den Kontext zu erleben, in dem die Lösung später angewendet werden soll. Daher sollte der Requirements Engineer zumindest einige der Schritte des Gesamtprozesses selbst durchführen. Der Requirements Engineer sollte es vermeiden, Annahmen zu treffen. Er sollte stattdessen alle Schlussfolgerungen zum Ausdruck bringen, damit der „Meister“ diese bestätigen oder korrigieren kann.

Es ist wichtig, während des Apprenticing viele Fragen zu stellen, um so viele Informationen wie möglich zu erhalten und falsche Annahmen zu vermeiden.

Der „Auszubildende“ macht sich während des Apprenticing Notizen.

Der Auszubildende kann manche Aufgabe(n) möglicherweise bereits selbst durchführen, wenn der Meister dies für geeignet hält.

Ergebnisverarbeitung

Unmittelbar nach dem Apprenticing sollte der Requirements Engineer so viele Aspekte aus dem Apprenticing wie nur möglich schriftlich erfassen, um das erworbene Wissen zu bewahren.

Anschließend werden die im Apprenticing gewonnenen Daten analysiert.

Die folgenden Fragen können als Orientierung verwendet werden:

- Auf welche Schwierigkeiten bin ich gestoßen?
- Was könnte man einfacher machen?
- Was empfand ich als kompliziert?
- Was wäre der größte Vorteil?

Typische Arbeitsprodukte

Bei der Vorbereitung, Durchführung und Nachbereitung einer Apprenticing-Aktivität werden üblicherweise die folgenden Arbeitsergebnisse erzeugt oder aktualisiert:

- Notizen aus dem Apprenticing (währenddessen und im Anschluss an das Apprenticing erfasst)
- Arbeitsergebnisse, die während des Apprenticing gesammelt wurden (z. B. aufgenommene Screenshots, Schulungsmaterial)
- verarbeitete Ergebnisse (sollten vom „Meister“ überprüft werden!)

Möglichkeiten

Apprenticing hilft dabei, die Umgebung des Systems und den Anwendungsbereich zu verstehen. Der Requirements Engineer erlebt die Schwierigkeiten und angewendeten Workarounds, anstatt lediglich etwas über sie zu erfahren. Die Interaktion mit Stakeholdern während des Apprenticing verbessert gewöhnlich die Beziehung und dadurch später auch die Kommunikation mit ihnen. Da der Requirements Engineer die Arbeitsumgebung des Stakeholders kennengelernt hat, wird er möglicherweise eher als Teil der Gruppe und weniger als Außenstehender betrachtet. Dies hilft dem Requirements Engineer außerdem dabei, die Situation von Stakeholdern zu verstehen und dadurch in Folgeinteraktionen empathischer zu sein.

Durch das Apprenticing kann implizites Wissen erhoben und in explizites Wissen verwandelt werden.

Herausforderungen

Die Qualität des Apprenticing steigt und fällt mit der Motivation und den didaktischen Fähigkeiten der Stakeholder. Daher ist es wichtig, diese Aspekte vor dem Apprenticing zu klären. Vom Stakeholder ist eine beträchtliche Investition gefordert: Während des Apprenticing kann er seiner Arbeit nicht nachgehen oder zumindest nicht in dem sonst üblichen Tempo.

Varianten

Wenn der Requirements Engineer die Aufgaben nicht aktiv ausführt sondern den Stakeholder nur für einen längeren Zeitraum begleitet, Fragen stellt und Dinge vom Stakeholder erklärt bekommt, wird dies gewöhnlich als „Job-Shadowing“ bezeichnet.

Hinweis:

Job Shadowing für einen kurzen Zeitraum ist eher eine Variante des Contextual Inquiry.

3.1.2.3 Contextual Inquiry

Was ist Contextual Inquiry?

Contextual Inquiry ist eine iterative Technik, bei der Felddaten gesammelt werden und der Requirements Engineer einige mit Bedacht ausgesuchte Anwender in der Tiefe studiert, um ein besseres Verständnis der Arbeitsvorgänge der gesamten Anwenderbasis zu erlangen [BeHo1998]. Es handelt sich dabei um eine umfassende Kombination aus Beobachtungen und Diskussionen. Der Requirements Engineer (und andere begleitende Teammitglieder) beobachten die Benutzer in ihrem eigenen Arbeitskontext. Darüber hinaus diskutiert der Requirements Engineer mit den Benutzern ihre Aufgaben und befasst sich mit ihnen, um unausgesprochene Aspekte ihrer Arbeit aufzudecken. Ziel ist es, Strukturen und Muster zu erkennen und herauszufinden, wie der Benutzer seine Arbeit organisiert.

Contextual Inquiry basiert auf vier Prinzipien:

- *Kontext:* Begeben Sie sich in den Kontext des Anwenders, um ihn bei der Durchführung seiner Aufgaben zu beobachten.
- *Partnerschaft:* Versuchen Sie, eine partnerschaftliche Beziehung aufzubauen: Versuchen Sie gemeinsam, die Arbeitsvorgänge des Benutzers nachzuvollziehen.
- *Interpretation:* Entwickeln Sie ein gemeinsames Verständnis mit dem Anwender zu den relevanten Aspekten der Tätigkeit.
- *Fokus:* Definieren Sie bei der Vorbereitung des Contextual Inquiry Ermittlungsziele und richten Sie Ihre Ermittlung darauf aus, die relevanten Daten zu erheben, um die Ziele zu erreichen.

Die Hauptunterschiede zwischen Apprenticing und Contextual Inquiry bestehen darin, dass das Apprenticing in der Regel wesentlich mehr Zeit erfordert (gewöhnlich 1 bis 3 Tage) und der Requirements Engineer in der Rolle des „Auszubildenden“ tatsächlich die relevanten Aufgaben ausführt. Eine Contextual-Inquiry-Sitzung dauert hingegen 60 bis 90 Minuten und

der Requirements Engineer beobachtet und spricht lediglich mit dem Stakeholder, der die Aufgabe ausführt.

Die Rolle der Teilnehmer

Der Anwender ist Experte des Fachgebiets. Der Requirements Engineer ist der Experte für das Erkennen von Arbeitsstrukturen, Mustern und Unterschieden in der Arbeitsorganisation von Einzelpersonen. In dieser Inquiry-Sitzung sind die beiden Partner, die versuchen, gemeinsam die relevanten Aspekte der Arbeit des Benutzers aufzudecken.

Andere Teilnehmer übernehmen die Rolle des Beobachters. Jeder von ihnen kann sich auf unterschiedliche Aspekte konzentrieren, z. B. verwendete Arbeitsergebnisse, den physischen Kontext, Kommunikation, Prozess, Kultur usw.

Vorbereitung

Die Vorbereitung eines Contextual Inquiry hängt von der Zielgruppe ab. Wenn der Zugang zu Benutzern begrenzt oder womöglich kostenintensiv ist (z. B. Börsenhändler, Topmanager, Anwender im Ausland usw.), kann der Aufwand für die Vorbereitung um ein Vielfaches höher sein als für die tatsächliche Ausführung der Inquiry. In solchen Fällen muss das Verständnis des Requirements Engineers davon, was er erfahren möchte, im Voraus aufgebaut werden (z. B. durch Interviews mit einer Person der Zielgruppe, die bereits in Rente ist; indem man zuarbeitendem Personal im Kontext Fragen stellt; durch die Teilnahme an Anwenderschulungen; durch das Ansehen von Videos und Studieren von Tutorials oder schriftlichen Unterlagen). Ist die Zielgruppe leicht zugänglich, kann das grundlegende Verständnis durch ein einziges Interview mit jemandem gewonnen werden, der die Benutzer kennt. Beim ersten Contextual Inquiry kann man sich dann darauf konzentrieren, die Aufgaben und den Kontext des Benutzers wirklich kennenzulernen.

Das sind die allgemeinen Aufgaben zur Vorbereitung eines Contextual Inquiry:

- Legen Sie in Abhängigkeit vom Ermittlungsziel die Anzahl, die Art und den Ort der Anwender fest, die besucht werden sollen.
- Vereinbaren Sie mit den ausgewählten Benutzern Termine und bitten Sie sie, typische – oder spezielle – Aufgaben vorzubereiten, an denen Sie während des Contextual Inquiry arbeiten.
- Wenn Sie eine Reihe von Contextual Inquiries planen, bereiten Sie die einzelnen Inquiry-Sitzungen iterativ vor:
 - Wer nimmt daran teil (involvieren Sie nach Möglichkeit Stakeholder und/oder andere Teammitglieder, damit diese auch Anwender in diesem Gebiet erleben)?
 - In Abhängigkeit vom Ermittlungsziel: Worauf soll der Schwerpunkt der Inquiry gelegt werden? Wer konzentriert sich auf welchen Aspekt?
 - Wie sollen die Erkenntnisse aufgezeichnet werden (bereiten Sie Checklisten für Ihre Notizen vor; prüfen Sie, ob Audio-, Video oder Fotoaufnahmen erlaubt sind usw.)?

Anwendung

Nachdem Sie vor Ort eingetroffen sind, stellen Sie die einzelnen Teilnehmer und Ihr Ermittlungsziel vor. Erklären Sie das partnerschaftliche Setting: Der Benutzer ist der Experte für die vorliegende Aufgabe; der Requirements Engineer ist der Experte für das Erkennen von Arbeitsstrukturen, Mustern und Unterschieden in der Arbeitsorganisation der Mitarbeiter. Bitten Sie anschließend den Anwender, die vorbereiteten Aufgaben auszuführen und zu erläutern, was er tut und die Gründe dafür. Wenn Sie die Erlaubnis haben, vergessen Sie nicht, die Aufzeichnung zu starten.

Beobachten Sie den Anwender während des Contextual Inquiry, machen Sie Fotos (wenn zulässig), sammeln Sie Arbeitsergebnisse, mit denen der Benutzer arbeitet, machen Sie sich Notizen, und stellen Sie dem Benutzer regelmäßig Fragen über Aspekte seiner Vorgehensweise bei den Aufgaben. So ist der Benutzer gezwungen, Dinge explizit zu machen, was dem Requirements Engineer ermöglicht, die Haltungen, Ziele, Probleme und Bedürfnisse des Benutzers kennenzulernen. Darüber hinaus erhält das Team Informationen zu den echten Prozessen und Reibungspunkten im Arbeitsablauf, zu kulturellen Aspekten des Benutzers und seiner Kollegen, zum Informationsfluss, der physischen Umgebung usw.

Nach dem Contextual Inquiry führen Requirements Engineers manchmal ein Interview mit dem Benutzer durch, um eine Liste von vorbereiteten Fragen durchzugehen. Dies wäre jedoch die Hinzufügung einer anderen Ermittlungstechnik. Schließen Sie das Contextual Inquiry mit einer Nachbesprechung ab.

Ergebnisverarbeitung

Anmerkung: Die Ergebnisverarbeitung für Feldbeobachtung und Contextual Inquiry ist im Allgemeinen ähnlich.

Jede Contextual-Inquiry-Sitzung führt zu einer großen Menge an gesammelten Rohdaten. Diese müssen analysiert und aggregiert werden. Für die Ergebnisverarbeitung eines Contextual Inquiry könnten folgende unterstützende Techniken genutzt werden:

- Die Analyse des Benutzer-Aufgabe-System-Kontext-Problems, d. h. die Anwendung von perspektivenbasiertem Lesen, um die Informationen zu diesen fünf Aspekten aus den gesammelten Rohdaten zu extrahieren. Das Ergebnis ist ein intermediäres Arbeitsergebnis, auf dessen Basis die nachfolgenden Analysen vorgenommen werden können. Diese Analysen sollten im Team, bevorzugt unter Beteiligung von Stakeholdern und technischen Teammitgliedern durchgeführt werden.
- Alternativ kann das gesammelte Material in einem Affinitätsdiagramm verarbeitet werden. [BaCC2015]
- Anhand der bei der Inquiry-Sitzung gesammelten Rohdaten oder basierend auf den intermediären Arbeitsergebnissen, die gemäß der beiden vorherigen Punkte erstellt wurden, können Sie Folgendes identifizieren und dokumentieren:
 - reale Benutzergruppen oder Personas
 - Prozesse und Reibungspunkte innerhalb dieser Prozesse
 - kulturelles Modell, physisches Modell und Flussmodell [BeHo1998]

- Liste mit wichtigen Arbeitsergebnissen, die bei der Fertigstellung der Aufgabe verwendet wurden
- Liste der Probleme
- Ziele, Probleme, Bedürfnisse und Anforderungen

Wiederholen Sie diese Schritte für jedes Contextual Inquiry, das Sie durchführen, und erweitern Sie so systematisch Ihr Verständnis der vorgenannten Aspekte (Benutzer, Aufgaben, Systeme, Kontext und Probleme).

Typische Arbeitsprodukte

Bei der Vorbereitung, Durchführung und Nachbereitung eines Contextual Inquiry werden üblicherweise die folgenden Arbeitsergebnisse erzeugt oder aktualisiert:

- Notizen und Audioaufzeichnungen des Contextual Inquiry
- Arbeitsergebnisse, die während des Contextual Inquiry gesammelt wurden (z. B. aufgenommene Bilder oder Videos, ausgefüllte Formulare, Notizen, Listen, Werkzeuge usw.)
- Verarbeitete Ergebnisse, z. B. wie von [BeHo1998] vorgeschlagen:
 - Skizzen von interessanten sachlichen Zusammenhängen
 - physische Modelle, die den Kontext der Arbeit darstellen
 - ein kulturelles Modell, das die Zusammenarbeit zwischen den teilnehmenden Benutzern verdeutlicht
 - ein Flussmodell, das Prozessunterbrechungen aufzeigt

Möglichkeiten

Eine Herausforderung bei Feldbeobachtungen besteht darin, Fehlinterpretationen des beobachteten Benutzerverhaltens zu vermeiden. Beim Contextual Inquiry ist es hingegen ein zentrales Merkmal, solche Beobachtungen sofort im Kontext aufzugreifen und zu verifizieren. Darüber hinaus müssen Benutzer nicht darüber nachdenken, wie sie ihre Arbeitsweise bei einer Präsentation in einem Meetingraum angemessen erklären. Sie gehen stattdessen einfach ihrer Arbeit nach (Aufgaben, die sie gewöhnlich jeden Tag ausführen) und erklären dem Requirements Engineer, was sie gerade tun. Dadurch reflektieren sie ihren eigenen Handlungsablauf und machen ihr Expertenwissen explizit. Der Requirements Engineer und der Benutzer diskutieren auf Basis einer gerade ausgeführten Arbeitsaufgabe Probleme, sachliche Beziehungen, Motivationen und Verbesserungsmöglichkeiten.

Das Contextual Inquiry ist die Königsdisziplin der personenzentrierten Anforderungsermittlung [RiFi2014]. Der Requirements Engineer kann so in einem kurzen Zeitraum effektiv umfangreiche Informationen über die Benutzer, deren Ziele und Aufgaben, die verwendeten Systeme, den relevanten Kontext und echte Probleme herausfinden.

Herausforderungen

Abhängig von Faktoren wie dem Fachwissen des Requirements Engineers, der Verfügbarkeit geeigneter Anwender für die Ausführung des Contextual Inquiry, dem geografischen Standort, an dem sich der zu beobachtende Kontext befindet, und der Anzahl der

potenziellen Benutzergruppen, kann die Anzahl der zu planenden Contextual Inquiries hoch sein und der Durchführungsaufwand beträchtlich.

Gehen Sie sicher, dass Sie die korrekten Benutzer auswählen. Da es sich beim Contextual Inquiry um eine qualitative Ermittlungstechnik handelt, muss die Verteilung der ausgewählten Benutzer nicht zwingend repräsentativ in einem statistischen Sinne sein. Dennoch sollten sie die erwartete Benutzergruppe repräsentieren.

Wenngleich der Requirements Engineer das Contextual Inquiry steuern und die Benutzer zur Teilnahme motivieren kann, ist das Contextual Inquiry nur so lange möglich, wie der Benutzer sich wirklich daran beteiligt. Mit einem unmotivierten Benutzer wird das Contextual Inquiry keine zufriedenstellenden Ergebnisse erbringen.

Das Contextual Inquiry ist nicht nur eine einfache Kombination aus Beobachtung und Interview. Die Vorbereitung und Implementierung eines Contextual Inquiry ist völlig anders und erfordert geeignetes Training und Erfahrung.

Varianten

Das kurze ethnografische Interview [DeDe2011] wird als „Top-down“-Ansatz bezeichnet – im Gegensatz zum „Bottom-up“-Ansatz von Contextual Inquiries –, da man hierbei mit einem teilweise strukturierten Interview beginnt, an das anschließend eine Beobachtung dessen folgt, wie Benutzer ihre Aufgaben im realen Kontext erledigen, mit Schwerpunkt auf Prozessen und Werkzeugen. Auch Arbeitsergebnisse werden gesammelt und diskutiert.

3.1.3 Kollaborationstechniken

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 und Crowd-based Requirements Engineering.

3.1.3.1 Anforderungsworkshops

Workshop ist ein Überbegriff für gruppenorientierte Techniken. Sie können auf verschiedene Weise durchgeführt werden und 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 Dutzend oder Hundert Stakeholdern.

Was ist ein Anforderungsworkshop?

In diesem Handbuch konzentrieren wir uns auf den Anforderungsworkshop nach der Definition von [Gottesdiener2002]:

„Ein Anforderungsworkshop ist ein strukturiertes Meeting, in dem eine sorgfältig ausgewählte Gruppe von Stakeholdern und Sachverständigen zur Definition, Erstellung und Verfeinerung der Arbeitsergebnisse (z. B. Modelle und Dokumente), welche die Benutzeranforderungen darstellen, zusammenarbeitet und diesbezüglich zu einem Abschluss kommt.“

Ein Anforderungsworkshop unterscheidet sich im Wesentlichen in zweierlei Hinsicht von allen anderen Meetings:

- Es handelt sich um ein speziell vorbereitetes Meeting (d. h. es wird geplant, strukturiert und moderiert, engl. *facilitated meeting*).
- Es dient schwerpunktmäßig der Ermittlung⁶ von Anforderungen.

Die Rolle der Teilnehmer

In einem Anforderungsworkshop gibt es vier Rollen, die zu unterscheiden sind: *Moderator*, *Teilnehmer*, *Protokollant* und *Workshop-Sponsor*.

Der Requirements Engineer übernimmt die Rolle des *Moderators*. Es liegt in seiner Verantwortung, den Anforderungsworkshop vorzubereiten (siehe Abschnitt Vorbereitung) und zu moderieren (siehe Abschnitt Anwendung), um sicherzustellen, dass alle relevanten Informationen aus dem Workshop aufgezeichnet werden (siehe Abschnitt Anwendung) und die Anforderungen aus den im Anforderungsworkshop gesammelten Informationen abzuleiten (siehe Abschnitt Ergebnisverarbeitung).

Der Moderator sollte dem Ergebnis gegenüber neutral sein, und er muss kein Sachverständiger sein. Es kann sogar hinderlich sein, wenn der Moderator auch fachlicher Experte ist, da dann das Risiko besteht, dass er ein (verborgenes) Interesse am Ergebnis hat.

Die *Teilnehmer* sind Stakeholder. Ihre Aufgabe ist es, basierend auf dem Ziel des Workshops und unter Anleitung des Moderators, die Workshop-Produkte zu erstellen.

Der *Protokollant* zeichnet die Arbeit der Gruppe auf. Dies kann schriftlich und/oder durch die Aufnahme von Bildern oder Videos erfolgen. In Abhängigkeit von der Komplexität des Anforderungswshops kann diese Rolle auch gemeinsam vom Requirements Engineer und dem Moderator übernommen werden. Der Protokollant sollte auch kein Sachverständiger sein. Obgleich er natürlich in ausreichendem Maß etwas von dem behandelten Thema verstehen muss, sodass er in der Lage ist, die Informationen korrekt zu erfassen.

Der *Workshop-Sponsor* autorisiert und legitimiert den Workshop sowie die Ziele und die Ergebnisqualität des Workshops. Er ist dafür verantwortlich, dass alle benötigten Ressourcen (z. B. Personen, Budget) verfügbar sind.

⁶ Solche Workshops decken häufig auch Aspekte der Konsolidierung ab. Dies ist jedoch nicht der Fokus dieses Kapitels.

Vorbereitung

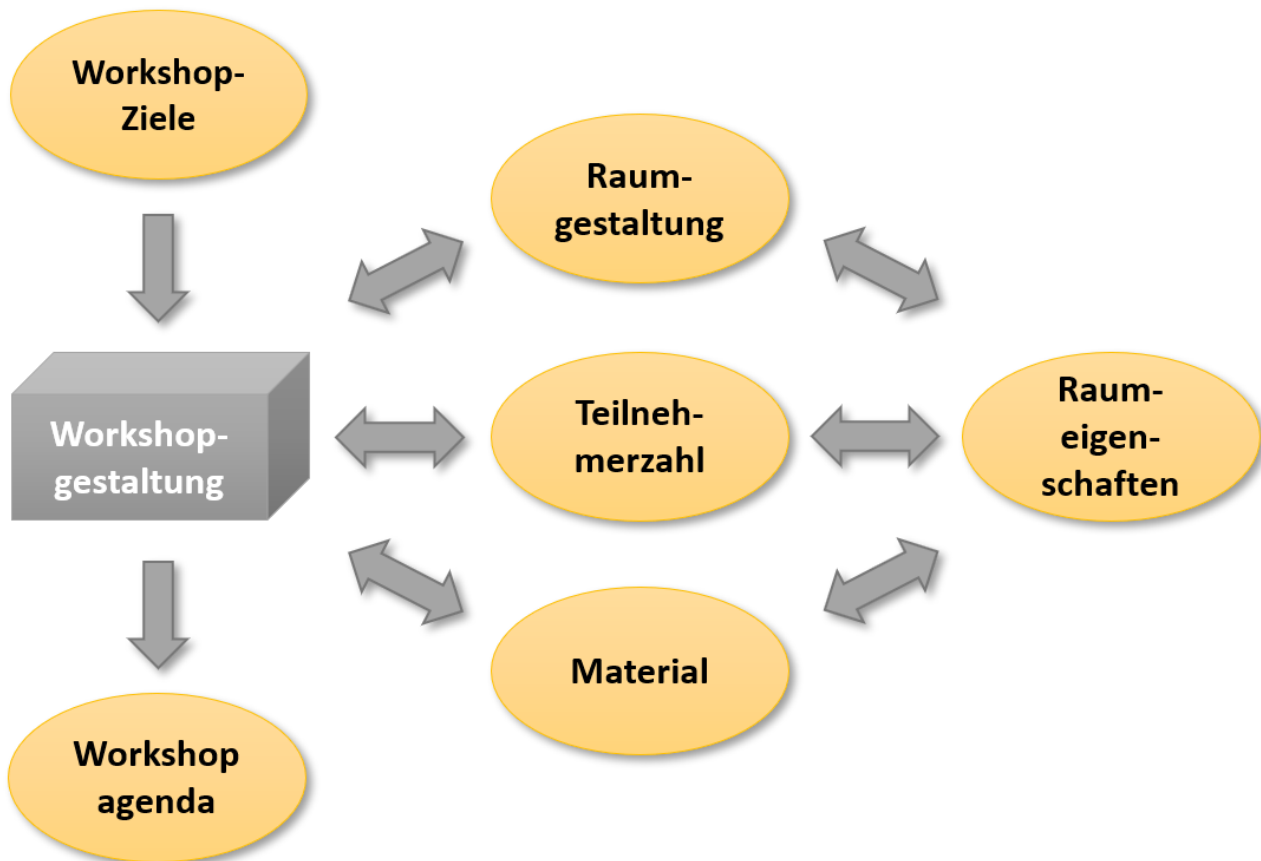


Abbildung 19: Typische wechselseitige Abhängigkeiten von vorbereitenden Tätigkeiten für Workshops

Ein *Workshop-Design* beeinflusst verschiedene Faktoren und wird durch sie beeinflusst (siehe Abbildung 19). Da einige dieser Faktoren sowohl beeinflussen als auch beeinflusst werden, ist das Erstellen eines *Workshop-Designs* ein iterativer Prozess.

Definieren Sie die *Workshop-Ziele* (Ermittlungsziele) und die erforderliche Ergebnisqualität im Rahmen der Definition der Ermittlungsaktivität. Beide Themen müssen mit dem *Workshop-Sponsor* geklärt werden.

Die Einrichtung des Raums hängt vom *Workshop-Design* ab, umgekehrt hängt aber auch das *Workshop-Design* von der *Einrichtung des Raums* ab. Die Einrichtung des Raums und die *Eigenschaften des Raums* beeinflussen sich hingegen gegenseitig.

Im Idealfall wäre das *Workshop-Design* unabhängig von den Eigenschaften des Raums. In den meisten Fällen wird der Raum jedoch entweder im Voraus ausgewählt oder die Auswahl des Raums wird durch die Entfernung und/oder das Budget eingeschränkt. Daher muss sich die *Workshop-Vorbereitung* in den meisten Fällen an den Eigenschaften des Raums orientieren, was uns zur Vorbereitung des Raums führt: Welche Materialien (z. B. Flipcharts, Pinnwände, Whiteboards, Poster) werden wo im Raum platziert? Werden Tische benötigt? Wie sollten diese aufgestellt werden? Werden Stühle benötigt? Wie sollten diese aufgestellt werden? Wo werden die Ergebnisse des Workshops erfasst?

Die Eigenschaften des Raums und die *Anzahl der Teilnehmer* sind ebenfalls wechselseitig voneinander abhängig: je mehr Teilnehmer, desto größer der Raum oder je kleiner der Raum, desto weniger Teilnehmer. Das Workshop-Design ist auch von der Anzahl der Teilnehmer abhängig. Einige Workshop-Modelle sind nicht im gleichen Maß skalierbar wie andere. Die ideale Anzahl von Teilnehmern für einen Anforderungsworkshop beträgt zwischen sieben und zwölf Teilnehmern. Die Anzahl der Teilnehmer sollte 16 nicht überschreiten, sofern es nicht auch mehr als einen Moderator gibt [Gottesdiener2002].

Ebenso wie die Eigenschaften des Raums, sollten *Werkzeuge und Materialien* keine beeinflussenden Faktoren für das Workshop-Design darstellen – was sie in vielen Fällen aber faktisch sind. Sie werden gewöhnlich auch von den Eigenschaften des Raums beeinflusst: Wenn es beispielsweise in dem Raum kein großes Whiteboard gibt, kann dies während des Workshops natürlich nicht als Hilfsmittel verwendet werden. Stattdessen muss man auf andere Hilfsmittel wie Post-its zurückgreifen.

Wenn sich diese wechselseitigen Abhängigkeiten stabilisiert haben, können die Teilnehmer eingeladen werden. Der Moderator muss mit dem Workshop-Sponsor klären, wer zum Workshop eingeladen werden soll. Es muss auch abgeklärt werden, ob der Sponsor anwesend ist. Im Allgemeinen sollte der Workshop-Sponsor zur Eröffnung und nach Möglichkeit auch zum Abschluss des Workshops vor Ort sein. Während des Workshops ist die Präsenz des Workshop-Sponsors gewöhnlich nicht erforderlich und in manchen Fällen ist sie auch nicht zu empfehlen (z. B. wenn sich die hierarchische Position des Sponsors und die Workshop-Ergebnisse überlagern). Kommunizieren Sie Zeitpunkt und Ort sowie die Ziele des Workshops in der Einladung. Im Idealfall können Sie darin schon eine initiale *Workshop-Agenda* aufnehmen.

Die *Workshop-Agenda* ist das Ergebnis des Workshop-Designs. Sie dient als Orientierung für den Workshop. Eine Workshop-Agenda sollte genügend Puffer enthalten, damit die Zeiten während des Workshops angepasst werden können. Einige Workshop-Aktivitäten können wesentlich mehr Zeit in Anspruch nehmen als geplant, andere dagegen weniger.

Anwendung

Da die detaillierte Anwendung in hohem Maß vom Workshop-Design abhängt, enthält dieser Abschnitt nur allgemeine Ratschläge.

Vor dem Eintreffen der Teilnehmer muss der Raum vorbereitet werden (z. B. Poster aufhängen, Tische und Stühle platzieren usw.). Sie sollten auch sicherstellen, dass Erfrischungen für die Teilnehmer verfügbar sind.

Heißen Sie die Teilnehmer bei ihrer Ankunft willkommen.

Eröffnen Sie den Workshop und erklären Sie die Ziele sowie das erwartete Ergebnis des Workshops (dies kann auch der Workshop-Sponsor übernehmen).

Legen Sie die Grundregeln für den Workshop fest, im Idealfall in Zusammenarbeit mit den Teilnehmern.

Führen Sie die Gruppe durch den Workshop. Stellen Sie sicher, dass alle Teilnehmer involviert werden und der gesamte Input von den Teilnehmern aufgezeichnet wird. Überprüfen Sie

immer wieder, ob einer der folgenden Punkte angepasst werden muss: Workshop-Aktivitäten, Zeitrahmen, verwendete Werkzeuge, Raumklima.

Schließen Sie den Workshop ab: Danken Sie den Teilnehmern für ihren Input. Klären Sie die nächsten Schritte und Verantwortlichkeiten. Informieren Sie sie darüber, was mit den Workshop-Ergebnissen gemacht wird.

Ergebnisverarbeitung

Nach dem Workshop verarbeitet der Moderator oder der Protokollant die Workshop-Dokumentation und stellt sie den Teilnehmern zur Verfügung.

Da Workshop-Ergebnisse, ebenso wie die Einrichtung eines Workshops, sehr unterschiedlich ausfallen können, kann für die Ergebnisverarbeitung keine allgemeine Orientierung gegeben werden.

Typische Arbeitsprodukte

Arbeitsergebnisse hängen von der Art des Workshops (siehe Abschnitt „Varianten“), dem Workshop-Design und der Art der Aufzeichnung ab.

In der Regel werden Varianten der folgenden Arbeitsergebnisse erzeugt:

- Workshop-Agenda
- Workshop-Dokumentation (z. B. Bilder, Flipcharts, digitale Notizen, Protokolle)

Möglichkeiten

Anforderungswshops unterstützen die Kommunikation im Team und die Entscheidungsfindung. Sie bringen unterschiedliche Stakeholder zusammen, um ein besseres gegenseitiges Verständnis von Projekt oder Produkt zu erreichen. Potenzielle Konflikte und Missverständnisse, die bei anderen Ermittlungstechniken möglicherweise gar nicht zum Vorschein kommen, können im Laufe von Anforderungswshops identifiziert und gelöst werden.

Herausforderungen

Da Anforderungswshops viele Ressourcen (Personen, Zeit und Geld) erfordern, lassen sie sich ohne einen engagierten Sponsor kaum realisieren. Die für die Vorbereitung von Workshops und die Ergebnisverarbeitung benötigte Zeit wird häufig unterschätzt. Die Workshop-Ergebnisse hängen stark vom Workshop-Design ab. Unerfahrene Moderatoren neigen dazu, den erforderlichen Aufwand zu unterschätzen, und ihnen fehlt häufig die Flexibilität und das Instrumentarium, um einen effektiven Workshop vorzubereiten und durchzuführen und schließlich die Ergebnisse effizient zu verarbeiten.

Hinweis 3.1.4:

Wenn Sie in der Rolle des Moderators neu sind, sollten Sie in allen Phasen des Workshops mit einem erfahrenen Moderator zusammenarbeiten.

Varianten

Es gibt viele verschiedene Varianten von Anforderungsworkshops. Sie lassen sich mit anderen Workshop-Modellen kombinieren oder können deren Form annehmen, so etwa:

- **World Café** [BrIs2005]:
Das Setting besteht aus mehreren Tischen. Für jeden Tisch ist ein dediziertes Thema vorgesehen sowie ein Moderator. Die Gruppe eines Tisches (4 bis 6 Personen) bespricht das Thema 15 bis 30 Minuten lang. Anschließend werden die Teams gemischt und jeder nimmt an einem anderen Tisch Platz.
- **Open-Space-Technologie** [Owen2008]:
Die Teilnehmer schlagen Themen für die Diskussion vor. Es werden mehrere Themen zur gleichen Zeit an unterschiedlichen Orten (in anderen Räumen) diskutiert. Der Topic Owner ist für die Dokumentation der Diskussion verantwortlich. Den Teilnehmern steht es frei, sich an einer Diskussion zu beteiligen oder diese zu verlassen (das sogenannte „Gesetz der zwei Füße“). Am Ende der Open-Space-Phase kommen die Gruppen zusammen und die Topic Owner präsentieren der gesamten Gruppe die Ergebnisse.
- **Design Thinking:**
Vergleichen Sie hierzu die Beschreibung des Musters in Abschnitt 1.4.4.

3.1.3.2 Crowd-based Requirements-Engineering

Was ist Crowd-based Requirements-Engineering?

Crowd-based Requirements-Engineering (CrowdRE) ist auch als Crowd-centric Requirements Engineering (CCRE) bekannt.

Wir definieren CrowdRE gemäß [Groen et al.2017] als „Überbegriff für automatisierte oder halbautomatisierte Ansätze zum Sammeln und Analysieren von Informationen von einer Crowd, um daraus validierte Benutzeranforderungen abzuleiten.“

Solche Informationen können etwa Feedback sein, das in einem Forum gegeben wurde, Bewertungen und Reviews in App-Stores, Nutzungsdaten usw. Typische Techniken des CrowdRE sind die Auswertung von Texten und Nutzungsverhalten.

Ein großer Vorteil von CrowdRE ist das gesammelte Feedback der Crowd zur Unterstützung der Anforderungsermittlung. CrowdRE stützt sich auf die Zusammenarbeit zwischen den Menschen, die Teil der Crowd sind, und den Requirements Engineers.

Die Rolle der Teilnehmer

Der wichtigste Teilnehmer beim CrowdRE ist „die Crowd“, die gewöhnlich aus aktuellen oder künftigen Nutzern des zu entwickelnden Produkts besteht. Das CrowdRE ist jedoch nicht ausschließlich auf Nutzer beschränkt [LiFi2012].

Eine solche Crowd kann bereits vorhanden sein oder sie wird im Laufe des CrowdRE „geschaffen“, z. B. durch den Aufbau einer Gamification-Erfahrung, durch welche die Crowd gebildet wird [Snijders et al.2015].

Dem Requirements Engineer kommt bei dieser Ermittlungstechnik keine dominante Rolle zu. Es sammelt oder veranlasst die Techniken für das Sammeln von Daten von der Crowd und bewertet die Ergebnisse. Bei einigen Formen des CrowdRE wird es für die Crowd-Mitglieder nicht einmal erkennbar sein, dass sie an einer Anforderungsermittlung beteiligt sind (z. B. Reviews in App-Stores).

Da die Anwendung des CrowdRE speziell entwickelte Software erfordert (z. B. eine gamifizierte Ermittlungsplattform, Analysewerkzeuge für Benutzerdaten, ein Feedback-Modul im zu entwickelnden Produkt), wird für das CrowdRE gegebenenfalls auch ein Entwicklungsteam für die Erstellung einer solchen Software benötigt.

Vorbereitung

Da das CrowdRE sehr unterschiedliche Ausprägungen annehmen kann, gibt es für die Vorbereitung kein standardisiertes Vorgehen.

Im Allgemeinen lässt sich sagen, dass für das zu entwickelnde Produkt die richtige Crowd definiert werden muss, ebenso wie eine angemessene Möglichkeit, die Crowd einzubinden und die Ermittlungsergebnisse zu bewerten. Die Ermittlungsergebnisse sind in der Regel enorme Mengen von Rohdaten (Big Data). [Snijders et al.2015] und [LiFi2012] empfehlen konkrete Methoden/Werkzeuge.

Möglichkeiten

Das CrowdRE bietet die Möglichkeit, repräsentative Daten von tatsächlichen Benutzern zu erheben. Mit den richtigen Werkzeugen und einer geeigneten Vorbereitung bietet das CrowdRE das Potenzial, die Gesamtkosten der Ermittlung im Laufe der Zeit zu reduzieren. Damit lässt sich die Kundenbindung verbessern und auch die Kreativität der Kunden direkt nutzen, indem man sie aktiv einbindet. Darüber hinaus können Anforderungen nicht nur von der Crowd ermittelt, sondern auch priorisiert und validiert werden.

Herausforderungen

Für das CrowdRE ist in der Regel sehr viel Vorbereitungszeit nötig. Selbst wenn Daten verwendet werden, die bereits „vorhanden“ sind (z. B. Bewertungen in einem App-Store), müssen geeignete Wege gefunden und angewendet werden, um die Daten auszuwerten.

Wenn zur Förderung des CrowdRE bestimmte Plattformen bereitgestellt werden, dann müssen diese entwickelt oder gekauft und administriert werden. Lässt sich keine Crowd zur Teilnahme am CrowdRE motivieren, dann sind die Ergebnisse wertlos. Gibt es dagegen eine sehr aktive Crowd, dann werden enorme Mengen von Daten erzeugt, die zur Ableitung von Anforderungen ausgewertet werden müssen.

Ein weiterer potenzieller Stolperstein, den das CrowdRE bereithält, ist die Schaffung einer voreingenommenen Sicht, da nur das Feedback der Personen berücksichtigt wird, die sich an der Crowd-Kommunikation beteiligen. Das CrowdRE ist außerdem anfällig für Manipulationen und Scherze (bei einem Crowdsourcing-Wettbewerb von Henkel, in dessen Rahmen ein neues Verpackungsdesign für Geschirrspülmittel gefunden werden sollte, erhielt eine Verpackung mit dem Slogan „Schmeckt lecker nach Hähnchen“ die meisten Stimmen [Katie2017]).

Varianten

Crowd-Tests sind ein verwandter Bereich, in dem die Crowd zur Abgabe von wertvollem Feedback instrumentalisiert wird. Wenn ein (Usability-) Defekt identifiziert wird, werden die Teilnehmer einer Crowd-Testaktivität häufig um Vorschläge für eine bessere Lösung gebeten. In einem solchen Fall kann es nützlich sein, mit den Testern zu kooperieren, um Defekte und Anforderungen aus derselben Crowd-Basis zu sammeln.

3.1.4 Artefaktbasierte Techniken

Artefakte (auch Arbeitsergebnisse genannt) sind Ergebnisse menschlicher Tätigkeit, 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 ein zeitaufwändiges Unterfangen, Arbeitsergebnisse detailliert zu untersuchen – vor allem, wenn die Arbeitsergebnisse zu viele (irrelevante) Informationen enthalten oder schlecht strukturiert sind. Dennoch sind artefaktbasierte Techniken sehr wirksam und bieten signifikante Vorteile, insbesondere wenn Stakeholder nicht ohne Weiteres verfügbar sind.

In diesem Abschnitt behandeln wir drei artefaktbasierte Techniken: perspektivenbasiertes Lesen, Systemarchäologie und Wiederverwendung von Anforderungen.

Perspektivenbasiertes Lesen ist die Haupttechnik und eine grundlegende Fähigkeit für die beiden anderen Techniken. Zwischen diesen Techniken gibt es auch einige Überschneidungen. Daher ist es wichtig, die Essenz der einzelnen Techniken zu verstehen.

3.1.4.1 Perspektivenbasiertes Lesen

Was ist perspektivenbasiertes Lesen?

Perspektivenbasiertes Lesen wurde erstmals von [BaGL1996] als eine Review-Technik untersucht. Sie fanden heraus, dass Reviewer mehr Defekte in weniger Zeit fanden, wenn sie bei Anforderungsdokumenten das perspektivenbasierte Lesen anstelle ihrer üblichen Review-Weise anwendeten. Eine Review-Technik besteht darin, dass jeder Benutzer die Perspektive eines bestimmten Nutzers des Anforderungsdokuments einnimmt (z. B. Entwickler, Tester oder Architekt).

Diese Technik hat sich seitdem auch als eine Technik für die Anforderungsermittlung bewährt, wenn Dokumente als Anforderungsquellen dienen.

Die Rolle der Teilnehmer

Die einzige Rolle, die für das perspektivenbasierte Lesen erforderlich ist, ist die des Lesers.

Vorbereitung

Definieren Sie das bzw. die Ermittlungsziele und die Ergebnisqualität im Rahmen der Definition der Ermittlungsaktivität (siehe 1.3.1).

Definieren Sie das Quelldokument und wählen Sie die passende(n) Perspektive(n) dafür aus.

Welche Perspektiven geeignet sind, hängt vom Quelldokument und dem Ermittlungsziel ab. Folgende Perspektiven könnten geeignet sein: Anforderungen für bestimmte Funktionalitäten, Anforderungen an die Usability, funktionale Anforderungen, nicht-funktionale Anforderungen usw.

Klären Sie, wie viele Leser (Requirements Engineers) am perspektivenbasierten Lesen teilnehmen.

Anwendung

Ein Leser kann in einer Lesesitzung nur eine Perspektive einnehmen. Ist mehr als eine Perspektive für ein Quelldokument relevant, dann muss der Requirements Engineer diese Perspektiven nacheinander einnehmen.

Gibt es mehrere Leser, dann übernimmt jeder Leser eine andere Perspektive. Die Leser können die Dokumente parallel lesen (wenn es ein digitales Dokument ist oder wenn mehrere analoge Kopien verfügbar sind), sogar im selben Raum oder asynchron. In letzterem Fall sollte ein Zieldatum vereinbart werden.

Beim Lesen des Quelldokuments scannen die einzelnen Leser das Dokument auf Inhalte, die für seine aktuelle Perspektive relevant sind, und lesen anschließend alle potenziell

interessanten Abschnitte gründlicher. Sobald sie auf Informationen stoßen, die aus ihrer Perspektive relevant sind, zeichnen sie diese in geeigneter Weise auf.

Beispiel:

Das Quelldokument ist das Benutzerhandbuch eines Wettbewerbers. Ein Leser nimmt die Perspektive „Qualitätsanforderungen“ ein, ein anderer die Perspektive „beeindruckende Features“. Beide sehen das Benutzerhandbuch durch und konzentrieren sich dabei auf Informationen, die aus ihrer jeweiligen Perspektive relevant sind. Wenn sie im Handbuch einen potenziell interessanten Abschnitt finden, lesen sie diesen gründlich und extrahieren die Anforderungen. Der erste Leser hält vielleicht bei allen Sätzen an, die Zahlen enthalten (Indikatoren für potenzielle nicht-funktionale Anforderungen); der zweite Leser hält möglicherweise bei bestimmten Kapiteln an, in denen Systemfunktionalitäten beschrieben werden.

Ergebnisverarbeitung

Passagen, die für das System als potenziell relevant identifiziert werden, müssen durch die Anwendung anderer Ermittlungstechniken oder durch ein Review der Anforderungen verifiziert werden. Wir empfehlen dringend, das perspektivenbasierte Lesen mit Techniken zu kombinieren, bei welchen Stakeholder im Mittelpunkt stehen (z. B. Interview, Contextual Inquiry).

Typische Arbeitsprodukte

Bei der Vorbereitung, Durchführung und Nachbereitung des perspektivenbasierten Lesens werden üblicherweise folgende Arbeitsergebnisse erzeugt oder aktualisiert:

- Notizen aus der ursprünglichen Ermittlung geeigneter Dokumente
- Potenziell: physische oder digitale Kopien von Dokumenten mit Markierungen und/oder Notizen zu relevanten Abschnitten
- Dokumentation von extrahierten Informationen (Anmerkungen)
- verarbeitete Ergebnisse

Möglichkeiten

Wenn Stakeholder nicht ohne Weiteres verfügbar sind, ist das perspektivenbasierte Lesen eine gute Möglichkeit, um potenzielle Anforderungen zu ermitteln. Darüber hinaus wird bereits dokumentiertes Wissen wiederverwendet und geht nicht verloren.

Herausforderungen

Es ist häufig schwer festzustellen, ob ein Dokument noch aktuell oder gültig ist. Wenn beim perspektivenbasierten Lesen veraltete Dokumente verwendet werden, sind die verwertbaren Ergebnisse gemessen an der investierten Zeit unter Umständen rar.

Wie bei allen artefaktbasierten Techniken, weiß der Requirements Engineer nie genau, welche aus der Dokumentation ermittelten Anforderungen für das zu entwickelnde System relevant bleiben. Dafür werden weitere Techniken benötigt.

Varianten

Nicht zutreffend.

3.1.4.2 Systemarchäologie

Was ist Systemarchäologie?

Bei der Systemarchäologie werden Anforderungen aus vorhandenen Systemen extrahiert: Altsysteme ebenso wie Wettbewerbersysteme oder sogar analoge Systeme (Systeme in einem anderen Kontext mit ähnlicher Funktionalität). Wir bezeichnen diese in der folgenden Diskussion als *Quellsysteme*.

Diese Technik wird hauptsächlich angewendet, wenn ein vorhandenes System über viele Jahre hinweg verwendet (und möglicherweise verändert) wurde und jetzt durch ein neues System ersetzt werden soll, da beispielsweise die Technologie des Altsystems mit den angrenzenden Systemen nicht mehr kompatibel ist, dessen Performance die Anforderungen nicht mehr erfüllt oder neue Funktionalitäten benötigt werden.

Die Rolle der Teilnehmer

Bei der Systemarchäologie sind keine speziellen Rollen involviert.

Vorbereitung

Definieren Sie das bzw. die Ermittlungsziele und die Ergebnisqualität im Rahmen der Definition der Ermittlungsaktivität (siehe Abschnitt 1.3.1). Das Ermittlungsziel gibt Ihnen möglicherweise auch Hinweise darauf, wo Sie nachsehen können: Wenn beispielsweise das Ermittlungsziel darin besteht, die Anforderungen für die Daten des Systems zu ermitteln, dann ist gegebenenfalls die Systemdatenbank ein guter Ausgangspunkt.

Wählen Sie das Quellsystem aus, aus dem Sie die Anforderungen ableiten möchten. In den meisten Fällen wäre das ein aktuell in Betrieb befindliches System (Altsystem), das durch das zu entwickelnde System ersetzt werden soll. Wettbewerbersysteme oder Systeme, die analog zu dem zu entwickelnden System sind, können auch als Quellen für die Systemarchäologie dienen.

Sammeln Sie als Nächstes alle potenziell relevanten Dokumente für das Quellsystem, z. B. Benutzerhandbuch, Testfälle, Dokumentation zur Architektur und Projektchartas. Wenn Sie an Anforderungsdokumentation gelangen können, nutzen Sie die Technik der Wiederverwendung von Anforderungen (Abschnitt 3.1.4.3).

In manchen Fällen ist keine aktuelle, nützliche Dokumentation für das Quellsystem verfügbar, oder die Dokumentation für die Systemarchäologie reicht nicht aus. Dann benötigen Sie gegebenenfalls das in einer geeigneten Umgebung ausführbare Quellsystem (in der Regel eine Testumgebung, in der Daten geändert werden können, ohne Livedaten zu beeinträchtigen) und/oder den Quellcode. Wenn Sie Informationen aus dem Quellcode extrahieren müssen, Sie aber die Programmiersprache nicht kennen, werden Sie jemanden zur Unterstützung benötigen.

Anwendung

Die Anwendung ist unterschiedlich, je nachdem, ob Dokumentation als Quelle für die Systemarchäologie dient oder ob es der Quellcode oder das System selbst ist.

Dokumentation

Wenn Ihnen Dokumentation als Quelle dient, z. B. Benutzerhandbücher oder Testfälle, wenden Sie das perspektivenbasierte Lesen an (Abschnitt 3.1.4.1).

Quellcode

Abhängig von der verwendeten Programmiersprache gelten unterschiedliche Lesestrategien für den Quellcode (z. B. Assembler- vs. objektorientierte Programmiersprache). Gut dokumentierter Code wird außerdem viel einfacher zu lesen sein als nicht dokumentierter Code.

Es ist generell nützlich herauszufinden, wie der Code strukturiert ist und welche Namenskonventionen verwendet wurden. Außerdem sollte man sich mit dem Programmierstil vertraut machen, bevor man ins Detail geht.

Eine Quellcodeanalyse ist nur erforderlich, um spezifische Details der Implementierung herauszufinden.

Beispiel:

Das Quellsystem lässt sich in einer Testumgebung ausführen, und Sie konnten die wichtigsten Features und Funktionalitäten extrahieren. Es gibt jedoch ein Feature, für das Sie ein bestimmtes Verhalten in der Testumgebung nicht simulieren können, da Ihnen nicht genügend Testdaten zur Verfügung stehen. Daher suchen Sie dieses spezielle Feature im Quellcode und analysieren die dafür implementierten Regeln.

Ausführbares Quellsystem (UI-Analyse)

Ist das Quellsystem in ausführbarer Form verfügbar, kann das System auf strukturierte Weise analysiert werden. Folgende Verfahren haben sich bewährt:

- **Folgen Sie den im System implementierten Geschäftsprozessen:**
 - Folgen Sie einem Business Case nach dem anderen.
 - Achten Sie auf Ausnahmen und Verzweigungen.
 - Oftmals nützlich: Dokumentieren Sie das gewonnene Wissen mithilfe von Modellen (z. B. BPMN, UML).
- **Folgen Sie der UI-Struktur:**
 - Analysieren Sie einen Bildschirm nach dem anderen, beginnend mit dem Startbildschirm.
 - Analysieren Sie ein Element nach dem anderen und folgen Sie dabei der Schreibrichtung der verwendeten Sprache (bei Englisch z. B. von links nach rechts und von oben nach unten).

- Dokumentieren Sie die Bildschirmablauflogik und die Funktionalität von Schaltflächen anhand von Screenshots und/oder Storyboards oder anderen geeigneten Methoden.

Ergebnisverarbeitung

Anforderungen, die für das zu entwickelnde System als potenziell relevant identifiziert werden, müssen durch die Anwendung anderer Ermittlungstechniken oder durch ein Review der Anforderungen verifiziert werden. Wir empfehlen dringend, die Systemarchäologie mit weiteren Techniken zu kombinieren, bei denen Stakeholder im Mittelpunkt stehen (z. B. Interview, Contextual Inquiry).

Typische Arbeitsprodukte

Bei Vorbereitung, Durchführung und Nachbereitung der Systemarchäologie werden üblicherweise folgende Arbeitsergebnisse erzeugt oder aktualisiert:

- Notizen aus der ursprünglichen Ermittlung geeigneter Quellsysteme und deren Dokumentation
- Potenziell: physische oder digitale Kopien von Dokumenten mit Markierungen und/oder Notizen zu relevanten Abschnitten
- Dokumentation von extrahierten Informationen (Anmerkungen und Modellen oder Zeichnungen)
- verarbeitete Ergebnisse (z. B. modellbasierte Dokumentation von identifizierten Prozessen)

Möglichkeiten

Durch die Systemarchäologie wird sichergestellt, dass Anforderungen, die im Quellsystem implementiert sind, nicht verloren gehen. Sie ist vor allem nützlich, wenn keine andere Dokumentation verfügbar ist, um herauszufinden, was das System *tatsächlich* tut. Zudem ist sie nicht von der Verfügbarkeit von Stakeholdern abhängig.

Herausforderungen

Die Systemarchäologie ist potenziell sehr zeitintensiv, da viele Dokumente oder Codezeilen auf Relevanz geprüft und gelesen werden müssen. Ein besonderer Stolperstein ist, dass diese Technik nur dazu beiträgt, die bestehende Funktionalität zu ermitteln. Damit ist noch nicht geklärt, ob diese Funktionalität noch benötigt wird oder überhaupt korrekt ist. Im schlechtesten Fall werden Fehler aus dem alten System wieder in das neue System implementiert!

Varianten

Nicht zutreffend.

3.1.4.3 Wiederverwendung von Anforderungen

Was ist die Wiederverwendung von Anforderungen?

Wenngleich jedes Projekt per definitionem einzigartig ist und damit in einem einzigartigen Produkt resultiert, besteht bei jedem Projekt eine gute Chance, dass es Wiederverwendungspotenzial bietet, auch für Anforderungen (z. B. Rollen und Profile oder Anmeldeverfahren).

Durch bewusstes Wiederverwenden wird vermieden, dass man das Rad immer wieder neu erfindet.

Es gibt verschiedene Formen der Wiederverwendung von Anforderungen. Bei Produktlinien kann in der Regel ein hoher Prozentsatz der Anforderungen wiederverwendet werden, entweder so wie sie sind oder mit gewissen Änderungen.

Beispiel:

Die Fluxkompensatoren der ersten und der zweiten Version der Zeitmaschine von DeLorean werden sich in einigen Features unterscheiden, insgesamt jedoch dieselbe Funktionalität aufweisen.

Selbst wenn zwei Systeme auf den ersten Blick völlig unterschiedlich erscheinen, können ihre Anforderungen dennoch Ähnlichkeiten aufweisen.

Beispiel:

Die Einstellvorrichtung für Zeit und Ort und das Heizsystem derselben Maschine habe völlig verschiedene Funktionalitäten. Aber da sie beide Teil desselben Systems sind (der Zeitmaschine), ist ihnen ein hoher Prozentsatz nicht-funktionaler Anforderungen gemein (z. B. maximale Beschleunigung).

Sogar Systeme, die keine Schnittstellen untereinander haben und die nicht Teil eines gemeinsamen Supersystems sind, können gemeinsame Anforderungen haben, da durch ihre Prozesse ähnliche Schritte abgedeckt werden.

Beispiel:

Ein Flugzeug und eine Zeitmaschine haben insofern etwas gemeinsam, dass sie beide Personen befördern. Einige Anforderungen bezüglich lebenserhaltender Systeme (z. B. Sauerstoffversorgung) oder Sicherheitsanforderungen sind wahrscheinlich für beide gleich.

Die Wiederverwendung von Anforderungen umfasst drei Aspekte: die Ermittlung, die Dokumentation und das Anforderungsmanagement. Der Aspekt der Ermittlung deckt das allgemeine Prinzip der Wiederverwendung und deren Rolle bei der Erhebung von Anforderungen für ein neues Projekt ab. Der Aspekt der Dokumentation deckt die Frage ab, wie Anforderungen zur Unterstützung einer späteren Wiederverwendung dokumentiert

werden sollen. Das Anforderungsmanagement greift die Frage auf, wo und wie Anforderungen zur Unterstützung einer späteren Wiederverwendung organisiert werden sollen.

In diesem Handbuch behandeln wir nur den ersten Aspekt.

Hinweis 3.1.5:

Wenn Ihr Produkt in verschiedenen Varianten verfügbar sein wird oder wenn es mehrere Versionen geben wird, sollten Sie einige Überlegungen und Zeit in den Aufbau der Dokumentationsstruktur investieren. Verwenden Sie dazu ein Anforderungsmanagement-Werkzeug, um eine Wiederverwendung der ermittelten Anforderungen zu erleichtern.

Die Rolle der Teilnehmer

Bei der Wiederverwendung von Anforderungen sind keine speziellen Rollen involviert.

Vorbereitung

Definieren Sie das bzw. die Ermittlungsziele und die Ergebnisqualität im Rahmen der Definition der Ermittlungsaktivität (siehe Abschnitt 1.3.1).

Wählen Sie die Dokumente aus, die Anforderungen für die Wiederverwendung enthalten könnten. Typische Dokumente hierfür sind Anforderungsspezifikationen von angrenzenden Systemen oder von Vorgängersystemen des zu entwickelnden Systems. Die Wiederverwendung von funktionalen Anforderungen ist in der Regel nur zwischen ähnlichen Systemen möglich. Nicht-funktionale Anforderungen können auch zwischen Systemen wiederverwendet werden, die auf den ersten Blick scheinbar nicht viele Ähnlichkeiten aufweisen. Dadurch sind nicht-funktionale Anforderungen für die Wiederverwendung von Anforderungen besonders interessant.

[RoRo2013] schlagen vor, mithilfe ihres *Volere*-Templates, [Robles2012], Kapitel 15, nach Dokumenten mit potenziell wiederverwendbaren Inhalten zu suchen:

1. Der Projektzweck: Gibt es andere Projekte in der Organisation, die kompatibel sind oder die im Wesentlichen dieselben Fachgebiete oder Arbeitsbereiche abdecken?
2. Der Auftraggeber, der Kunde und andere Stakeholder: Können Sie eine vorhandene Liste von Stakeholdern, einen Stakeholder-Plan oder eine Stakeholder-Analysearbeitsblatt wiederverwenden? Benutzer des Produkts: Binden andere Produkte dieselben Benutzer ein und haben dadurch ähnliche Usability-Anforderungen?
3. Vorgeschriebene Randbedingungen: Wurden Ihre Randbedingungen bereits bei einem anderem Projekt angegeben? Gibt es irgendwelche organisationsübergreifenden Randbedingungen, die auch für Ihr Projekt gelten?
4. Namenskonventionen und Definitionen: Sie können sehr wahrscheinlich Teile eines vorhandenen Glossars nutzen.
5. Relevante Fakten und Annahmen: Schenken Sie relevanten Fakten aus kürzlich durchgeführten Projekten Ihre Aufmerksamkeit. Gelten Annahmen aus anderen Projekten auch für Ihr Projekt?

6. Der Systemumfang der Arbeit: Bei Ihrem Projektgegenstand besteht eine sehr hohe Wahrscheinlichkeit, dass es ein angrenzendes System zu anderen Projekten wird, die in Ihrer Organisation ausgeführt werden. Nutzen Sie die Schnittstellen, die durch Modelle anderer Arbeitskontexte erstellt wurden. Berücksichtigen Sie den Systemumfang Ihrer Arbeit und bringen Sie in Erfahrung, ob in anderen Projekten bereits ähnliche Geschäftsereignisse definiert wurden.
7. Geschäftsdatenmodell und Data Dictionary: Gibt es Geschäftsdatenmodelle aus sich überschneidenden oder miteinander verbundenen Projekten, die Sie als Ausgangspunkt verwenden könnten?

Anwendung

Suchen Sie in den identifizierten Dokumenten nach Anforderungen, die für Ihr aktuell zu entwickelndes Produkt relevant sein könnten. Dies ist eine Anwendung des perspektivenbasierten Lesens (vgl. Abschnitt 3.1.4.1). [RoRo2013] empfehlen, sich bei der Suche nach wiederverwendbaren Anforderungen auf die Verben zu konzentrieren, da anhand von Verben Prozesse dargestellt werden.

Es wird nur wenige wiederverwendbare Anforderungen geben, die Ihnen auf dem Silbertablett serviert werden. Um die anderen zu finden, werden Sie das Gelesene interpretieren und abstrahieren müssen.

Ergebnisverarbeitung

Die Anforderungen, die als potenziell zur Wiederverwendung geeignet identifiziert werden, müssen durch die Anwendung anderer Ermittlungstechniken oder durch ein Review der Anforderungen verifiziert werden. Wir empfehlen dringend, die Wiederverwendung von Anforderungen mit Techniken zu kombinieren, bei welchen Stakeholder im Mittelpunkt stehen (z. B. Interview, Contextual Inquiry).

Typische Arbeitsprodukte

Bei der Vorbereitung, Durchführung und Nachbereitung der Wiederverwendung von Anforderungen werden üblicherweise die folgenden Arbeitsergebnisse erzeugt oder aktualisiert:

- Notizen aus der Ermittlung geeigneter Dokumente
- potenziell: physische oder digitale Kopien von Original-Anforderungsdokumenten mit Markierungen und/oder Notizen zu wiederverwendbaren Abschnitten
- verarbeitete Ergebnisse, z. B. abstrakte Modelle von vorhandenen Spezifikationen (siehe [RoRo2013])

Möglichkeiten

Durch die Wiederverwendung von Anforderungen lässt sich die Neuentwicklung von bereits Vorhandenem vermeiden. Außerdem trägt dies dazu bei, wichtige Anforderungen nicht zu vergessen. Wenn die verwendeten Anforderungsspezifikationen gut strukturiert und gut geschrieben sind, können sehr viele Anforderungen innerhalb kurzer Zeit ermittelt werden.

Herausforderungen

Wenn die verwendeten Anforderungsspezifikationen weder gut strukturiert noch gut geschrieben und auch nicht aktuell sind, kann die Wiederverwendung von Anforderungen zu einem sehr zeitaufwendigen Unterfangen werden und unter Umständen nicht das gewünschte Ergebnis erbringen.

Eine andere potenzielle Gefahr besteht darin, dass falsche Anforderungen übernommen werden, die nicht auf das neue System anwendbar sind oder die sogar bereits im Originaldokument falsch waren. Hinzu kommt, dass durch das Kopieren vorhandener Lösungen möglicherweise Innovationen unterdrückt werden.

Varianten

Nicht zutreffend.

3.2 Entwurfs- und Ideenfindungstechniken (K2)

Die herkömmliche Aufgabe des Requirements Engineering bestand im Sammeln und Dokumentieren der notwendigen Anforderungen von allen relevanten Stakeholdern für einen nachfolgenden Entwicklungsprozess (vgl. [Boehm2006]). Diese Aufgabe führte zur Anwendung von Erhebungstechniken, die bereits vorgestellt wurden (siehe Abschnitt 3.1). Der zunehmende Einfluss von Software als Innovationstreiber in vielen Geschäftsbereichen führte schließlich zu einem neuen Verständnis des Requirements Engineering als einer kreativen Aktivität zur Problemlösung (vgl. [Couger1996] oder [Maiden et al.2010]).

Außerhalb der Software- und Requirements Engineering-Welt hat sich der weiter gefasste Begriff „Entwurfstechniken“ (engl. design techniques) entwickelt.

Entwurfstechniken umfassen Kreativitätstechniken zur Ideenentwicklung und bieten weiterhin zusätzliche oder kombinierte Techniken, um Ideen auszuarbeiten und weitere Erkenntnisse zu einer bestimmten Idee zu gewinnen [Kumar2013]. Beliebte Entwurfstechniken sind u. a. die Verwendung von Prototypen (z. B. Mock-ups), Storyboards und Szenarien.

Nachstehend präsentieren wir zwei Kreativitätstechniken (das Brainstorming und die Analogietechnik) und zwei Entwurfstechniken (die Verwendung von Prototypen und Szenarien/Storyboards) als Beispiele für Techniken, die in verschiedenen Kontexten einsetzbar sind. In der Literatur finden sich jedoch viele weitere Techniken (vgl. z. B. [Kumar2013]). Die Disziplin der Entwurfstechniken beinhaltet auch mehrere Prinzipien, die einerseits den Entwicklungsprozess von Systemen unterstützen (z. B. Prozesskonzepte) und andererseits die Entwicklung des Systems an sich (z. B. Art, Form oder Funktionalität des Systems). Die Präsentation und Besprechung dieser Prinzipien geht über den Umfang dieses Handbuchs hinaus. Ein guter Überblick über Entwurfsprinzipien wird beispielsweise in [LiHB2003] gegeben.

Voraussetzungen für Kreativität

Auch wenn es viele Techniken gibt, die auf das Erzeugen kreativer Ergebnisse abzielen, bietet keine von ihnen einen garantierten Erfolg. In unserem Gehirn müssen verschiedene

Mechanismen zusammenkommen, damit kreative Ideen möglich werden. Vereinfacht gesagt, müssen die folgenden vier Voraussetzungen erfüllt sein, damit Kreativität entstehen kann:

- **Zufall** – und somit Zeit – damit eine Idee entstehen kann:
Eine Idee ist eine neue Verbindung zwischen Neuronen. Ob eine solche Verbindung entsteht, ist in erster Linie eine Sache des Zufalls. Dies kann jedoch durch die positive Beeinflussung der anderen drei Voraussetzungen für Kreativität verbessert werden, vor allen Dingen, indem man ihr Zeit gibt.
- **Wissen** über das betreffende Thema, wodurch die Wahrscheinlichkeit steigt, dass eine wirklich relevante Idee entsteht.
Wissen ist der Nährboden, auf dem kreative Ideen gedeihen können. Durch den Erwerb von Wissen zu einem bestimmten Thema werden die entsprechenden Regionen in unserem Gehirn aktiviert, wodurch sich die Chance auf die Entstehung neuer neuronaler Verbindungen erhöht. Das bedeutet, wenn wir neue, innovative Ideen zu einem bestimmten Thema entwickeln möchten, müssen wir uns zunächst Wissen zu diesem Thema aneignen, um die entsprechenden Regionen in unserem Gehirn zu aktivieren. Das bedeutet nicht, dass nur Fachexperten mit Innovationen in ihrem Wissensgebiet aufwarten können; Außenstehende können ebenso frische, unkonventionelle Ideen einbringen – **sofern** sie genug über das entsprechende Thema wissen (z. B. ein bestimmter Schritt in einem Prozess, der verbessert werden sollte, oder Funktionalitäten und Randbedingungen eines vorhandenen Produkts, die überarbeitet werden müssen).
- **Motivation**, da unser Gehirn nur kreativ sein kann, wenn für seinen Besitzer ein direkter Nutzen daraus entsteht:
Kreativ zu sein, ist eine Aufgabe, für die unser Gehirn viel Energie braucht. Die wesentliche Richtlinie unseres Gehirns lautet: Energie sparen. Daher wird es die erforderliche Energie nur dann investieren, wenn es sich davon einen Vorteil verspricht. Das könnte ein höherer Status (d. h. eine bessere Überlebenschance) oder ganz einfach Spaß sein.
- **Sicherheit**, da nutzlose Ideen keine negativen Auswirkungen haben dürfen:
Wenn wir fürchten, dass eine nutzlose Idee in irgendeiner Weise unseren Status oder unsere Sicherheit bedroht, fährt unser Gehirn jegliche Kreativität zurück und wird „leer“. Es schaltet in einen Sicherheitsmodus und hält an bekannten und sicheren Vorgehensweisen fest.

Bei der Anwendung von Entwurfs- oder Ideenfindungstechniken muss sichergestellt werden, dass von den Teilnehmern die vier Voraussetzungen für Kreativität erfüllt werden.

3.2.1 Brainstorming

Was ist Brainstorming?

Das Brainstorming wurde von Alex F. Osborn [Osbo1948] als eine Gruppenkreativitätstechnik zur Unterstützung der Entwicklung neuer Ideen zu einer gegebenen Frage oder einem Problem entwickelt. Wie bei den meisten Kreativitätstechniken, ist der entscheidende Aspekt

des Brainstormings, Bewertungen zurückzustellen, indem man die Ideenfindung von der Analyse der Ideen trennt. [MaGi2001] und [Pohl2010] geben einen guten Überblick über das Thema Brainstorming im Bereich des RE.

Die Rolle der Teilnehmer

Die Ideen werden von den Teilnehmern einer Brainstorming-Sitzung entwickelt. Ein Moderator kümmert sich um die Einhaltung der Regeln des Brainstormings. Zur Unterstützung des Moderators bei der Visualisierung der entwickelten Ideen kann der Protokollant hinzugezogen werden.

Vorbereitung

Vor einem Brainstorming muss eine Leitfrage definiert werden. Das Ermittlungsziel (siehe Lehrplan LE 1) kann als Ausgangspunkt oder sogar direkt als Leitfrage verwendet werden. In einem zweiten Schritt werden die Teilnehmer der Brainstorming-Sitzung ermittelt. Als Faustregel gilt, dass die Teilnehmer aus verschiedenen Bereichen kommen sollten. Es ist nicht nötig, dass alle Sitzungsteilnehmer ein tiefgehendes Verständnis der Frage haben. Es kann sogar von Vorteil sein, wenn einige Teilnehmer aus anderen Fachgebieten dabei sind, da sie gänzlich andere Sichtweisen auf die Fragestellung einbringen können.

Anwendung

Zu Beginn einer Brainstorming-Sitzung erklärt der Moderator den Teilnehmern die Leitfrage und die Regeln des Brainstormings. [Pohl2010] beschreibt diese sieben Regeln (gestützt auf Osborn) für das Brainstorming im Requirements Engineering:

- Quantität vor Qualität
- Freies Assoziieren und visionäres Denken sind ausdrücklich erwünscht.
- Das Übernehmen und Kombinieren von geäußerten Ideen ist erlaubt und erwünscht.
- Das Kritisieren von Ideen anderer Teilnehmer ist verboten, selbst dann, wenn eine Idee noch so absurd erscheint.
- Klärende Fragen sind zulässig.
- Selbst bei längerem Stillstand sollten Sie nicht gleich abbrechen – überstehen Sie mindestens zwei länger andauernde Phasen des Stillstands.
- Warten Sie ab, bis Ihr Brainstorming zu einem natürlichen Ende findet.

Der Moderator beginnt die Sitzung, und alle Teilnehmer können ihre Ideen zur Leitfrage äußern. Der Moderator (oder ein zusätzlicher Protokollant) visualisiert die Ideen für alle Teilnehmer. Der Ideenfluss in einer Brainstorming-Sitzung kommt gewöhnlich nach einer gewissen Zeit zum Stillstand. Solche Pausen sind aber noch kein Indikator für das Ende des Brainstormings. Schon eine neue Anmerkung eines Teilnehmers kann den Ideenfluss sofort wieder in Gang setzen. Die Gruppe sollte mindestens zwei solcher Pausen überwinden, bevor die Sitzung zu einem natürlichen Ende kommt (siehe vorgenannte Regeln).

Ergebnisverarbeitung

Nachdem die Brainstorming-Sitzung stattgefunden hat, muss die Ideenliste analysiert werden. Jede Idee besteht typischerweise aus einer kurzen Aussage. Daher muss

zusätzlicher Aufwand auf die detaillierte Ausarbeitung der besten Ideen verwendet werden, beispielsweise indem man einen zusätzlichen Workshop mit einigen Teilnehmern der Brainstorming-Sitzung durchführt. In der Regel werden bei der Verarbeitung von Ergebnissen eines Brainstormings Priorisierungstechniken angewandt.

Typische Arbeitsprodukte

Das Ergebnis einer Brainstorming-Sitzung ist die Liste der erarbeiteten Ideen. Wenn alle Brainstorming-Teilnehmer einverstanden sind, kann eine Audio- oder Videoaufzeichnung des Brainstormings erstellt und als zusätzliche Grundlage für die spätere Analyse verwendet werden.

Chancen

Ein Brainstorming ist hilfreich, um innerhalb kurzer Zeit eine große Anzahl von Ideen zu entwickeln. Ideen, die zu Beginn möglicherweise als absurd betrachtet wurden, können der Gruppe als Inspiration für die Entwicklung anderer unerwarteter und innovativer Ideen dienen. Ein wesentlicher Nutzen einer guten Brainstorming-Sitzung besteht darin, dass sie diesen gruppenspezifischen Effekt unterstützt (d. h. Ideen, die andere zu wiederum neuen Ideen inspirieren).

Herausforderungen

Brainstorming-Ergebnisse sind nur grobe Entwürfe von Ideen. Die eigentliche Requirements-Engineering-Arbeit beginnt nach der Brainstorming-Sitzung. Die entwickelten Ideen müssen bewertet, priorisiert und weiter detailliert werden.

Der Hauptantrieb für die Entwicklung guter Ideen bei dieser Technik sind gruppenspezifische Effekte. Daher ist es die wichtigste Herausforderung in einer Brainstorming-Sitzung, eine kreative Atmosphäre herzustellen und aufrechtzuerhalten.

Varianten

Über die Jahre sind viele verschiedene Varianten des Brainstormings entstanden, z. B.:

- Das Brainstorming Paradox folgt demselben Verfahren wie ein normales Brainstorming. Der Unterschied ist das Brainstorming-Thema, das das Gegenteil des normalen Brainstorming-Themas ist. Das Ziel dabei ist, Risiken und Gefahren im Zusammenhang mit einem Thema zu untersuchen (z. B. „Welche Faktoren tragen dazu bei, potenzielle Käufer zu verärgern, und sorgen dafür, dass sie unseren Webshop verlassen?“).

3.2.2 Analogietechnik

Was ist die Analogietechnik?

Analogietechniken (vgl. z. B. [Robertson2001]) bieten Unterstützung dabei, 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. Gute Quellen für Analogien sind verbundene Systeme (siehe Kapitel 2).

Die Rolle der Teilnehmer

Der Requirements Engineer übernimmt die Rolle des *Moderators*, der die Analogie entwickelt und der Gruppe vorstellt.

Die Teilnehmer der Analogietechnik arbeiten die gegebene Analogie aus und übertragen die entwickelten Ideen auf das ursprüngliche Problem.

Vorbereitung

Die Vorbereitung der Analogietechnik besteht aus zwei Schritten.

Wählen Sie für das bestehende Problem eine geeignete Analogie. Die ausgewählte Analogie kann nahe am ursprünglichen Problem liegen (z. B. dasselbe Problem in einem anderen Fachgebiet) oder auch weit davon entfernt. Das ursprüngliche Problem ist beispielsweise, den Einkaufsprozess von Lebensversicherungsprodukten über das Internet zu verbessern. Eine enge Analogie könnte der Einkaufsprozess eines anderen Produkts über das Internet sein (z. B. der Einkauf eines Buches anstatt einer Versicherungspolice in einem Onlineshop). Eine entfernte Analogie könnte eine Situation sein, in der ein Kellner den Gast zur Speisenauswahl berät (die Speisekarte ist die Analogie zur Versicherungspolice).

Wählen Sie die Teilnehmer für die Analogietechnik aus. Diese sollten Experten für das ursprüngliche Problem sein. Nach Möglichkeit sollten Experten (oder zumindest fachkundige Personen) für die ausgewählte Analogie eingeladen werden. Es ist nicht nötig, dass ein Teilnehmer sowohl Experte für das ursprüngliche Problem als auch für die Analogie ist. Es ist auch möglich, die Analogie gemeinsam mit den Teilnehmern auszuwählen.

Anwendung

Die Anwendung der Analogietechnik erfolgt in zwei Schritten:

- Arbeiten Sie die ausgewählte Analogie weiter aus. Die Teilnehmer arbeiten positive und negative Aspekte der Analogie detailliert aus (z. B. unter Verwendung von Whiteboards oder eines Flipcharts). Dieser Schritt sollte ohne Verweis auf das ursprüngliche Problem durchgeführt werden.
- Übertragung auf das ursprüngliche Problem. Sobald die Analogie ausgearbeitet wurde, untersuchen die Teilnehmer sämtliche Aspekte der Analogie und übertragen diese auf das ursprüngliche Problem.

Ergebnisverarbeitung

Die Teilnehmer erstellen eine Liste mit Aussagen zum ursprünglichen Problem, die aus der Analogie abgeleitet wurden. Um ein abschließendes Ergebnis zu erzielen, müssen die Teilnehmer die einzelnen Aussagen prüfen und den Wert diskutieren, den die Aussagen für das vorhandene Problem haben. Während dieser Diskussion können die Teilnehmer die übertragenen Aussagen weiter ausarbeiten, um deren Nutzen für das vorhandene Problem zu verbessern.

Ähnlich wie bei der Brainstorming-Technik, erfordert das Ergebnis der Analogietechnik zusätzlichen Aufwand, um die Ergebnisse zu analysieren und zu priorisieren.

Typische Arbeitsprodukte

Das Ergebnis der Analogietechnik ist eine detaillierte Beschreibung der Analogie, die Liste übertragener Aussagen für das ursprüngliche Problem und die finale Liste von bewerteten und detaillierten Aussagen für das ursprüngliche Problem.

Möglichkeiten

Eine gut gewählte Analogie erlaubt die Entwicklung innovativer Ideen zu einem vorhandenen Problem. Vor allem, wenn den Teilnehmern das bestehende Problem vorenthalten wird, kann die Analogietechnik unerwartete und unkonventionelle Ideen hervorbringen.

Ist das bestehende Problem zu schwierig und/oder vertraulich, kann die Analogietechnik dabei helfen, eine offene und konstruktive Arbeitsumgebung zu schaffen. Der anspruchsvolle Teil besteht natürlich in der Übertragung der Feststellungen auf das ursprüngliche Problem.

Herausforderungen

Der wichtigste Erfolgsfaktor für die Analogietechnik ist die Auswahl einer geeigneten Analogie. Wenn die Teilnehmer das ursprüngliche Problem kennen, ist es wichtig, das bestehende Problem aus der Diskussion über die Analogie herauszuhalten.

Varianten

Bei der Bisoziation [Koes1964], abgeleitet aus den Begriffen „bi“ (zwei völlig verschiedene Dinge) und „Assoziation“, assoziieren die Teilnehmer eines Kreativitätsworkshops Ideen zu einer vorhandenen Problemstellung mit etwas, das nichts mit dem Problem gemein zu haben scheint (z. B. fünf Bilder von berühmten Malern, einige besondere physische Objekte, kurze Videoclips von interessanten Tieren).

3.2.3 Prototyping

Was ist Prototyping?

Prototyp ist der Überbegriff für jedes intermediäre Arbeitsergebnisse, das zur Untersuchung von bestimmten Eigenschaften oder alternativen Lösungen für ein zu entwickelndes System mithilfe einer konkreten Erfahrung erstellt wird. In den meisten Fällen bezieht er sich auf Merkmale, die vorab nicht einfach verstanden, anhand von Modellen definiert oder in einer Dokumentation beschrieben werden können. Prototypen reichen von sehr einfachen Skizzen auf Papier über anklickbare Schnittstellensimulationen für physische Instanzen eines Geräts bis hin zu initialen Implementierungen einer Software; sie besitzen speziell ausgewählte – aber nicht alle – Merkmale des künftigen Systems und ermöglichen die Untersuchung bestimmter anderer Merkmale, die zu diesem Zeitpunkt noch unklar sind.

Was ein Arbeitsergebnis als Prototypen definiert, ist dessen Verwendung für die Untersuchung, die Erarbeitung, die Klärung, für Entwürfe, Tests, Validierungen usw. (siehe

[LiHB2003]). Kurz gesagt, besteht der Zweck der Verwendung von Prototypen darin, Systeme (oder bestimmte Teile davon) zu erleben, „... bevor sie Realität werden, sogar noch bevor wir ihren endgültigen Entwurf vorliegen haben, also viel weniger als ihre Implementierung“ [Buxton2007].

T. Z. Warfel [Warfel2009] beschreibt acht grundlegende Prinzipien für die Anwendung des Prototyping:

- Entwickeln Sie ein Verständnis für Ihre Benutzergruppe und deren Absichten.
- Planen Sie ein wenig – erstellen Sie für alles andere Prototypen.
- Definieren Sie Erwartungen.
- Erstellen Sie Skizzen.
- Es geht um einen Prototypen – nicht um die Mona Lisa.
- Wenn sich etwas nicht umsetzen lässt, simulieren Sie es.
- Erstellen Sie nur Prototypen, die Sie benötigen.
- Verringern Sie das Risiko – frühzeitig und häufig.

Die Rolle der Teilnehmer

Im Requirements Engineering werden Prototypen zur Anforderungsermittlung und Validierung verwendet. Bei der Ermittlung kommt den Teilnehmern in der Regel eine interne Rolle zu, z. B. als Requirements Engineer, Digital Designer, UX-Designer oder als Teil eines Entwicklungsteams.

Primäres Ziel der Teilnehmer ist es, das Problem nachzuvollziehen, geeignete Lösungen zu finden (in einem kreativen Prozess) und schließlich neue Anforderungen zu ermitteln oder vorhandene Anforderungen zu verfeinern. Bei der Validierung übernehmen Teilnehmer oftmals eine externe Rolle: Sie können etwa Benutzer oder andere Stakeholder außerhalb des Entwicklungsteams sein, welche die Implementierung bestimmter Anforderungen bewerten. Dies kann auch zu neuen Anforderungen und der Verfeinerung von vorhandenen Anforderungen führen.

In beiden Situationen können drei zentrale Rollen unterschieden werden:

- *Moderator*: Diese Rolle übernimmt gewöhnlich der Requirements Engineer, der über die Verwendung von Prototypen als Ermittlungstechnik entscheidet, der die Prototypen plant und managt, die Teilnehmer instruiert, Ergebnisse analysiert und Schlussfolgerungen zieht.
- *Entwickler*: Ein Prototyp muss entworfen, gebaut und getestet werden, bevor man ihn verwenden kann. Bei einfachen Prototypen wie Skizzen kann der Requirements Engineer diese Rolle übernehmen, bei komplexeren Prototypen ist gegebenenfalls die Unterstützung von Designern, Entwicklern oder Werkzeug-Experten nötig.
- *Beobachter*: Die genaue Untersuchung des Prototyps ist häufig eine Teamleistung, an der, je nach Größe und Komplexität, Requirements Engineers, Designer, Programmierer, Tester, Key User und andere Stakeholder involviert sein können. Um sich auf die Untersuchung der gewünschten Merkmale konzentrieren zu können, ist es wichtig, die Richtlinien oder Skripte des Moderators zu befolgen, aber auch einige Zeit für freies Erforschen einzuplanen.

Vorbereitung

Da Prototypen sehr unterschiedliche Formen annehmen können, ist für die Vorbereitung immer ein individueller Ansatz erforderlich [McEl2017]. Für eine einfache Skizze genügen möglicherweise Papier und Stift, für einen physischen Prototyp können hingegen ein 3-D-Drucker und eine umfangreiche Software erforderlich sein.

Ein Plan ist die eine – und wichtigste – Sache, die allen gemein ist: Der Requirements Engineer muss eine gründliche Vorstellung davon haben, welche Merkmale zu untersuchen sind und wie die Umsetzung auszusehen hat. Es muss auch klar sein, welche anderen, bereits etablierten Merkmale (die nicht untersucht werden müssen) vorhanden sein sollten und welche Aspekte man bewusst weglassen kann. Wenn Sie beispielsweise einen Prototypen für einen Webshop erstellen möchten, um die Performance zu analysieren, dann sollte die Funktionalität an sich funktionieren.

Es ist ein bewährtes Verfahren, die etablierten Merkmale des Prototyps zu testen, da sich Defekte in diesen Bereichen negativ auf die Untersuchung der Merkmale auswirken können, die nicht im Fokus stehen. Außerdem muss klar kommuniziert werden, welche Bestandteile oder Funktionalitäten nicht im Systemumfang enthalten sein sollen (d. h. im Prototyp nicht verfügbar oder funktionsfähig), vor allem, wenn externe Teilnehmer involviert sind.

Je nach Größe und Komplexität des Prototyps kann die Vorbereitung die Entwicklung von Richtlinien, Anweisungen, Skripten, Vorgehensweisen und Vorlagen für die Beobachter umfassen (siehe etwa die Vorbereitung eines Usability-Tests [UXQB2017]).

Anwendung

Die häufigsten Anwendungen für das Prototyping beinhalten [Warfel2009]:

- **Gemeinsame Kommunikation**

Das heißt, dass das Prototyping als ein Werkzeug für die Zusammenarbeit genutzt werden soll, als eine Art Lingua franca, also eine gemeinsame Sprache für alle Beteiligten. Das Ziel dabei ist es, ein gemeinsames Verständnis der Anforderungen zwischen den Geschäftsbereichen, Requirements Engineers, Designern, Entwicklern und Benutzern herbeizuführen. Dies beinhaltet häufig das einfache Skizzieren eines Entwurfs für das künftige System, wobei eine Reihe von Varianten eingeführt werden kann, aus denen man bei der weiteren Entwicklung auswählt.

- **Durcharbeiten eines Entwurfs**

Prototypen sind eine sehr gute Möglichkeit, um einen Entwurf tatsächlich durchzuarbeiten, ihn zu testen, festzustellen, welche Alternativen funktionieren, und die Details zu konkretisieren. Dies steht meist in Zusammenhang mit der Ausarbeitung und Verfeinerung von früheren allgemeinen Anforderungen und trägt zur Integrität und Konsistenz bei.

- **Internes Verkaufen Ihrer Idee**

„Zeigen ist besser als erzählen.“ Durch das Erstellen eines schnellen Prototyps mit unterschiedlichen Entwurfsoptionen ist es einfacher, Stakeholder von den Vorteilen Ihres Entwurfs zu überzeugen, die richtige Wahl zu treffen und Unterstützung dafür zu erhalten.

- **Usability-Tests**

Usability und andere Qualitätsmerkmale sind in Spezifikationen und Modellen offenkundig schwer zu erfassen, aber mithilfe eines Prototyps, selbst in einem frühen Projektstadium, leicht zu erkennen.

- **Einschätzen der technischen Realisierbarkeit und des Wertes**

Für jedem Entwurf eines neuen Systems bestehen die elementaren Fragen: „Kann es gebaut werden?“ und „Schafft es einen Mehrwert?“. Das Erstellen und Betreiben eines Prototyps kann Vertrauen erzeugen, da er positive Antworten auf diese Fragen liefert, oder – falls negativ – kostspielige Projektfehlschläge in einer frühen Phase verhindert.

Ergebnisverarbeitung

Sammeln Sie alle Erkenntnisse aus dem Prototyping, und entscheiden Sie gemeinsam im Team mit den zentralen Stakeholdern, was damit geschehen soll (z. B. bei einem Walkthrough). Für manches Feedback muss ein Review von einer größeren Gruppe von Stakeholdern erfolgen; einige Verbesserungsideen müssen von den technischen Mitarbeitern geprüft werden.

Nachfolgende Aktivitäten sind von der jeweiligen Projektsituation abhängig: beispielsweise die Entwicklung eines neuen Prototyps zur Untersuchung anderer Systemmerkmale, die Wahl einer bevorzugten Variante für die Weiterentwicklung oder die detaillierte Spezifikation der erkannten Anforderungen für einen ausgewählten Entwurf.

Typische Arbeitsprodukte

- der Prototyp
- Richtlinien, Anweisungen usw.
- Analyse und Dokumentation der Erkenntnisse und Schlussfolgerungen, beispielsweise Ideen, neue Anforderungen, geklärte vorhandene Anforderungen als künftige Referenz

Möglichkeiten

Das Prototyping ist ein vielseitiges Werkzeug, das in verschiedenen Phasen des Requirements Engineering anwendbar ist. Frühe Phasen profitieren von der Erschließung von Ideen, spätere Phasen von der Ausarbeitung und Verfeinerung von Anforderungen:

- Das Prototyping ermöglicht Requirements Engineers, frühzeitig im Projekt qualitatives Feedback von Stakeholdern zu erhalten und so später kostspielige Defekte zu vermeiden.
- Das Prototyping hilft Requirements Engineers dabei, über das Problem nachzudenken, einen Überblick über die Komplexität zu behalten und diese zu reduzieren und sich dadurch auf die Bereitstellung eines Mehrwerts für den Kunden zu konzentrieren.
- Das Prototyping kann als iterativer, partizipativer, gemeinsamer Designansatz genutzt werden, der dem Entwicklungsteam und den Stakeholdern die gemeinsame Ausarbeitung von Lösungen ermöglicht.

- Das Prototyping passt perfekt zu modernen Entwicklungsansätzen wie Agilität, DevOps, Lean und Design Thinking.

Herausforderungen

Abhängig von der Art des gewählten Prototyps, kann dessen Entwicklung äußerst aufwendig sein. Für ein realistisches, anklickbares funktionales Modell einer Benutzungsoberfläche (horizontaler Prototyp) oder einen physischen Prototyp können die Erstellung des Entwurfs, Entwicklung und Tests mehrere Tage erfordern. Die Herausforderung besteht darin, den Aufwand für die Erstellung des Prototyps mit dem aus den gewonnenen Erkenntnissen erwarteten Nutzen im Gleichgewicht zu halten. Denken Sie an die Aussage „Ein Prototyp ist nicht die Mona Lisa“, und widerstehen Sie der Versuchung, den Prototyp zu perfektionieren.

Eine weitere Herausforderung (vor allem bei komplexen Prototypen) ist, dass der Prototyp von einigen Stakeholdern als finales Produkt betrachtet werden könnte. Ein solches Missverständnis kann zu unrealistischen Erwartungen an den Projektzeitplan führen, da Stakeholder oft die Zeit unterschätzen, die für die Umwandlung eines Prototyps in eine produktionsreife Software nötig ist.

Varianten

Prototypen können zwar stark variieren, sowohl hinsichtlich der Intention als auch mit Blick auf die Funktionalität, aber es sind in der Regel drei Hauptmerkmale erkennbar: Genauigkeit (engl. fidelity), Lebenszyklus und Auffassung.

Bei Fidelity, oder Genauigkeit, geht es um die Kongruenz des Prototyps mit dem Zielprodukt:

- Ein *Low-Fidelity-Prototyp (LoFi-Prototyp)* erinnert gerade so viel an das künftige System, als er eine gewisse relevante Erfahrung mit den geplanten Eigenschaften ermöglicht, z. B. Skizzen für Bildschirme in einem Storyboard.
- Ein *High-Fidelity-Prototyp (HiFi-Prototyp)* ahmt die externe Benutzungsoberfläche des künftigen Systems in hohem Maße nach, so dass man auf den ersten Blick kaum einen Unterschied feststellen kann (auch hier gilt wieder: soweit es die geplanten Eigenschaften betrifft). Intern, also unter der Oberfläche, kann das System hingegen weit von seiner Fertigstellung entfernt sein.

Der Lebenszyklus ist die Beziehung eines Prototyps zum Zielprodukt:

- Ein *explorativer Prototyp* wird nur zu Untersuchungs- und Bewertungszwecken erstellt. Er wird auch als „*Wergwerf-Prototyp*“ bezeichnet, weil er nach der Verwendung verworfen wird. Diese Art von Prototyp ist häufig in der Massenproduktion anzutreffen.
- Ein *evolutionärer Prototyp* wird kontinuierlich ausgestaltet, erweitert, verbessert und Refactorings unterzogen, bis er zum finalen Produkt wird. Im Laufe der iterativen Entwicklung wird er zunächst für Untersuchungen verwendet und dann basierend auf den Feststellungen einem Refactoring unterzogen. Auf diese Art des Prototyping trifft man häufig bei der Einmalfertigung. Man könnte argumentieren, dass die agile/Scrum-Entwicklung gewöhnlich eine Art von evolutionärem Prototyping ist.

Die Auffassung, d. h. Prototypen können während der Entwicklung von unterschiedlichen Standpunkten aus betrachtet werden:

- Das *Feedback* kann der wesentliche Zweck sein, z. B. wenn ein Requirements Engineer einen bestimmten Aspekt der Lösung, etwa die Benutzungsoberfläche, gemeinsam mit einer Gruppe von Stakeholdern bewerten möchte.
- Das *Design/der Entwurf* können an erster Stelle stehen. Das ist der Fall, wenn ein Prototyp erstellt wird, um für ein bestimmtes Problem unterschiedliche Lösungen zu betrachten und miteinander zu vergleichen, wie es häufig in Industriedesign-Projekten vorkommt.

Für ein tieferes Verständnis von Prototypen, nicht nur im Zusammenhang mit der Anforderungsermittlung, sondern auch im Zusammenhang mit Digital Design, verweisen wir auf [DDP2021].

3.2.4 Verwendung von Szenarien und Storyboards

Was ist ein Szenario?

Der Begriff „Szenario“ leitet sich aus dem lateinischen Wort „scaenarium“ ab, das „Ort, wo die Bühne errichtet wird“ bedeutet. Heute wird der Begriff „Szenario“ u. a. für einen Entwurf oder die Zusammenfassung eines Stücks verwendet⁷. In derselben Bedeutung können Szenarien für die Erstellung des Entwurfs einer Systemverwendung genutzt werden [Carrol2003].

Szenarien können in schriftlicher oder visueller Form dokumentiert werden. Die visuelle Form eines Szenarios bezeichnet man als Storyboard. Ein Storyboard ist typischerweise die Darstellung einer Abfolge von Bildern einer Szene oder einer Handlung (z. B. für einen Film, eine Fernsehsendung oder einen Werbespot)⁸. Ein Storyboard kann gleichermaßen zur Beschreibung eines Handlungsablaufs für ein System verwendet werden.

Vorbereitung

Die Vorbereitung der Szenariotechnik besteht aus einer Entscheidung in Bezug auf die Szenariogeschichte(n) und aus dem Sammeln von Materialien und Informationen für die Entwicklung der Szenarien. Die Story oder Geschichte des Szenarios bezieht sich auf die konkrete Handlung oder die Aktionen, die beschrieben werden sollen. Denken Sie daran, dass Szenarien aus dem Bereich des Theaters stammen; einem Theaterstück liegt immer eine Story zugrunde. Die Story kann aus den Zielen oder Problemen abgeleitet werden, die erreicht bzw. gelöst werden sollen. Seien Sie sich dessen bewusst, dass das Definieren des Anwendungsbereichs der Story eine Herausforderung darstellt.

⁷ Siehe z. B. <https://www.duden.de/rechtschreibung/Szenario>

⁸ Siehe z. B. <https://www.duden.de/rechtschreibung/Storyboard>

Wenn eine Story zu umfangreich ist (z. B. ein durchgängiger Geschäftsprozess in einem Onlineshop), kann das Szenario zu lang werden. Lange Szenarien sind wie weitschweifige Bücher: Sie lassen sich nicht gut lesen.

Umgekehrt ist es auch ein Risiko, wenn der Umfang zu gering ist, da das Szenario dann zu detailliert wird und die Gefahr besteht, dass man sich in Einzelheiten verliert (beispielsweise das Hinzufügen eines Produkts zu einem Warenkorb könnte für ein Szenario zu eng gefasst sein). Als Faustregel gilt: Man sollte versuchen, einen Anwendungsbereich zu identifizieren, der einen in sich geschlossenen Teil der ganzen Story darstellt und der zu einem erkennbaren Ergebnis führt (z. B. die Suche nach Produkten in einem Onlineshop). Im Allgemeinen sollte sich ein Szenario nur auf einen bestimmten Anwendungsbereich konzentrieren. Falls man dasselbe Szenario für verschiedene Anwendungsbereiche beschreiben möchte, sollte jede Kombination aus Szenario und Anwendungsbereich als unabhängige Ermittlungsaktivität behandelt werden (siehe Kapitel 0).

Während der Vorbereitung werden in der Regel folgende Informationen gesammelt: Informationen zu typischen Benutzern, typischen Orten, an welchen das System verwendet wird und zu wichtigen Ereignissen oder Situationen, die innerhalb des Szenarios berücksichtigt werden sollten.

Darüber hinaus sollte für das Szenario ein Dokumentationsformat gewählt werden (siehe typische Arbeitsergebnisse für eine Liste von Formaten).

Anwendung

Die Anwendung der Szenariotechnik ist einfach: Beginnen Sie mit der Entwicklung des Szenarios. Die Entwicklung kann von einer Gruppe oder von Einzelpersonen ausgeführt werden. Wenn der Szenariokontext verschiedene Kompetenzen erfordert, ist jedoch eine Gruppenarbeit ratsam (z. B. bei komplexen Geschäftsprozessen).

Bei der Entwicklung eines Szenarios sollte man immer unterschiedliche Alternativen erstellen. Auf diese Weise können alternative Versionen untersucht werden, was die Chance auf ein optimales Endergebnis erhöht. Wird das Szenario als Zwischenergebnis genutzt (siehe Ergebnisverarbeitung), dann ist es nicht notwendig, die entwickelten Versionen in einer finalen Version zu konsolidieren. Präsentiert man Stakeholdern alternative Versionen eines Szenarios, so belebt das die Diskussion, und man kann die Meinung der Stakeholder zu den Alternativen im Entwicklungsprozess berücksichtigen.

Ergebnisverarbeitung

Szenarien können entweder in eine Anforderungsspezifikation oder ein Zwischenergebnis einfließen. Im ersten Fall sollte das Szenario in Einklang mit den Projektstandards dokumentiert werden. In letzterem Fall sind weitere Aktivitäten erforderlich, um Anforderungen aus dem Szenario abzuleiten. Dieser Schritt ist eine nachfolgende Technik, die nicht Bestandteil der Szenariotechnik ist.

Typische Arbeitsprodukte

Szenariobeschreibungen erfolgen typischerweise in schriftlicher Form. Die einfachste Form ist ein Prosatext:

John Doe wartet an der Haltestelle „Marktstraße“. Es ist kalt und regnerisch und er möchte schnell nach Hause. Leider scheint der Bus Verspätung zu haben. John möchte die Ankunftszeit des Busses nachsehen. Er nimmt sein Smartphone zur Hand, öffnet die App für den öffentlichen Nahverkehr und wählt die Funktion „Abfahrtszeiten von meiner Haltestelle“ aus. Die App nutzt Smartphone-Funktionalitäten (z. B. GPS), um die Haltestelle „Marktstraße“ als Johns aktuellen Standort zu ermitteln und zeigt den aktuellen Fahrplan einschließlich der Verspätung von elf Minuten an.

Die Szenariobeschreibung mit dedizierten Schritten stellt eine strukturiertere Herangehensweise dar:

1. John Doe wartet an der Haltestelle „Marktstraße“. Es ist kalt und regnerisch und er möchte schnell nach Hause. Leider scheint der Bus Verspätung zu haben. John möchte die Ankunftszeit des Busses nachsehen.
2. Der Benutzer nimmt sein Smartphone zur Hand und öffnet die App für den öffentlichen Nahverkehr.
3. Die App enthält ein Benutzermenü, aus dem der Nutzer die Funktion „Abfahrtszeiten von meiner Haltestelle“ auswählt.
4. Die App nutzt die Smartphone-Funktionalität (z. B. GPS), um die Haltestelle „Marktstraße“ als Johns aktuellen Standort zu ermitteln.
5. Die App zeigt den aktuellen Fahrplan und die Verspätung von elf Minuten an.

Der Unterschied zwischen dem kontinuierlichen Szenario und dem strukturierten Szenario ist die Aufgliederung in einzelne Schritte. Jeder Schritt bezieht sich in der Regel auf eine Interaktion zwischen dem Benutzer und dem System oder auf eine Aktivität des Nutzers im Kontext des Systems (Schritt 1 in dem Beispiel). Szenarien sind immer spezifisch in Bezug auf Personen (John Doe), Mengen (elf Minuten), Orte (Haltestelle Marktstraße) und relevante Kontexte (kalt und regnerisch). Das trägt dazu bei, das Szenario mit Leben zu füllen: eine echte Story in einem realen Kontext. Das ist wichtig, um Stakeholdern zu vermitteln, worum es bei der Lösung geht, und es hilft ihnen dabei, über ganz spezifische Probleme und Themen zu sprechen.

Hinweis:

Achten Sie darauf, Szenarien nicht mit Use Cases zu verwechseln [Cockburn2001]. Eine Use Case Spezifikation ist eine Technik, mit der Sie die allgemeine Interaktion zwischen einem Akteur und dem System dokumentieren. Szenarien sind hingegen eher wie Testfälle (sie können vielmehr die Basis für Testfälle bilden). Man kann sie als eine spezifische Instanz eines Durchgangs für einen Use Case betrachten.

Storyboards wurden bereits in der Einführung erwähnt; es handelt sich dabei um eine visuelle Darstellungsform für Szenarien. Storyboards sind nützlich, wenn ein Szenario zahlreiche Handlungsschritte enthält und der Kontext leichter visualisiert als beschrieben werden kann.

Möglichkeiten

Szenarien stellen eine gute und schlanke Technik für eine frühzeitige Erarbeitung von Ideen im Hinblick auf Prozesse und Aktivitäten dar. Sie können dazu verwendet werden, alternative

Möglichkeiten der Realisierung eines Prozesses in einem System zu diskutieren und zu untersuchen. Aufgrund ihrer schlanken Struktur sind sie einfach zu entwickeln und lassen sich schnell ändern. Als Faustregel gilt, dass es in jedem Entwicklungsprojekt ein gemeinsames – und explizites – Verständnis der Szenarien des zu entwickelnden Systems geben sollte. Die Szenariotechnik ist daher ein guter Kandidat für eine frühzeitige Ermittlung.

Herausforderungen

Für das Erstellen eines guten Szenarios sind gute Schreibfertigkeiten erforderlich (z. B. für das Strukturieren oder eine ausdrucksstarke Beschreibung der Story). Ohne diese Fähigkeiten geraten Szenarien häufig langweilig, und gute Ideen können aufgrund einer schwachen Präsentation verlorengehen. Ein gewandter Verfasser vermeidet außerdem detaillierte Szenarien, die zu viele unnötige Informationen für den Leser enthalten.

Über diese Schreibfertigkeiten hinaus, sind für die Anwendung von Storyboards grundlegende Zeichenkenntnisse sehr hilfreich, damit man zu den einzelnen Schritten aussagekräftige Bilder entwerfen kann. Dieses grundlegende Niveau ist nicht schwer zu erreichen: Ein einfaches Storyboard besteht aus Strichmännchen und einfachen Skizzen. Als schwierigere Herausforderung könnten sich die Hemmungen erweisen, seine Skizzen anderen Personen zu zeigen!

Varianten

Szenarien werden typischerweise positiv beschrieben, d. h. das Szenario endet mit einem positiven Ergebnis und der Benutzer erreicht, was er erreichen möchte. Eine andere gute Variante für Szenarien ist es, wenn man sich auf das negative Ergebnis konzentriert und beschreibt, was passieren könnte, wenn das Szenario nicht positiv endet. Im oben beschriebenen Bus-Szenario könnte beispielsweise das GPS nicht funktionieren. Was präsentiert die App in diesem Fall dem Benutzer? Wenn man ein Szenario auf diese Weise betrachtet, erhält man häufig viele zusätzliche Erkenntnisse und neue Ideen für das zu entwickelnde System.

Eine dritte Szenariovariante bilden Fälle von missbräuchlicher Verwendung [SiOp2005]. Bei einer solchen Zweckentfremdung geht es um einen beabsichtigten Fehlgebrauch von Systemfunktionalitäten, um dem Benutzer oder anderen Stakeholdern Schaden zuzufügen. Diese sind besonders für die Ermittlung von Anforderungen in Bezug auf Systemsicherheit nützlich.

Anstatt Storyboards zu zeichnen, könnte das Team Szenen schauspielerisch darstellen und Fotos aufnehmen und diese anschließend mit einem Zeichenprogramm anhand von Filtern gezielt verändern.

3.3 Denkwerkzeuge

In den vorherigen Abschnitten wurden einige allgemeine Techniken erläutert, die sich bei der Ermittlung von Anforderungen als erfolgreich erwiesen haben. Sie beschreiben Wege, um Informationen zu sammeln und Arbeitsergebnisse für die Dokumentation und Kommunikation von Anforderungen zu produzieren.

Dieser Abschnitt behandelt bestimmte unterstützende Techniken. Diese Techniken werden nicht für sich alleine verwendet, sondern in Verbindung mit den anderen Ermittlungstechniken. Wir bezeichnen sie als *Denkwerkzeuge*, weil damit das Ziel verfolgt wird, eine Denkweise zu stimulieren oder für eine Denkweise zu sorgen, die zum Erfolg der Ermittlung beiträgt.

3.3.1 In Abstraktionsebenen denken

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 detaillierteren Ebenen fortzufahren.

Um Abstraktionsebenen als ein Denkwerkzeug zu verstehen, muss man den Begriff „Abstraktion“ verstehen. Gemäß Cambridge Dictionary⁹ ist eine Abstraktion „die Situation, in der ein Thema sehr allgemein ist und nicht auf realen Situationen beruht“ (*“the situation in which a subject is very general and not based on real situations”*).

⁹ <https://dictionary.cambridge.org/dictionary/english/abstraction>, Zuletzt besucht im August 2025

Abstraktionsebenen

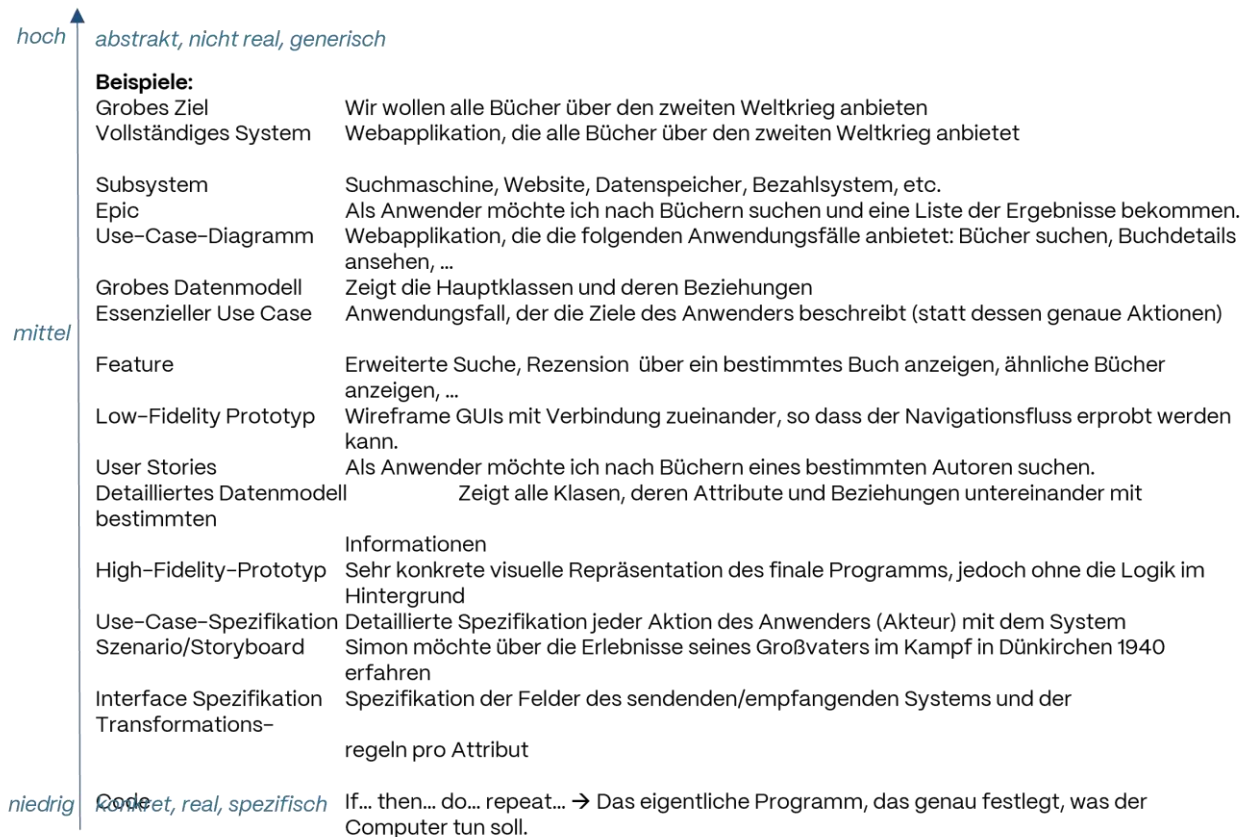


Abbildung 21: Darstellung des Konzepts von Abstraktionsebenen, illustriert mit aufschlussreichen Beispielen

In der Informatik wird Abstraktion dadurch erreicht, dass Informationen ausgeblendet werden [CoSh2007]. Programmiersprachen und APIs sind gute Beispiele dafür. Jede Anweisung in einer Programmiersprache und jede Methode in einer API bietet eine bestimmte Funktionalität, die sich hinter dem Namen der Anweisung oder der Methode verbirgt. Wenn jemand Näheres darüber erfahren möchte, kann er die verborgenen Informationen durch Lesen des Codes der Funktion oder der Methode entschlüsseln. Ziel einer Abstraktion ist es, die Komplexität zu verringern, indem man detaillierte Informationen hinter einem einzigen, einfacheren Konstrukt verbirgt. Nachfolgend beschreiben wir zwei Beispiele für die Anwendung von Abstraktionsebenen im Requirements Engineering.

Drei wesentliche Abstraktionsebenen im Requirements Engineering

Wir haben im Requirements Engineering eine natürliche intellektuelle Grenze, die zur Verringerung von Komplexität verwendet werden kann: Die Grenze zwischen System und Kontext (Systemgrenze, siehe [IREB2020]). Der Kontext hat mit allem zu tun, was sichtbar und für den Benutzer oder ein anderes System erfahrbar ist. Dies schließt die Benutzungsoberfläche und technische Schnittstellen ein. Wenn wir die Anforderungen eines Systems auf der Abstraktionsebene diskutieren, ist es nicht notwendig, über technische Details zu sprechen, durch welche sich diese Anforderungen realisieren lassen. Requirements Engineers tun dies beispielsweise durch ein Use Case Diagramm oder eine

User Story Map: Sie visualisieren die vom System für die Akteure bereitgestellte Funktionalität im Kontext und lassen dabei alle Implementierungsdetails außer Acht.

Das System an sich lässt sich weiter in ein logisches und ein technisches System unterteilen. Das logische System ist eine idealisierte Beschreibung und enthält logische Datenstrukturen, Funktionalitäten und Verhaltensbeschreibungen bestimmter Elemente auf der Systemkontextebene (z. B. die Spezifikation der zuvor genannten Use Cases). Das technische System bezieht sich auf die Realisierung in Bezug auf Hardware, technische Datenstrukturen und Softwarekomponenten (z. B. Datenbanken und Frameworks). Nehmen Sie als ein Beispiel das Betriebssystem eines Computers: Das ist die API zu den technischen Details der Hardware, z. B. wie spezielle Speicherelemente adressiert werden oder wie der Prozessor einzelne Operationen ausführt, gleichzeitig oder auf andere Weise.

Die Trennung zwischen Systemkontext, logischem und technischem System kann für Folgendes verwendet werden:

- Zur Strukturierung von Anforderungsspezifikationen, um komplexe Inhalte leichter zugänglich zu machen;

Eine Interaktion zwischen einem Benutzer und einem System wird beispielsweise auf der Kontextebene beschrieben, während die detaillierte Datenstruktur und die funktionalen Anforderungen in einem späteren Teil der Spezifikation auf der Systemebene festgelegt werden. Der Leser einer Spezifikation kann dann entscheiden, ob er die Details (z. B. eines Datenmodells) lesen möchte, oder ob er sie zugunsten des Gesamtverständnisses überspringt.

- Zur Strukturierung des Entwicklungsprozesses einer Spezifikation;

Ein Extrem wäre beispielsweise, einen Ansatz zu verfolgen, bei dem man zunächst in die Breite geht, und zuerst so viel wie möglich auf der Kontextebene versteht, bevor man die Systemebene detailliert betrachtet. Ein alternativer extremer Ansatz wäre, zunächst in die Tiefe zu gehen und einen Aspekt auf der Kontextebene mit allen nötigen Details auf der Systemebene auszuarbeiten, bevor man mit dem nächsten Element auf der Kontextebene fortfährt. Unter praktischen Gesichtspunkten ist eine Mischung aus zwei Ansätzen ratsam. Erfolgskritische Aspekte eines System sollten etwa anhand eines Ansatzes ausgearbeitet werden, der zuerst in die Tiefe geht, während man bei einfachen Aspekten oder Elementen, von denen man bereits ein gutes Verständnis hat, zuerst in die Breite gehen kann, da die Details sowieso klar sind und das Risiko, wichtige Details zu vergessen, gering ist.

3.3.2 In Problemen und Zielen denken

Die Entwicklung eines Systems beruht häufig darauf, dass ein Kunde ein bestimmtes Problem hat. In anderen Fällen wird die Entwicklung durch den Wunsch eines Kunden ausgelöst, ein bestimmtes Ziel zu erreichen. In den meisten Fällen besteht bei Kunden jedoch keine ausgeprägte Klarheit, Sicherheit oder Offenheit in Bezug auf ihre wahren Probleme

und Ziele. Ein Kunde schlägt mitunter eine Lösung vor, ohne überhaupt erklären zu können, welches Problem damit gelöst oder welches Ziel realisiert werden sollte!

Daher ist es unverzichtbar, dass der Requirements Engineer nicht voraussetzt, dass ein einzelnes Problem oder Ziel der Ausgangspunkt für die Entwicklung ist. Um ein Gesamtbild der zusammenhängenden Probleme und Ziele hervorzubringen, muss die vorliegende Situation gründlich analysiert werden, bevor eine gangbare Lösung konzipiert werden kann (siehe [LoSL2017]). Zunächst einmal muss Klarheit über die Definitionen herrschen.

Ein *Problem* ist ein Aspekt im Kontext des Stakeholders, der aktuell als negativ wahrgenommen wird. Eine erwartete negative Erfahrung im Kontext des Stakeholders, die in der Zukunft liegt, wird als *Risiko* bezeichnet (z. B. ein potenzielles künftiges Problem). Oftmals wird ein bestimmter Zustand in einem Kontext als Problem wahrgenommen, da er *verhindert*, dass der Stakeholder etwas Wünschenswertes tut oder ein Ziel erreicht.

Ein *Ziel* ist ein erwarteter positiver Aspekt im Kontext eines Stakeholders, der in der Zukunft liegt. Ein bestimmter künftiger Zustand in einem Kontext wird häufig als ein Ziel wahrgenommen, wenn dieser Zustand den Stakeholder *in die Lage versetzt*, etwas Wünschenswertes zu tun.

Es ist wichtig, sich zu vergegenwärtigen, dass Probleme und Ziele *mentale Konstrukte* sind: Sie existieren nicht in der Realität, sondern nur in den Köpfen der Stakeholder. Daher können Sie nur durch eine gezielte Ermittlung festgestellt werden. Folglich kann ein bestimmter (zukünftiger) Zustand eines Kontextes für einen Stakeholder ein Ziel darstellen, während er für einen anderen ein potenzielles Problem bedeutet.

Probleme und Ziele sind immer miteinander verbunden. Ein Problem impliziert das Ziel, den negativen Zustand in eine positive Richtung zu lenken. Ein Ziel wird nur als solches anerkannt, wenn es im gegenwärtigen Zustand irgendetwas gibt, das verhindert, dass es bereits stattfindet.

Probleme und Ziele sind auch durch ein anderes geistiges Konstrukt miteinander verbunden: die *Lösung*.

Eine Lösung ist die Roadmap für eine Intervention im Kontext des Stakeholders: Sie beschreibt die Art und Weise, auf die der aktuelle Zustand in der Gegenwart in einen gewünschten Zustand in der Zukunft verändert werden kann. Lösungen werden in einem kreativen Designprozess entwickelt, der bei den ermittelten Problemen und Zielen beginnt.

Explizite Probleme und Ziele können Ausprägungen impliziter Probleme oder Ziele auf allgemeinerer oder detaillierterer Ebene sein. Sie treten jedoch niemals für sich alleine auf: Sie sind immer Teil einer großen „Familie“ aus „Eltern“ und „Kindern“. Um ein vollständiges Bild zu erhalten, müssen auch die „Geschwister“ in dieser Familie festgestellt werden.

Den Eltern eines bestimmten Problems kommt man auf die Spur, indem man nach Ursachen sucht. *Was verursacht dieses Problem?*

Die Eltern eines Ziels findet man, indem man das Verhalten analysiert, das vorherrscht, wenn das Ziel einmal erreicht ist. *Warum hat der Stakeholder dieses Ziel?*

Die Kinder in der Familie (also die Probleme und Ziele auf detaillierterer Ebene) findet man durch Lösungen. Mit jeder Maßnahme einer realisierbaren Lösung wird auf einer detaillierteren Ebene ein neues Ziel (und ein Problem) für jemanden festgelegt, der für die Umsetzung verantwortlich ist und der vor der Herausforderung steht, dies möglichst effektiv zu tun.

Aus dem folgenden Klassenmodell geht die Beziehung zwischen diesen Konzepten hervor. Ein Problem impliziert ein Ziel und wird von Problemen auf allgemeinerer Ebene verursacht. Ein Ziel ermöglicht Ziele auf höherer Ebene. Ein Problem kann durch eine Lösung behoben werden, mit der das betreffende Ziel erreicht wird. Mit einer Lösung werden bestimmte auszuführende Maßnahmen festgelegt. Mit der Implementierung jeder Maßnahme wird wiederum mindestens ein detaillierteres Ziel festgelegt.

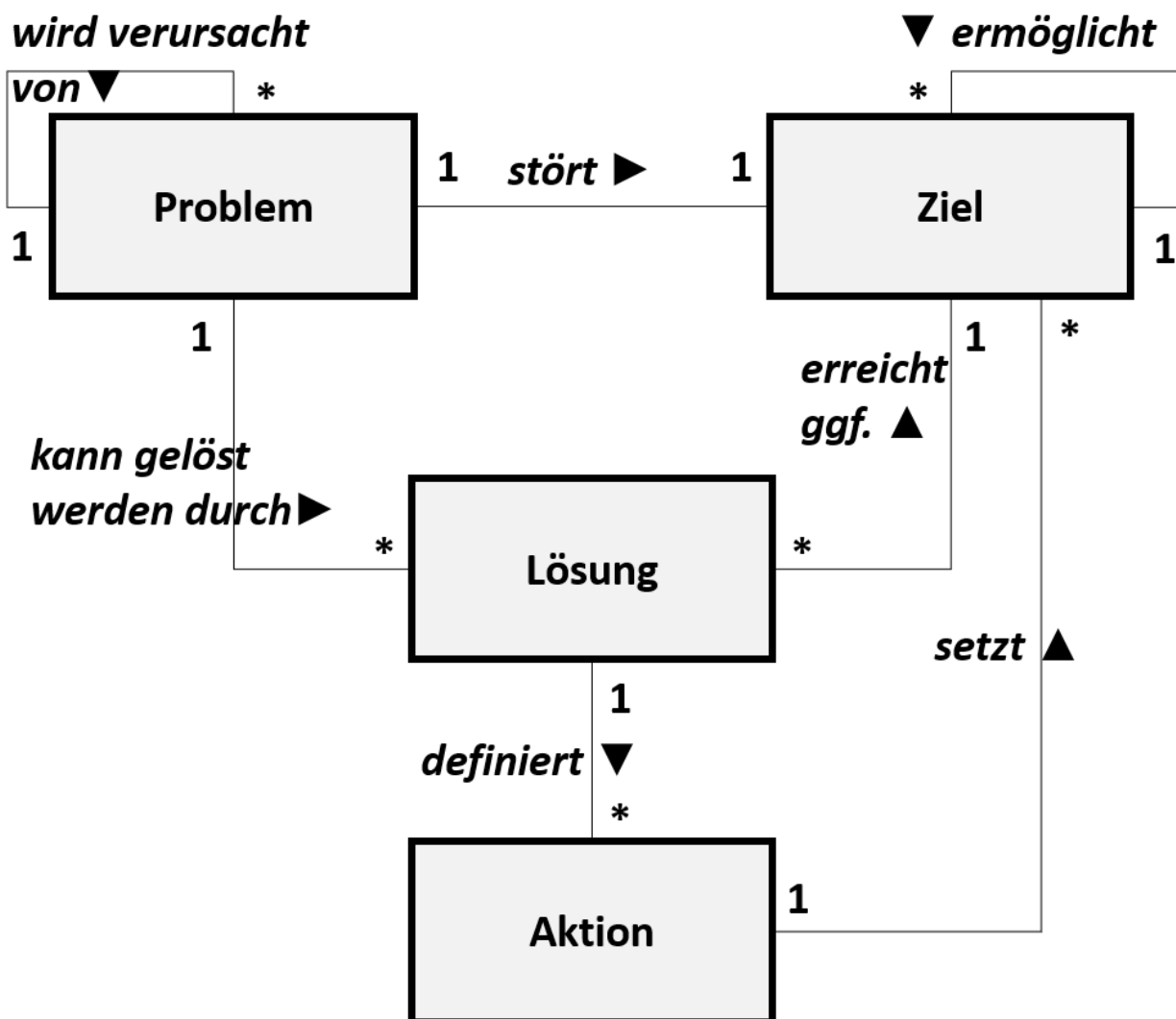


Abbildung 22: Die Beziehung zwischen Problemen, Zielen und Lösungen

Das Denken in Problemen und Zielen ist an sich keine Technik. Es handelt sich um eine Denkweise, die den Requirements Engineer bestärkt, sich noch näher mit dem Kontext des Stakeholders auseinanderzusetzen, bevor er eine scheinbar offensichtliche, aber vielleicht

inadäquate Lösung aufgreift. Diese Denkweise ist für alle Arten von Ermittlungstechniken relevant, wenn man auf ein Problem oder ein Ziel trifft.

Versuchen Sie bei einem Problem Folgendes herauszufinden:

- Handelt es sich um ein Problem im aktuellen Kontext (welches Stakeholders) oder ist es die Erwartung eines zukünftigen Problems (überlegen Sie sich in diesem Fall Wege, um zu verhindern, dass diese Zukunft zur Realität wird)?
- Welche gegenwärtigen oder vergangenen Situationen haben dieses Problem verursacht (dadurch können Sie Hinweise auf mögliche Lösungen erhalten)?
- Welche zukünftigen Maßnahmen werden durch das Problem (das damit verbundene Ziel) verhindert? Was würde welcher Stakeholder tun (tun können), wenn das Problem nicht existieren würde?

Stellen Sie in Bezug auf ein Ziel folgende Überlegungen an:

- Welche Situationen in der Gegenwart (oder Erwartungen an die Zukunft) verhindern, dass das Ziel erreicht (das damit verbundene Problem gelöst) wird? Was geschieht, wenn keine Maßnahmen ergriffen werden?
- Welche zukünftigen Maßnahmen sind möglich, wenn das Ziel erreicht ist? Wer erzielt welchen Vorteil (Wert/Nutzen)?

In Bezug auf Lösungen:

- Seien Sie vorsichtig, wenn ein Kunde direkt eine bestimmte Lösung vorschlägt. Stellen Sie sicher, dass Sie alle damit verbundenen Probleme und Ziele wirklich nachvollziehen können. Prüfen Sie nach, ob die vorgeschlagene Lösung das Problem behebt und das Ziel dadurch realisiert wird.
- Berücksichtigen Sie den Wert: die Balance zwischen dem erwarteten Nutzen einer Lösung, den (Gesamt-) Kosten für deren Implementierung und das Risiko eines Fehlschlags.

Durch das Analysieren des gesamten Umfelds der damit verbundenen Probleme und Ziele gelangen Sie leichter zu der Lösung, die den höchsten Gesamtwert erbringt. Eine von einem Stakeholder genannte Lösung oder ein Ziel hat möglicherweise einen Wert für diese Person, für das Unternehmen insgesamt kann sie aber unter Umständen ein Risiko bedeuten. Da das Ziel eines Stakeholders für einen anderen Stakeholder ein Problem darstellen kann, hilft Ihnen das Denken in Problemen und Zielen auch dabei, Anforderungskonflikte zu identifizieren und zu lösen.

Bei der Diskussion von Problemen und Zielen erzählen Stakeholder häufig nur einen Teil der ganzen Story. Da Probleme und Ziele mentale Konstrukte sind, die lediglich in den Köpfen bestimmter Stakeholder existieren, sind sie naturgemäß subjektiv – was sich jedoch schwer zeigen lässt. Wenn Sie als Requirements Engineer Schwierigkeiten damit haben, die Beschaffenheit eines Problems oder den Wert eines Ziels zu verstehen, so kann dies an verdeckten, subjektiven Komponenten liegen. Fragen nach dem *Warum* können Ihnen dabei helfen, diese zu klären.

3.3.3 Vermeidung von Transformationseffekten

Aufgrund unterschiedlicher Wissensstände, verschiedener kultureller und sozialer Hintergründe oder beruflicher Erfahrung, können Stakeholder dieselben Informationen auf unterschiedliche Weise wahrnehmen. Die Realität oder eine angeforderte Systemfunktionalität werden von der persönlichen Wahrnehmung einer Person gefiltert, in persönliches Wissen verwandelt und später in einer mehr oder weniger gut ausgestalteten Aussage formuliert. Die Aussage stellt das Wissen und im Idealfall ein hohes Maß an relevanten Informationen über die Realität dar. Man unterscheidet zwei Arten der Transformation [BaGr2005] [Rupp et al.2014]:

Wahrnehmungstransformationen: Sie treten auf, weil jede Person die Realität auf andere Weise wahrnimmt und sich ein individuelles Bild davon macht.

Darstellungstransformationen: Sie treten aufgrund einer Änderung auf, die stattfindet, sobald eine Person ihr Wissen in natürlicher Sprache äußert.

Diese Transformationsprozesse können zu einem Verlust und der Verzerrung von Informationen führen, welche der Requirements Engineer aufdecken muss, um vollständige Anforderungen von hoher Qualität zu dokumentieren. Abbildung 23 illustriert diese Transformationsprozesse.

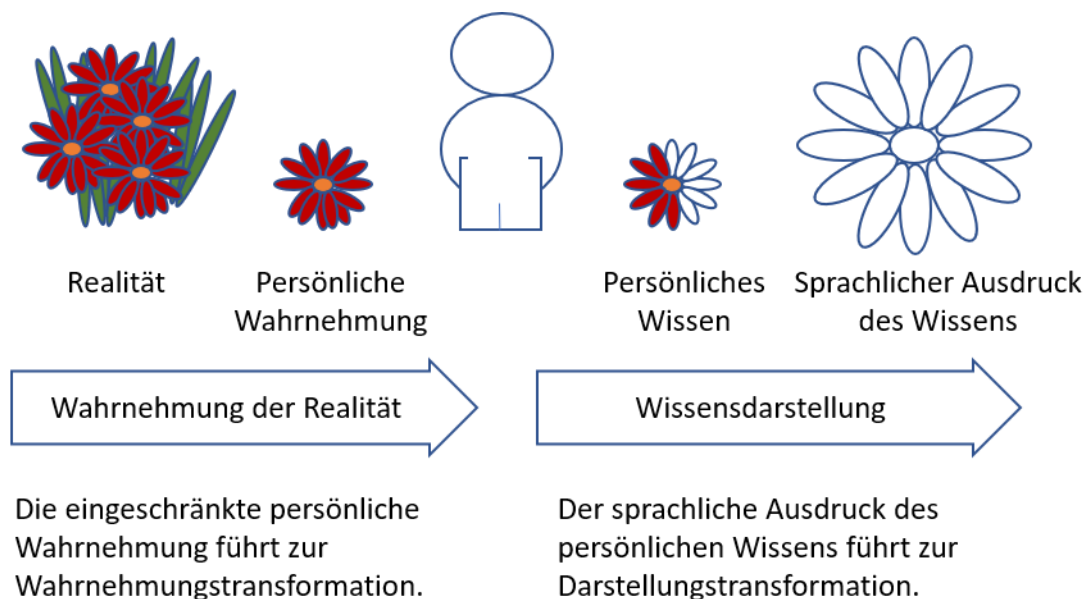


Abbildung 23: Transformationsprozesse

Um herauszufinden, welche Informationen verloren gegangen sind oder verzerrt wurden, muss der Requirements Engineer die verschiedenen Kategorien von Transformationsprozessen kennen.

Die **Tilgung** ist ein Prozess, bei dem wir bestimmten Dimensionen unserer Erfahrung selektiv Aufmerksamkeit widmen und andere ausblenden. Die Tilgung reduziert die Welt auf ein Maß, bei dem wir das Gefühl haben, damit umgehen zu können [BaGr2005]. Tilgung ist ein Indikator dafür, dass der Requirements Engineer fehlende Informationen ermitteln muss.

Beispiel:

Ein Softwareentwickler berücksichtigt in erster Linie die logischen Informationen (Kundendaten) einer Schnittstelle. Ihm sind möglicherweise nicht alle benötigten physischen Informationen (Material des Steckers) bekannt.

Generalisierung ist der Prozess, durch den Elemente des Modells einer Person aus deren ursprünglicher Erfahrung losgelöst werden und die gesamte Kategorie darstellen, von der die Erfahrung ein Beispiel ist [BaGr2005]. Generalisierung ist ein Indikator dafür, dass der Requirements Engineer die fehlenden Informationen ermitteln muss, um zu entscheiden, welche Informationen generalisiert werden können und welche tatsächlich nur unter bestimmten Umständen gelten.

Beispiel:

Für einen Stakeholder enthält eine Adresse eines Kunden immer den Straßennamen und die Hausnummer. In einem anderen Land sind jedoch Hausnummern gar nicht üblich.

Durch **Verzerrung** können wir ein Ereignis oder ein Vorkommen in ein Konzept aus Vorwissen einpassen. Sie verändert unsere Interpretation von Ereignissen, damit sie in unser bestehendes Verständnis passen [GrBa2017]. Menschen ersetzen Teile der Realität durch ihr persönliches Glaubenssystem. Verzerrung ist also ein Indikator dafür, dass der Requirements Engineer die fehlenden Fakten ermitteln muss, um zu entscheiden, welche der Aussagen die Realität darstellen.

Beispiel:

Ein Stakeholder ist möglicherweise davon überzeugt, dass zwei Systeme auf Basis derselben Datenbank funktionieren, weil er festgestellt hat, dass aktualisierte Daten sofort in beiden Systemen verfügbar sind. Tatsächlich gibt es aber zwei Datenbanken, die lediglich sehr schnell synchronisiert werden.

Alle vorgenannten Transformationskategorien sind Bestandteil unserer Kommunikation und wir greifen jeden Tag darauf zurück. In einem vertrauten Kontext, oder wenn wir mit bekannten Stakeholdern arbeiten, sind Transformationsprozesse weniger wahrscheinlich, aber im Allgemeinen beinhalten ein neues System oder neue Funktionalitäten auch neues Wissen.

Um rein kommunikationsbedingte Effekte zu vermeiden, sollte der Requirements Engineer in der schriftlichen und verbalen Kommunikation einige grundlegende Maßnahmen anwenden [Rupp et al.2014]:

- Aussagen immer in vollständigen Sätzen formulieren
- immer Aktivsätze formulieren
- im Glossar definierte Begriffe verwenden
- konsistente Terminologie verwenden und Synonyme oder Homonyme vermeiden

- Prozesse mithilfe von Vollverben ausdrücken

Weitere Informationen über Kommunikation im Requirements Engineering finden Sie in Abschnitt 5.2.

Hinweis 3.3.1:

Beziehen Sie das Suchen nach passenden Definitionen in Ihre Ermittlungsaktivitäten ein. Insbesondere Prozessverben können Ihnen bei der Entwicklung einer einheitlicheren Kommunikation helfen. Begriffsmodelle (z. B. mit UML-Klassendiagrammen) können dazu beitragen, in einer Anforderung den korrekten Datensatz zu adressieren.

Hinweis 3.3.2:

Die Verwendung von Anforderungsschablonen verringert Transformationsdefekte, da für eine Anforderung grundlegende Informationen in einer Schablone erfasst werden müssen.

Requirements Engineers können das in [Rupp et al.2014] vorgestellte SOPHIST-Regelwerk nutzen, das beim Analysieren von Aussagen (oder Anforderungen) hilfreich ist, um Defekte auf Basis der Transformationskategorien (Tilgung, Generalisierung, Verzerrung) zutage zu bringen.

Eine Regel (von insgesamt 18) deckt mindestens einen Transformationseffekt ab. Der Requirements Engineer sucht nach bestimmten Signalwörtern (Schritt 1: Identifizieren) und leitet aus dem Signalwort Fragen für die Stakeholder ab (Schritt 2: Analysieren). Nachdem der Stakeholder die Fragen beantwortet hat, korrigiert oder vervollständigt der Requirements Engineer die Anforderung (Schritt 3: Lösen). Die folgenden Regeln sind Instanzen von Transformationseffekten.

Hinweis 3.3.3:

Es kann hilfreich sein, wenn Sie eine bestimmte Regel in Ihren persönlichen Entwicklungsplan aufnehmen und versuchen, diese in der Kommunikation, in der Dokumentation und in Reviews von Anforderungen umzusetzen. Nach vier Wochen können Sie versuchen, die nächste Regel anzuwenden.

Nominalisierungen auflösen

Nominalisierungen können den Prozess (oder Prozessschritte), die in der Anforderung aufgegriffen werden, verschleiern, sodass nicht klar ist, wie der Prozess durchgeführt werden soll. Die Informationen werden verzerrt (Verzerrung).

Beispielaussage: „Das Bibliothekssystem soll eine Archivierung anbieten.“

Schritt 1: Identifizieren

„Archivierung“ ist die Nominalisierung (also das Signalwort) des Verbs „archivieren“. „Daten archivieren“ ist eine Funktionalität mit festgelegten Schritten. Weder die Schritte, noch die Prozessdetails sind klar (z. B. wann und wo der Prozess beginnt).

Schritt 2: Analysieren

Die Frage, die der Stakeholder beantworten muss, könnte lauten: „Welche Prozessdetails umfasst ‚Archivierung‘?“

Schritt 3: Lösen

In diesem Beispiel führt der Stakeholder (der Bibliothekar) die Prozessschritte „Kundendaten auswählen“ und „Ausgewählte Kundendaten archivieren“ manuell aus.

„Das Bibliothekssystem muss dem Bibliothekar die Möglichkeit bieten, **Kundendaten für die Archivierung auszuwählen.**“

„Wenn ein Kunde aktuell nichts ausgeliehen hat, muss das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, die **ausgewählten Kundendaten zu archivieren.**“

In diesem Beispiel wurde die Nominalisierung aufgelöst, und die beiden Prozessschritte wurden in zwei separaten, aber wechselseitig abhängigen Anforderungen dokumentiert.

Substantive ohne Bezugsindex

Wenn die Aussage eines Stakeholders Nomen ohne einen Referenzindex enthält, ist nicht klar, welche Objekte oder Akteure adressiert werden. Die Formulierung ist zu allgemein, um implementiert zu werden (Generalisierung).

Beispielaussage: „Das System soll dem Benutzer die Daten anzeigen.“

Schritt 1: Identifizieren

In diesem Beispiel ist unklar, welche **Daten** für welchen **Benutzer** angezeigt werden sollen.

Schritt 2: Analysieren

Die Frage, die der Stakeholder beantworten muss, könnte lauten:

- „Welcher Benutzer soll die Daten lesen?“
- Antwort: „Der Bibliothekar“.
- „Welche Daten sollen dem Bibliothekar angezeigt werden?“
- Antwort: „Sämtliche statistisch berechneten Daten von Bibliotheksmedien.“

Schritt 3: Lösen

„Das Bibliothekssystem muss dem Bibliothekar **sämtliche statistisch berechneten Daten von Bibliotheksmedien** anzeigen.“

Hinweis 3.3.4:

Um den Referenzindex zu ermitteln, können Sie die in der Stakeholder-Liste definierten Rollen oder die Rollendefinition des Systems einem überprüfen. Um den korrekten Datensatz zu finden, können Sie das Datenmodell des Systems überprüfen.

Hinweis 3.3.5:

Manchmal kann es hilfreich sein, einen allgemeineren Begriff für einen übergeordneten Datensatz zu definieren. Registrierungsdaten und Zahlungsdetails können beispielsweise als Kundendaten generalisiert werden, was anschließend im Glossar definiert wird.

Universalquantoren

Wenn die Aussage eines Stakeholders Universalquantoren enthält, dann kann die Menge von Objekten entweder zu allgemein oder zu spezifisch sein und muss angepasst werden.

Beispielaussage: „Das Bibliothekssystem soll jedem Kunden die Möglichkeit bieten, sämtliche Kundendaten zu ändern.“

Schritt 1: Identifizieren

Abhängig von der Definition von Kundendaten, könnte die Anforderung korrekt formuliert sein. Aber in diesem Fall wird nach dem Datensatz und der Referenz zum Kunden gefragt.

Schritt 2: Analysieren

Die Frage, die der Stakeholder beantworten muss, könnte lauten:

- Kunden können also jeden Datensatz, der jemals im Bibliothekssystem gespeichert wurde, ändern?
Antwort: „Nein, natürlich nicht. Nur sein Passwort und die Registrierungsdaten seines Profils.“
- „Heißt das, jeder einzelne Kunde kann die Registrierungsdaten aller Kunden ändern?“
Antwort: „Nein, natürlich nicht! „Ein Kunde kann nur **seine eigenen Registrierungsdaten** ändern.“

Schritt 3: Lösen

„Das Bibliothekssystem muss **jedem einzelnen** Kunden die Möglichkeit bieten, **seine eigenen Kundenregistrierungsdaten zu ändern**.“

Am Ende wurde **jeder einzelne** Kunde als korrekter Quantor identifiziert. Der Umfang der Kundendaten war auf **Kundenregistrierungsdaten** begrenzt und der Hinweis, dass ein Kunde andere als **seine eigenen Kundenregistrierungsdaten** ändern kann, wurde korrigiert.

Unvollständige Bedingungsstrukturen

Wenn eine Anforderung Bedingungen enthält, können diese auch mehrere Aspekte beinhalten. Um die Anforderung zu vervollständigen, sollten alle relevanten Bedingungen untersucht werden. Andernfalls bleiben einige Aspekte getilgt (Tilgung).

Beispielaussage: „Wenn ein Medium nicht beschädigt und nicht reserviert ist, soll das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, mit dem Verleihvorgang fortzufahren.“

Schritt 1: Identifizieren

Das Signalwort „wenn“ zeigt eine Bedingung an. Das Signalwort „und“ zeigt an, dass die Bedingung mehrteilig ist.

Schritt 2: Analysieren

Was geschieht, wenn das Medium beschädigt und/oder reserviert ist?

Wie verhält sich das System?

Schritt 3: Lösen

Wenn das Medium **nicht beschädigt, aber reserviert** ist, muss das Bibliothekssystem dem Bibliothekar eine Fehlermeldung anzeigen.

Ist das Medium **beschädigt und nicht reserviert**, muss das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, den Verleihvorgang zu stoppen.

Wenn das Medium beschädigt und reserviert ist, ...

Am Ende müssen sämtliche Kombinationen festgestellt werden und der Requirements Engineer muss für jeden einzelnen Pfad das Verhalten ermitteln.

Durch die detaillierte Analyse von Anforderungen wird das Risiko minimiert, die Realität falsch zu verstehen, aber der Requirements Engineer muss auch die Kosten der Ermittlungsaktivität berücksichtigen. In Abhängigkeit von der Abstraktionsebene (siehe 3.3.1), dem Projektfortschritt und den Eigenschaften der Stakeholder ist eine gewisse Ungenauigkeit tolerabel.

Hinweis 3.3.6:

Wenn die Kommunikation schwierig oder Mechanismen kompliziert erscheinen, bitten Sie mindestens zwei verschiedene Stakeholder um Informationen zu dem entsprechenden Thema.

3.3.4 Modelle als Denkwerkzeug

Der IREB CPRE Foundation Level Lehrplan [IREB2020] vermittelt verschiedene Arten von Modellen (z. B. Datenflussdiagramme, Aktivitätsdiagramme), um Anforderungen zu dokumentieren. Modelle erlauben es, sich auf eine bestimmte Perspektive des Systems zu konzentrieren: Daten, Funktionen oder Verhalten. Dieser Fokus lässt sich nicht nur bei der Dokumentation von Anforderungen anwenden, er kann auch als Denkwerkzeug bei der Anforderungsermittlung dienen. Der Requirements Engineer kann ein bestimmtes Modell auswählen und sich auf die Perspektive konzentrieren, die dieses Modells bietet; das Modell kann ein explizites oder ein implizites Denkwerkzeug sein.

Wird ein Modell als *explizites Denkwerkzeug* verwendet, dann entwickelt der Requirements Engineer das Modell gemeinsam mit den Stakeholdern. Der Requirements Engineer sollte sich darüber im Klaren sein, dass Modelle in solchen Situationen nur nützlich sind, wenn die Modellierungssprache von allen beteiligten Stakeholdern gut verstanden wird.

Ein Beispiel: Ein Requirements Engineer möchte einen bestimmten Geschäftsprozess ausarbeiten, den ein System unterstützen soll. Diese Ausarbeitung könnte im Rahmen eines Workshops mithilfe von Aktivitätsdiagrammen erfolgen. Das Aktivitätsdiagramm, das den Geschäftsprozess abbildet, wird gemeinsam mit den Stakeholdern entwickelt, z. B. indem man es auf einem Whiteboard oder großen Flipcharts aufzeichnet. In einer solchen Situation dient die Notation des Aktivitätsdiagramms als Instrumentarium dafür, was während des Workshops ausgedrückt und dokumentiert werden kann. Der Requirements Engineer muss auf die Inhalte achten, die während des Workshops entwickelt werden, da Stakeholder mit Modellierung in der Regel nicht so vertraut sind. Wenn die Stakeholder die gewählte Notation nicht richtig anwenden, muss der Requirements Engineer sie beim Erstellen eines geeigneten Modells unterstützen. Solche Fehler treten häufig auf, wenn Stakeholder eine wichtige Information ausdrücken möchten, die nicht in das Modell passt.

Hinweis 3.3.7:

Informationen, die nicht in eine ausgewählte Modellierungsnotation passen, sollten nicht unberücksichtigt bleiben. Erarbeitet eine Gruppe von Stakeholdern beispielsweise ein Aktivitätsdiagramm, kommen oftmals Anforderungen im Zusammenhang mit Datenstrukturen auf. Diese Anforderungen können in einem Aktivitätsdiagramm nicht richtig dokumentiert werden. Damit solche Anforderungen nicht verloren gehen, sollten sie beispielsweise im Workshop-Protokoll festgehalten werden, damit eine spätere Analyse möglich ist.

Modelle können auch als *implizites Denkwerkzeug* verwendet werden. In diesem Fall nutzt der Requirements Engineer eine bestimmte Modellierungssprache, um seine Gedanken bei der Anforderungsermittlung zu strukturieren. Das Modell wird kein expliziter Bestandteil der Ermittlungsaktivität und wird nicht mit den beteiligten Stakeholdern diskutiert.

Stattdessen verwendet der Requirements Engineer die Informationen, die er während der Ermittlungsaktivität erhalten hat, um das Modell zu entwickeln. Das ausgearbeitete Modell nutzt er als einen Reflexionspunkt für seine Gedanken und als Stichwort für das Stellen weiterer Fragen oder die Suche nach zusätzlichen Informationen.

Ein Requirements Engineer, der beispielsweise Anforderungen für ein Onlinesystem für den Verkauf von Unfallversicherungen ermittelt, möchte die dafür benötigten Daten verstehen. Er beschließt, Interviews mit Unfallversicherungsexperten durchzuführen und vorhandene papierbasierte Anwendungsformulare zu analysieren.

Er kann mit den Anwendungsformularen beginnen und aus deren Inhalt ein einfaches Datenmodell ableiten. Das Ergebnis dieser Analyse ist ein initiales Datenmodell mit Entitäten, die der Requirements Engineer nicht in Gänze versteht, d. h. es fehlen Beziehungen zwischen Entitäten und die Attribute sind unvollständig. Nun kann der Requirements Engineer anhand dieses unvollständigen Modells die Interviews mit den Versicherungsexperten vorbereiten. Anstatt das Modell beim Interview zu zeigen, verwendet der Requirements Engineer es als Orientierung für das Interview. Die Antworten der Experten können nun mit bekannten Elementen in dem Modell verknüpft werden, wodurch der Requirements Engineer Lücken leichter identifizieren und weitere klärende Fragen zu diesen Bereichen stellen kann.

Hinweis 3.3.8:

Modelle, die als Denkwerkzeug erstellt werden, sollte man nicht mit dokumentierten Anforderungen verwechseln. Das Denkwerkzeug ist ein Zwischenergebnis, das dazu verwendet wird, in einer Folgeaktivität detaillierte, dokumentierte Anforderungen zu entwickeln. Daher ist es ratsam, als Denkwerkzeuge genutzte Modelle zu verwerfen (oder zu archivieren), sobald die Erkenntnisse aus dem Modell in einem Anforderungsdokument erfasst wurden. Andernfalls besteht das Risiko, dass das Denkwerkzeug im Verlauf des Projekts Verwirrung stiftet, da solche Modelle in der Regel nicht aktuell gehalten werden und daher schnell überholt oder mit dem Anforderungsdokument nicht mehr konsistent sind.

3.3.5 Mind-Mapping

Das Mind-Mapping ist eine Aktivität, bei der ein Konzept in Form einer sogenannten Mindmap visuell dargestellt wird. Mit anderen Worten ist eine Mindmap ein grafisches Denkwerkzeug [BuBu2005]. Indem man das Hauptthema ins Zentrum setzt und die Ideen sich in Zweigen von diesem Thema aufgliedern, können Gedanken und Ideen sortiert und umstrukturiert werden. Es sollten Texte und Bilder sowie Farben verwendet werden. Um die Darstellung für das Gehirn „anregender“ zu gestalten, sollte man „langweilige“ Darstellungen (gerade, einfarbige Linien) vermeiden.

Die Idee beruht auf Studien und der Funktionsweise des menschlichen Gehirns. Statt einer linearen oder lateralen Darstellung, wie etwa in Büchern oder Listen, organisiert das Gehirn Wissen mehrdimensional, auch als radial oder strahlenförmig bezeichnet. Eine Mindmap ist ein Ausdruck strahlenförmigen Denkens zur Unterstützung des natürlichen Denkvorgangs [BuBu2005]. Die wesentlichen Merkmale einer Mindmap sind:

1. Der behandelte Sachverhalt ist in einem **zentralen Bild** in jeder Mindmap hervorgehoben.
2. Die wichtigsten Themen dieses Sachverhalts gehen von diesem zentralen Bild aus strahlenförmig in **Zweigen** ab.

- Die Zweige umfassen **wichtige Bilder oder Schlüsselbegriffe** und werden anhand weiterer Zweige, die **Unterthemen** darstellen, verfeinert.

Beim Mind-Mapping wird eine strukturierte Mindmap mit **Hierarchien, Visualisierungen** und **Assoziationen** zwischen Zweigen erstellt, wie in Abbildung 24 dargestellt.



Abbildung 24: Beispiel einer Mindmap mit dem Inhalt dieses Unterkapitels

Es gibt verschiedene Anwendungsmöglichkeiten für eine Mindmap. Sie kann auch unterstützend für Aktivitäten der Ermittlung und Konfliktlösung eingesetzt werden.

Mnemonic – um sich an Wissen zu erinnern

- Eine Mindmap kann zur Visualisierung eines Protokolls in einem Meeting oder einem Workshop verwendet werden. Sie kann sogar während eines Interviews erstellt werden, da darin lediglich die wichtigsten Schlüsselbegriffe erfasst werden. Zudem bietet eine Mindmap einen Überblick über das Hauptthema, das in einem Meeting besprochen werden muss.

Analytisch – um Muster zu erkennen

- Es kann sehr hilfreich sein, unterschiedliche Aspekte eines Themas visuell darzustellen, um Muster in dem verfügbaren Wissen zu erkennen. Auf diese Weise kann der Requirements Engineer Kategorien erstellen oder ein Ungleichgewicht im Wissensumfang feststellen.

Kreativ – um neue Aspekte in einem Gebiet zu entwickeln

- Die Mindmap kann auch in Workshops verwendet werden, um die Ergebnisse eines Brainstormings oder einer Diskussion zu visualisieren. Die Mindmap ist eine schnelle und motivierende Möglichkeit, um Stakeholder zu involvieren; durch den Input von anderen wirkt sie inspirierend.

Dialogisch – um Ähnlichkeiten und Unterschiede zwischen Konzepten und Ansichten zu visualisieren

- Wenn Stakeholder verschiedene Meinungen haben, kann es hilfreich sein, die Kernpunkte verschiedener Personen zu visualisieren und in einer Gruppe zu diskutieren. Anschließend können die Stakeholder eine gemeinsame Mindmap erstellen, um die Absprachen zu dokumentieren.

Um bestmöglich von der Mindmap als Denkwerkzeug zu profitieren, sollte der Autor einige Richtlinien befolgen.

Grundprinzipien des Mind-Mapping:

- Fügen Sie in der Mitte jeder Mindmap ein repräsentatives, zentrales Bild zum Thema ein;
- Verwenden Sie für jeden Zweig oder jedes Thema Schlüsselbegriffe;
- Verwenden Sie Hierarchien und eine Nummerierung, um die Mindmap zu organisieren;
- Organisieren Sie die Mindmap, um den Überblick zu behalten, und lassen Sie Freiraum zur Klärung;
- Verwenden Sie Bilder, Symbole und Metaphern, um alle menschlichen Sinne anzusprechen;
- Visualisieren Sie Verbindungen zwischen Unterthemen und Zweigen mithilfe von Farben, Codes und Pfeilen;
- Betonen Sie wichtige Aspekte durch Variationen in der Schriftart, der Art von Zeilen sowie der Farbe und Größe der visuellen Darstellung;
- Verwenden Sie unterschiedliche Größen in der Struktur Ihrer Mindmap;
- Nutzen Sie Farben und „Wolken“, um zusammengehörige Zweige einzurahmen;
- Fügen Sie Ihrer Mindmap offene oder inspirierende Fragen hinzu.

Weitere Orientierung bietet [BuBu2005].

Der Requirements Engineer kann alle Arten von Werkzeugen (z. B. Canvases oder Boards) zur Erstellung von Mindmaps verwenden. Eine Variation des Mind-Mappings ist die Verwendung eines computerbasierten Werkzeugs, mit dem sich Ergebnisse digital dokumentieren und dann über einen Projektor präsentieren lassen. Das erleichtert das Reorganisieren der Mindmap sehr. Solche Werkzeuge ermöglichen sogar die gleichzeitige Bearbeitung einer Mindmap von überall auf der Welt, was die Möglichkeiten für die Zusammenarbeit erweitert.

Die Zweige der Mindmap enthalten oft wichtige Begriffe für das Glossar oder das Begriffsmodell, und Assoziationen zwischen Zweigen können als „Abhängigkeiten“ zwischen Objekten dokumentiert werden.

3.4 Ermittlungstechniken mit Attributen beschreiben

Requirements Engineers sollten, abhängig von dem konkreten Kontext und den Bedürfnissen in der aktuellen Situation, die anzuwendende Ermittlungstechnik mit Bedacht wählen. Viele Wissenschaftler empfehlen Modelle und Frameworks, um Ermittlungstechniken den gegebenen Umständen zuzuordnen. Carrizo et al. bieten beispielsweise umfangreiche Untersuchungen zur systematischen Auswahl von Ermittlungstechniken an [CRCN2014], während Tiwari und Singh eine Methode für die Auswahl von Techniken zur

Anforderungsermittlung vorschlagen und die zugehörige Forschungsarbeit zu diesem Thema übersichtlich zusammenfassen [TiSi2017].

In den vorgenannten Ansätzen konzentrieren sich die Autoren zwar häufig auf die Variabilität des vorliegenden Problems (Projekt, Benutzer, Kontext usw.), bieten jedoch eine direkte Zuordnung zu geeigneten Ermittlungstechniken an.

Zum Erlernen der Techniken für die Anforderungsermittlung, ohne jeweils das Pro und Kontra im Detail auswendig wissen zu müssen (d. h. nicht wie von [YoAs2015] vorgeschlagen), stellen wir *identifizierende Attribute* vor, die beim Klassifizieren von Ermittlungstechniken helfen. Mit dieser zusätzlichen Ebene zwischen der Problemstellung (wie ermittelt man Anforderungen unter den gegebenen Umständen) und dem Lösungsspektrum (Hunderte von verfügbaren Ermittlungstechniken), kann der Lernende sich auf die wichtigsten Problemlösungskonzepte der Anforderungsermittlung konzentrieren, das heißt: Welche identifizierenden Attribute muss eine Ermittlungstechnik aufweisen, um das Ermittlungsziel zu erreichen (siehe Abschnitt 1.3).

Die nachstehende Tabelle 3.4.3-1 enthält eine Liste von identifizierenden Attributen. Wir definieren ein Attribut als *identifizierend*, wenn es sich um eine grundlegende Eigenschaft einer Ermittlungstechnik handelt.

Die Merkmale jeder einzelnen Ermittlungstechnik lassen sich durch eine Kombination dieser Attribute beschreiben. Beispielsweise ist die Technik des *Interviewens* durch die Attribute *dialogorientiert* und *fragend* gekennzeichnet. Ein Interview könnte auch *beobachtend* sein, wenn der Requirements Engineer es in der Anwendungsumgebung des Endanwenders durchführt. Doch ist *beobachtend* keine Kerneigenschaft (d. h. identifizierend) des Interviews, da es auch telefonisch oder in einer anderen Umgebung durchgeführt werden kann, in der Beobachtungen nicht möglich sind.

Die Klassifizierung einer langen Liste verfügbarer Techniken nach relevanten Attributen kann dabei helfen, die richtigen Techniken für eine bestimmte Situation auszuwählen. Tabelle 3.4.3-2 bietet eine solche Klassifizierung für einige Techniken.

„Es gibt ‚Good Practices‘ in einem bestimmten Kontext, doch es existieren keine ‚Best Practices‘“ [KaBP2002]: Jede Anwendungssituation erfordert eine bestimmte Kombination aus Ermittlungstechniken, die anhand der relevanten Attribute für diese Situation identifiziert werden.

Oder anders ausgedrückt: Die relevanten Attribute sollten angesichts des Ermittlungsziels, der Projektsituation, der Stakeholder usw. bestimmt und die Ermittlungstechnik(en), die genau zu diesen identifizierenden Attributen passen, ausgewählt werden. Tabelle 3.4.3-3 enthält ein Beispiel für eine solche Zuordnung von Kontext und Attributen.

Tabelle 3.4.3-1: 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
Befragend	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;
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;	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);

Attribut	Kurze Beschreibung	Geeignet, um folgende Ziele zu erreichen	Geeignet in folgenden Situationen
Zustimmend oder Widerspruch hervorrufend	Relevante Aspekte einer Lösungsmöglichkeit demonstrierend, 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;
Artefaktbasiert	Erhebung und Analyse vorhandener Arbeitsergebnisse (z. B. Dokumente, Modelle, Produkte oder eingesetzte Systeme)	Um Anforderungen von bestehenden Arbeitsergebnissen abzuleiten; um (Basisfaktoren/) 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;

Tabelle 3.4.3-2: Ausgewählte Ermittlungstechniken, beschrieben durch ihre identifizierenden Attribute

Techniken	Attribute						
	Dialog-orientiert	Befragend	Beobachtend	Zustimmend oder Widerspruch hervorrufend	Artefaktbasiert	Kreativitätsfördernd	Erlebend
Interview							
Fragebogen							
Anforderungsworkshop (z. B. Fokusgruppe)							
Feldbeobachtung							
Apprenticing							
Contextual Inquiry							
Kreativitätstechniken (Brainstorming, ...)							
Systemarchäologie							
Perspektivenbasiertes Lesen							
Wiederverwendung von Anforderungen							
Prototyping							

Techniken	Attribute						
	Dialog-orientiert	Befragend	Beobachtend	Zustimmend oder Widerspruch hervorrufend	Artefaktbasiert	Kreativitätsfördernd	Erlebend
Szenarien							
Storyboards							
User Walkthrough							
Usability-Tests							
Erraten / Erahnen von Anforderungen							
Ausarbeitung von User-Storys (Card, Conversation, Confirmation)							
Untersuchung von Tagebüchern							
Sortieren von Karten							
...							

Legende:

- | **Identifizierendes Attribut, d. h. es handelt sich um eine Schlüsseleigenschaft der Technik – ohne dieses Attribut ist es eine andere Technik.**

<leer>

Das Attribut ist keine Kerneigenschaft (d. h. **kein identifizierendes Attribut**), wenngleich es möglich ist, dass dieses Attribut unter bestimmten Umständen bei dieser Technik genutzt wird.

Tabelle 3.4.3–3: Ermittlungsziele, Randbedingungen und Projektsituationen, durch die identifizierende Attribute festgelegt sind, die geeignete Ermittlungstechniken enthalten sollten:

Fragestellung der Untersuchung/Randbedingung/Projektsituation	Identifizierende/s Attribut/e von Techniken, die Sie anwenden können
Wenn Sie das implizite Wissen von Stakeholdern ermitteln müssen	Beobachtend
Wenn es nicht möglich ist, Benutzer während ihrer Arbeit an einer Aufgabe zu unterbrechen	Beobachtend
Wenn relevante Aspekte eines Vorgängersystems nicht klar sind oder eine wichtige Rolle spielen	Beobachtend und/oder dialogorientiert
Wenn Sie bereits (Teile eines) ein existierendes System, ein Vorgängersystem oder ähnliche Systeme haben, oder wenn Sie bereits mögliche Lösungen oder Mock-ups erstellt haben, z. B. durch Prototyping	Zustimmend oder Widerspruch hervorrufend
Wenn die Umgebung, in der das zukünftige System verwendet wird, eine wichtige Quelle für Anforderungen darstellt (Lärmpegel, Zeitdruck, physische Bedingungen etc.)	Erlebend
Wenn Sie wichtige Usability-Anforderungen ermitteln müssen	Erlebend
Wenn Sie sich mit Ihrem Projekt in unbekanntem Gebiet bewegen	Zustimmend oder Widerspruch hervorrufend, kreativitätsfördernd
Wenn kein Stakeholder bestimmte Anforderungen in einem bestimmten Detaillierungsgrad angeben kann	Zustimmend oder Widerspruch hervorrufend
Wenn Sie ein komplexes System entwickeln, das von Gelegenheitsnutzern verwendet werden soll	Zustimmend oder Widerspruch hervorrufend
Wenn Sie nicht-menschliche Anforderungsquellen identifiziert haben	Artefaktbasiert
Wenn Sie herausfinden wollen, wo im Prozess sich Schwachstellen befinden	Artefaktbasiert, beobachtend und erlebend
Wenn Sie herausfinden möchten, wie und mit welchen Arbeitsergebnissen Ihre künftigen Anwender tatsächlich arbeiten	Erlebend und Artefaktbasiert

Fragestellung der Untersuchung/Randbedingung/Projektsituation	Identifizierende/s Attribut/e von Techniken, die Sie anwenden können
Wenn Sie Material zur Vorbereitung von Ermittlungsaktivitäten mit Stakeholdern benötigen	Artefaktbasiert
Wenn Sie einen Katalysator benötigen, damit Stakeholder Ideen für Anforderungen bekommen, an die sie sich sonst nicht erinnert hätten oder die ihnen nicht bewusst gewesen wären	Artefaktbasiert
Wenn Sie ein innovatives neues System entwickeln, das neue Features oder neue Wege der Interaktion bieten soll (d. h. wenn Sie nach Basisfaktoren gemäß Kano-Modell suchen)	Kreativitätsfördernd
Wenn die Lösung für Ihr Problem nicht offensichtlich ist und Sie und Ihr Team einen neuen Ansatz präsentieren sollen	Kreativitätsfördernd
Wenn der Requirements Engineer die Lösung spezifizieren muss, anstatt nur Stakeholdern zuzuhören und die Anforderungen, die ihm genannt werden, zu erfassen	Erlebend
Wenn die Requirements Engineers nach der initialen Implementierung mit der Pflege der Lösung fortfahren sollen (d. h. das gewachsene und fundierte Know-how, das der Requirements Engineer während seiner Ermittlungsaktivität gewonnen hat, ist für sich genommen eine wertvolle Ressource für den Sponsor)	Erlebend
Wenn relevante Stakeholder in bestimmten Projektkonstellationen kaum verfügbar sind und es daher einfacher ist und/oder schneller geht, wenn sich der Requirements Engineer mit dem zu untersuchenden Anwendungsgebiet beschäftigt, um selbst Anforderungen zu entwickeln (und diese von den Stakeholdern nur noch prüfen lässt)	Erlebend
Wenn Sie Anforderungen in Bezug auf die aktuelle Situation, die Bedürfnisse und mögliche Lösungen klären müssen	Dialogorientiert
Wenn Sie sicherstellen wollen, dass Sie dieselbe Sprache sprechen wie die Stakeholder, vor allem auf spezialisierten Anwendungsgebieten	Dialogorientiert
Wenn ein hohes Maß an Benutzerinteraktion vorherrscht	Erlebend, Zustimmung oder Widerspruch hervorrufend
Wenn ein hohes Maß an technischer Integration vorherrscht	Artefaktbasiert

Fragestellung der Untersuchung/Randbedingung/Projektsituation

Identifizierende/s
Attribut/e von
Techniken, die Sie
anwenden können

Wenn das Maß an Innovation (=> Kano) hoch ist

Kreativitätsfördernd,
Zustimmung oder
Widerspruch
hervorrufend

4 Konfliktlösung

Während der Anforderungsermittlung erhebt der Requirements Engineer eine große Sammlung von Anforderungen, häufig aus unterschiedlichen Quellen (siehe Kapitel 2), mithilfe verschiedener Techniken sowie mit unterschiedlichen Abstraktionsebenen und Detaillierungsgraden (siehe Kapitel 3). Ermittlungstechniken alleine stellen aber noch nicht sicher, dass diese Sammlung als Ganzes alle Qualitätskriterien für Anforderungen erfüllt (siehe [IREB2020]). Werden Qualitätskriterien nicht erfüllt, sind unter Umständen zusätzliche Ermittlungsaktivitäten erforderlich, um die Qualität zu verbessern.

Zusätzliche Aktivitäten sind notwendig, um diese Sammlung in eine einzige, konsistente Konstellation von Anforderungen zu verwandeln, die die wesentlichen Merkmale des Systems erfasst. Oftmals stellt sich heraus, dass einige Anforderungen gegensätzlich sind: inkonsistent, inkompatibel, widersprüchlich. Dies wird gewöhnlich durch Meinungsverschiedenheiten zwischen bestimmten Stakeholdern verursacht. Folglich ist die „Abgestimmtheit“ ein äußerst wichtiges Qualitätskriterium, das immer geprüft werden sollte: Alle Stakeholder müssen sämtliche für sie relevanten Anforderungen verstehen und diesen zustimmen. Stimmen einige der Stakeholder nicht zu, so handelt es sich um einen Anforderungskonflikt, der entsprechend aufgelöst werden sollte.

Eines der Ziele des Requirements Engineering ist es, einen Konsens zwischen den Stakeholdern über die Anforderungen zu erreichen. Eine der Hauptaufgaben in dieser Hinsicht ist der Umgang mit Anforderungskonflikten.

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.

In unserem Alltag sind wir häufig in Konflikte involviert. Da sie in aller Regel nicht sehr angenehm zu lösen sind, ist eine häufige Strategie, sich ihnen zu entziehen. Dies hat zur Folge, dass Menschen nicht miteinander sprechen, sich nach anderen Arbeitsbereichen umsehen oder sogar die Arbeitsstelle wechseln, damit sie sich nicht mehr mit dem Konflikt auseinandersetzen müssen.

Der Umgang mit Anforderungskonflikten kann aufreibend und zeitintensiv sein, vor allem wenn es sich um persönliche Probleme handelt. Es ist allerdings unverzichtbar, die folgenden Aspekte zu berücksichtigen:

1. Das Lösen persönlicher Probleme ist nicht Teil der Jobbeschreibung und muss anhand verschiedener Managementaktivitäten eskaliert werden. Siehe Abschnitt 4.2 – Konfliktanalyse.
2. Vor Anforderungskonflikten davonzulaufen ist allerdings keine Option, da ungelöste Anforderungskonflikte zur Folge haben, dass die Anforderungsdokumente von geringerer Qualität sind und Stakeholder frustriert sind.

Ein stetes Bewusstsein für Konflikte und die regelmäßige Durchführung von Reviews [IREB2020] trägt dazu bei, Konfliktindikatoren zu erkennen und Daten für die Konfliktlösung zu sammeln.



KI-gestützte Werkzeuge, basierend auf generativer KI und sogenannter Large Language Models (LLM), können bei der Bearbeitung von Konflikten im Requirements Engineering hilfreich sein. KI-gestützte Analysewerkzeuge ermöglichen es beispielsweise, Kommunikationsdaten wie Gesprächsprotokolle, Stakeholder-Feedback oder schriftliche Anforderungen automatisiert zu untersuchen. So lassen sich frühzeitig Widersprüche und Konflikte zwischen Anforderungen erkennen und deren mögliche Ursachen sichtbar machen. Darüber hinaus können KI-basierte Assistenzsysteme in Workshops eingesetzt werden, um unterschiedliche Sichtweisen in Echtzeit zu erfassen, zu bündeln und übersichtlich darzustellen. Auf diese Weise unterstützt KI den Requirements Engineer gezielt bei der Vorbereitung, Durchführung und Dokumentation von Konfliktlösungsprozessen. Dabei ergänzt KI das methodische Vorgehen und stärkt die reflektierte und zielgerichtete Bearbeitung konfliktärer Anforderungen, ohne die notwendige menschliche Urteilsfähigkeit zu ersetzen.

4.1 Konfliktidentifizierung

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: [Glasl2004] definiert einen sozialen Konflikt als „... eine Interaktion zwischen Akteuren (Individuen, Gruppen, Organisationen usw.), wobei wenigstens ein Akteur eine Differenz bzw. Unvereinbarkeiten im Wahrnehmen und Denken bzw. Vorstellen und im Fühlen und im Wollen mit dem anderen Akteur (den anderen Akteuren) in der Art erlebt, dass beim Verwirklichen dessen, was der Akteur denkt, fühlt oder will eine Beeinträchtigung durch einen anderen Akteur (die anderen Akteuren) erfolge.“

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.“ [Rupp et al.2014] 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:** Ein Stakeholder leugnet seine eigenen Aussagen oder Aussagen anderer Stakeholder oder bestreitet eine zuvor getroffene Vereinbarung bzw. abgestimmte Aussage.
- **Gleichgültigkeit:** Ein Stakeholder trägt nicht zu einer Diskussion bei (oder möchte nicht dazu beitragen) oder bestätigt etwas ohne kritisches Hinterfragen.
- **Pedanterie:** Ein Stakeholder findet bei allen Vorschlägen Fehler und immer ein winziges Problem, das er nutzt, um Argumente gegen den Vorschlag anzubringen.
- **Zerfragen:** Ein Stakeholder nutzt seine Position oder sein Wissen, um alle Aussagen sehr kritisch zu hinterfragen. Es mag so wirken, als wollte der Stakeholder sicherstellen, dass wichtige Anforderungen nicht vergessen werden, aber tatsächlich wird dadurch der Ermittlungsprozess verlangsamt.
- **Fehlinterpretation:** Ein Stakeholder interpretiert Fakten absichtlich falsch, um den Ermittlungsprozess zu erschweren oder zu verlangsamen.
- **Verschleierung:** Ein Stakeholder verbirgt bewusst oder unbewusst Informationen und äußert sie nur auf Rückfrage.
- **Delegation:** Ein Stakeholder legt sich nur lose auf Aussagen fest und verlangt, dass andere sie detaillierter erläutern.

Häufig anzutreffende Indikatoren in der Dokumentation sind (mit Beispielen):

- **Widersprüchliche Aussagen von Stakeholdern:** Stakeholder einigen sich im Verlauf von Workshops auf eine Anforderung, die nicht konsistent mit einer Anforderung ist, die aus einem Interviewprotokoll einer vorherigen Ermittlungsaktivität stammt.
- **Widersprüchliche Ergebnisse aus Dokumenten- und Systemanalyse:** Die Schnittstellenspezifikation eines Systems enthält eine vorläufige Adresse für einen Kunden, im System kann allerdings nur eine zweite Hauptadresse eingegeben werden.
- **Auf Detailebene inkonsistente Anforderungen:** Das System identifiziert doppelte Kundendatensätze über Namen, Geburtsdatum und Anschrift, aber es gibt einige Kunden ohne Adresse.
- **Inkonsistente Verwendung von Begriffen in der Spezifikation:** Stakeholder verwenden die Begriffe Kunde, Benutzer und Mitarbeiter in unterschiedlichen Bedeutungen, wenden aber nicht die Definitionen aus dem Glossar an.

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. Mit den meisten Ermittlungsaktivitäten bringt er Stakeholder dazu, klar Position zu beziehen, wodurch unerwartete Probleme oder existierende Konflikte aufgedeckt werden.

4.2 Konfliktanalyse

Es sollte den Requirements Engineer misstrauisch machen, wenn es keinerlei Anzeichen für Anforderungskonflikte gibt. In diesem Fall sollte er Reviews ansetzen. Sobald ein Verdacht auf einen Konflikt besteht, kann er eine Datensammlung für einen Konflikt in die Aktivitäten der Anforderungsermittlung aufnehmen.

Wurde ein Konflikt identifiziert, muss der Requirements Engineer klären, ob es sich dabei 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. Bei Konfliktanalysen gibt es verschiedene Indikatoren für Konflikte. Es kann hilfreich sein, zunächst die Indikatoren zu sammeln und später zu überprüfen. Sind mehr Informationen verfügbar, dann ist es einfacher, eine geeignete Lösung für den Konflikt zu finden.

Die folgenden Eigenschaften [Rupp et al.2014] eines Konflikts können dabei helfen, die Art des Konflikts zu verstehen und eine passende Lösung zu finden.

4.2.1 Eigenschaften von Anforderungskonflikten

Die *Art des Konflikts* bestimmt dessen Eigenschaften und den Grad der persönlichen Beteiligung der Stakeholder. Dies ist einer der wichtigsten Indikatoren, auf dessen Basis Konfliktlösungstechniken ausgeschlossen oder angewendet werden sollten (siehe Abschnitt 4.3 – Konfliktlösung). In manchen Fällen ist die Bestimmung der Konfliktart unter Umständen schwierig. Ist das der Fall, sollten verschiedene Konfliktarten in Betracht gezogen werden.

Der *Sachverhalt* ist das einem Konflikt zugrunde liegende Problem. Herauszufinden, was das eigentliche Problem einer Diskussion ist, kann sehr schwierig sein; dies hängt von der Art des Konflikts und dessen Vorgeschichte ab. Es ist jedoch auch sehr nützlich, um eine entsprechende Lösung zu finden. Der Requirements Engineer sollte in Bezug auf die bestehenden Optionen neutral sein. Er kann die Analyse aktiv unterstützen, indem er die Aussagen der Beteiligten spiegelt.

Betroffene Anforderungen stellen repräsentative Aussagen zu dem Konflikt dar. Sie lassen sich für die Analyse und Darstellung der Details verwenden. Sobald für einen Konflikt eine Lösung gefunden wurde, sollte die Dokumentation der relevanten Anforderungen ein Leichtes sein.

Hinweis 4.2.1:

Der Requirements Engineer kann Ihre Stakeholder darum bitten, ihre Zweifel als Anforderungen zu formulieren, um die Aussagen von anderen Konfliktparteien bestätigen oder ablehnen zu lassen. Dies unterstützt eine präzise und reflektierende Kommunikation.

Beteiligte Stakeholder können die Autoren oder andere Beteiligte sein, die in irgendeiner Weise für die betreffenden Anforderungen verantwortlich sind. Sie sind Informationsquellen für die Analyse und können selbst Teil des Konflikts sein. Manchmal ist es hilfreich, weitere Stakeholder hinzuzuziehen, damit diese ihre Expertise einbringen und den Konflikt zwischen den Parteien aufgrund ihrer Autorität abschwächen oder sogar lösen.

Konfliktpositionen sind die Aussagen der Stakeholder oder eine verbale Zusammenfassung der Konzepte, die sie im Kopf haben. Als Requirements Engineer kann man diese vor den

anderen Beteiligten umformulieren (oder sie von Stakeholdern umformulieren lassen). Vage oder unklare Aspekte können erläutert werden, damit die Konfliktparteien das zugrunde liegende Problem leichter verstehen.

Die *Konfliktursache* ist der Grund dafür, weshalb die Stakeholder nicht selbstständig weiterarbeiten können. Wenn diese geklärt ist, findet sich der Weg zu einer möglichen Lösungstechnik oder gar zur Lösung leichter.

Gibt es eine Vorgeschichte oder *Historie* des Konflikts, so kann sie es neu hinzugekommenen Parteien erleichtern, die bisherigen Ansätze oder Argumente gegen betroffene Anforderungen oder Optionen zu verstehen. Es ist wahrscheinlich, dass nicht alle Parteien denselben Wissensstand haben, was sogar der Hauptgrund für den Konflikt sein kann.

Die *Konfliktfolgen* sind die mit den relevanten implementierten Anforderungen verbundenen Kosten. Diese Kosten sind manchen Beteiligten möglicherweise nicht klar. Sie sollten daher geschätzt werden, um zur Konfliktlösung beizutragen.

Die *Konflikttrisiken* können sehr wichtig für die Entscheidung sein, wann (d. h. in welcher Phase der Analyse) der Konflikt gelöst werden soll.

Projektrandbedingungen sind persönlicher, organisatorischer, inhalts- oder domänenspezifischer Art. Sie hängen mit der Konfliktart zusammen und beeinflussen die Wahl der geeigneten Konfliktlösungstechnik. So ist es beispielsweise möglich, dass in einer bestimmten Projektsituation nicht genügend Zeit ist (organisatorische Projektrandbedingung), um den Konflikt mithilfe der Technik der Einigung zu lösen (siehe Abschnitt 4.3 – Konfliktlösung).

4.2.2 Die Konfliktarten nach Moore

Die Konfliktart ist wichtig, um zu entscheiden, ob es sich bei einem vorliegenden Konflikt um einen Anforderungskonflikt handelt. [IREB2020] hat die fünf ursprünglichen Konflikttypen von Moore [Moor2014] um den Sachkonflikt erweitert, da dieser Konflikt im RE häufig auftritt:

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

Die meisten Anforderungskonflikte können als Daten-, Interessen- oder Wertekonflikt kategorisiert werden. Sachkonflikte, falls vorhanden, werden oft in frühen Projektphasen 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. Daher sollte der Requirements Engineer allen

Konfliktarten Aufmerksamkeit schenken, auch wenn die Lösung nicht in seinem Verantwortungsbereich liegt.

4.2.2.1 Sachkonflikt

Ein Sachkonflikt liegt vor, wenn die Konfliktparteien unterschiedliche sachliche Bedürfnisse haben, die zumeist durch die beabsichtigte Nutzung des Systems in unterschiedlichen Umgebungen verursacht werden. Manchmal ist das Vorhandensein eines Sachkonflikts für bestimmte Beteiligte von Anfang an klar, wird aber ignoriert, um den Projektstart nicht zu verzögern. Häufiger wird er allerdings festgestellt, wenn der Kreis der Stakeholder nach der ersten Projektphase erweitert wird.

Beispiele für Sachkonflikte:

- Ein System, das in verschiedenen Ländern mit jeweils eigener Gesetzgebung angewendet werden soll.
- Die Stakeholder einer Personalabteilung fragen nach konsolidierten Daten am Ende des Arbeitstages, während die Stakeholder des Service Desks Echtzeitdaten benötigen.

Es kann schwierig sein, einen solchen Konflikt zu lösen, da die zugrundeliegende Sachlage nicht geändert werden kann.

Vermeidungs- und Lösungsstrategien:

- Analysieren und dokumentieren Sie den Sachverhalt im Detail.
- Die Stakeholder sollten sich über die genaue Art des Konflikts einigen.
- Die Stakeholder sollten ihre Risiken, Kosten oder Schäden angeben, wenn der Konflikt nicht zu ihrer Zufriedenheit gelöst wird.

4.2.2.2 Datenkonflikt

Ein Datenkonflikt beruht auf einem Mangel an oder einer ungleichen Verteilung von Wissen, oder auf einer anderen Interpretation der den Konfliktparteien zur Verfügung stehenden Daten.

Beispiele für Datenkonflikte:

- Stakeholder können sich über die Existenz oder die Auslegung einer in einer Anforderung beschriebenen Geschäftsregel streiten.
- Stakeholder sind über den Grund für eine Anforderung unterschiedlicher Meinung.
- Stakeholder haben ein unterschiedliches Verständnis von den Begriffen und deren Definitionen, die in Anforderungen enthalten sind.

Wenn ein Datenkonflikt der Grund für eine Diskussion ist, lässt sich beobachten, dass die Kommunikation zwischen den Konfliktparteien sich auf die Seite „Sachinhalt“ des Vier-Seiten-Modells von Schulz von Thun [Schulz von Thun 1981] konzentriert. Stakeholder tauschen Informationen aus und teilen wichtige Fakten und Zahlen.

Vermeidungs- und Lösungsstrategien:

- Den Stakeholdern müssen zusätzliche Informationen bereitgestellt werden, die durch Austausch oder Sammlung ermittelt werden.
- Die für den spezifischen Datenkonflikt relevanten Informationen müssen identifiziert werden.
- Stakeholder sollten sich über einen gemeinsamen Datenerfassungsprozess mit Auswertungskriterien einigen.
- Stakeholder sollten sich auf Experten einigen, die relevante Informationen liefern.

4.2.2.3 Interessenkonflikt

Ein Interessenkonflikt beruht auf den unterschiedlichen Motivationen der Konfliktparteien. Motivationen können sich durch persönliche Ziele, durch Ziele in Bezug auf eine Gruppe oder auf eine Rolle bilden. Da ein Interessenkonflikt nicht auf dem Besitz von Informationen beruht (wie der Sachkonflikt), ist es für die Lösung dieser Konfliktart wichtig, die Anliegen und Bedürfnisse der Stakeholder zu verstehen. Im Fall von persönlichen Interessen legen Stakeholder ihre wahren Motive häufig nicht offen und suchen stattdessen nach künstlichen Argumenten.

Beispiele für Interessenkonflikte:

- Ein Stakeholder der Sicherheitsabteilung kann beispielsweise höhere Standards für die Verschlüsselung fordern, was mehr Zeit erfordert, während der Benutzer des Systems den Schwerpunkt auf die Systemleistung für seine tägliche Arbeit legt.
- Ein Stakeholder möchte aus Prestigegründen für seine Mitarbeiter, dass seine Abteilung die Verantwortung für die Implementierung einer Funktion erhält. Der Systemarchitekt tritt hingegen für eine andere Komponente ein, um die Stabilität der Systemarchitektur zu verbessern.
- Ein Stakeholder benötigt für seine Arbeit die Implementierung einer Funktion im nächsten Release, der Sponsor ist allerdings der Ansicht, dass andere Funktionen für die Mehrheit der Anwender wichtiger sind.

Wenn ein Interessenkonflikt der Grund für eine Diskussion ist, lässt sich beobachten, dass die Kommunikation zwischen den Konfliktparteien sich auf die Seite „Appell“ des Vier-Seiten-Modells von Schulz von Thun [Schulz von Thun 1981] konzentriert. Die Konfliktparteien versuchen, andere davon zu überzeugen, ihren Argumenten zu folgen und die Bedürfnisse der Rolle oder Gruppe zu verstehen.

Vermeidungs- und Lösungsstrategien:

- Interessen und Folgen müssen offengelegt und in einem anderen Kontext oder von einer anderen Instanz betrachtet werden, die objektiver sein kann.
- Aus diesen Interessen müssen die für den Konflikt relevanten Fakten extrahiert werden, um eine Lösung auf Basis von Fakten zu unterstützen.

4.2.2.4 Wertekonflikt

Ein Wertekonflikt beruht auf unterschiedlichen Werten und Prinzipien. Er bezieht sich auf den Interessenkonflikt, ist jedoch individueller und umfasst globale oder langfristige Perspektiven. Wenn eine Person ihre Rolle wechselt, können sich die Interessen ändern. Werte sind hingegen stabiler und ändern sich auf kurze Sicht kaum.

Beispiele für Wertekonflikte:

- Es gibt möglicherweise Stakeholder, die Produkte mit viel Plastik oder nicht recycelbare Produkte vermeiden. Andere bevorzugen dagegen preisgünstige Produkte. Hier steht der Wert des Umweltschutzes dem Wert einer günstigen Produktion entgegen.
- Es gibt Stakeholder, die eine grafische Darstellung bei Software für weniger wichtig halten als befehlsbasierte Anwendungen.
- Stakeholder halten Preisdiskriminierung (= ein Anbieter verkauft dasselbe Produkt zu unterschiedlichen Preisen an unterschiedliche Kunden) für die Benutzer einer E-Business-Website für unfair.

Wenn ein Wertekonflikt der Grund für eine Diskussion ist, lässt sich beobachten, dass die Kommunikation zwischen den Konfliktparteien sich auf die Seite „Selbstkundgabe“ des Vier-Seiten-Modells von Schulz von Thun [Schulz von Thun 1981] konzentriert. Die Konfliktparteien betonen, weshalb ihre Argumente aus ihrer Sicht wichtig sind und nennen eine Reihe von Argumenten, die ihre inneren Werte und Prinzipien erkennen lassen. Sie neigen dazu, auf ihren Argumenten zu insistieren und scheinen Erfahrung im Diskutieren in Bezug auf den entsprechenden Aspekt zu haben.

Vermeidungs- und Lösungsstrategien:

- Stakeholder sollten den Konfliktparteien einräumen, Argumenten zuzustimmen oder diese abzulehnen, ohne darüber zu urteilen, sondern diese zu tolerieren.
- Konfliktparteien sollten sich auf die Gemeinsamkeiten konzentrieren, in denen ihre Werte übereinstimmen.
- Konfliktparteien sollten sich auf ein gemeinsames globales Ziel und den größeren Kontext konzentrieren, anstatt auf ihre Differenzen und Details.

4.2.2.5 Strukturkonflikt

Ein Strukturkonflikt basiert auf einer ungleich verteilten Macht, dem Wettbewerb um knappe Ressourcen und auf strukturellen Abhängigkeiten, welche die Konfliktparteien beeinflussen. Das wahrgenommene Ungleichgewicht verursacht Probleme in der Kommunikation, bei der Ermittlung von Anforderungen und bei der Entscheidungsfindung. Ein anderer Grund für einen Strukturkonflikt können strenge Einschränkungen von Ressourcen oder Abhängigkeiten von Arbeitsergebnissen sein, die von anderen zugeliefert werden.

Beispiele für Strukturkonflikte:

- Stakeholder können Anforderungen unterdrücken, da sie glauben, dass eine Konfliktpartei mit größerer Autorität Argumente dagegen vorbringen könnte.

- Stakeholder mit größerem Einfluss in der Organisation können versuchen, die Priorität von Anforderungen zu ändern.
- Stakeholder, die mehr Transparenz wünschen, formulieren eine Anforderung für den Zugang zu einer bestimmten Anwendung, damit sie nicht von einer anderen Abteilung überrascht werden können, die Informationen immer zu spät liefert.

Wenn ein Strukturkonflikt der Grund für eine Diskussion ist, lässt sich beobachten, dass die Kommunikation zwischen den Konfliktparteien sich auf die Seite „Beziehungshinweis“ des Vier-Seiten-Modells von Schulz von Thun [Schulz von Thun 1981] konzentriert.

Konfliktparteien können die Diskussion über Anforderungen dazu nutzen, den Status quo entweder zu ändern oder ihn aufrechtzuerhalten. Abhängig von ihrem Standpunkt betonen die Konfliktparteien ihre Ablehnung gegenüber der Änderung (für die „machthabende“ Partei: Bewahrung) aktueller Strukturen oder Beziehungen oder den Wunsch, diese zu ändern.

Vermeidungs- und Lösungsstrategien:

- Verantwortlichkeiten (z. B. für die Bereitstellung von Anforderungen) und Ressourcen müssen von einer Partei in einer Seniorposition neu verteilt werden.
- Abhängigkeiten müssen von einer Partei in einer höheren Position aufgelöst werden.
- Es muss ein anderer Entscheidungsprozess implementiert werden.
- Externer Druck muss auf eine Weise neu definiert werden, dass er die Arbeit oder die Anforderungen der Konfliktpartei nicht beeinflusst.

Da die meisten der hier beschriebenen Lösungsstrategien unter Umständen nicht im Verantwortungsbereich des Requirements Engineers liegen, kann er Strukturkonflikte nur eskalieren und die anderen Stakeholder intervenieren lassen.

4.2.2.6 Beziehungskonflikt

Ein Beziehungskonflikt kann auf vergangenen, negativen Erfahrungen mit den Konfliktparteien basieren oder auf negativen Erfahrungen aus vergleichbaren Situationen mit ähnlichen Personen. Solche Konflikte stehen häufig mit Emotionen und Fehlkommunikation in Verbindung, was ihre Lösung deutlich erschwert.

Beispiele für Beziehungskonflikte:

- Stakeholder greifen die andere Konfliktpartei ohne wichtigen Grund oder auf sehr emotionale Weise verbal an. Fakten und faire Diskussionen scheinen irrelevant zu sein.
- In einer Diskussion akzeptiert ein Stakeholder der Software-Abteilung eine mit dem Systemarchitekten diskutierte Anforderung nicht, weil er den Eindruck hat, dass Architekten keine Ahnung von der Realität der Arbeit von Programmierern haben.
- Ein Stakeholder akzeptiert die Einladung zu einem Workshop des Requirements Engineers nicht, da auch sein Erzfeind an dem Meeting teilnimmt.

Wenn ein Beziehungskonflikt der Grund für eine Diskussion ist, lässt sich beobachten, dass die Kommunikation zwischen den Konfliktparteien sich auf die Seite „Beziehungshinweis“ des Vier-Seiten-Modells von Schulz von Thun [Schulz von Thun 1981] konzentriert.

Konfliktparteien können die Diskussion über Anforderungen dazu nutzen, ihre Ablehnung gegenüber dem Verhalten der anderen Konfliktpartei auszudrücken. Fakten und faire Diskussionen scheinen den Konfliktparteien relativ unwichtig zu sein.

Vermeidungs- und Lösungsstrategien:

- Stakeholder sollten sich auf Meetingregeln und -verfahren einigen, wenn Emotionen hochkochen.
- Der Requirements Engineer sollte Anforderungen mit den beteiligten Konfliktparteien separat ermitteln und ohne dabei die Anforderungsquellen offenzulegen. Auf diese Weise wird die Ermittlungsaktivität objektiver.
- Beugen Sie dem negativen Verhalten der Konfliktparteien vor.
- Review und Einigung sollten ohne direkte Einbeziehung der Konfliktparteien oder von einer neutralen dritten Partei durchgeführt werden.

Da die meisten der hier beschriebenen Lösungsstrategien nicht im Verantwortungsbereich des Requirements Engineers liegen, kann er Beziehungskonflikte nur eskalieren und die anderen Stakeholder intervenieren lassen.

4.3 Konfliktlösung

Eine Voraussetzung für die Auswahl der passenden Konfliktlösungstechnik ist das tiefgreifende Verständnis des Anforderungskonflikts.

Basierend auf dieser Analyse und den Projektrandbedingungen kann der Requirements Engineer eine geeignete Lösungstechnik auswählen. Alle in diesem Abschnitt beschriebenen Techniken sind ähnlich wie die Ermittlungstechniken in Kapitel 3 strukturiert.

Die folgenden allgemeinen Konfliktlösungstechniken werden unterschieden (siehe [IREB2020]):

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

4.3.1 Einigung

Was ist das?

Die Einigung ist das Ergebnis einer Diskussion, in der eine vorhandene Konstellation von Anforderungen oder eine Konfliktlösung ohne Änderung aus mehreren verfügbaren Optionen ausgewählt wird. Um eine solche Lösung zu erreichen, ist es unverzichtbar, dass genügend Zeit zur Verfügung steht, um die Positionen aller Stakeholder vollständig zu verstehen und sie davon zu überzeugen, dass die gewählte Option auch ihre bevorzugte Wahl ist.

Die Rolle der Teilnehmer

Der Moderator kann eine schlichtende Rolle übernehmen und die Stakeholder daran erinnern, sich auf konstruktive und effiziente Weise zu besprechen. Er sollte den Optionen neutral gegenüberstehen und von der Gruppe in seiner Rolle akzeptiert werden. Die beteiligten Stakeholder sollten sich auf Fakten konzentrieren und erlauben, dass andere Teilnehmer Fragen stellen.

Vorbereitung

Bei einer sehr großen Gruppe ist es ratsam, einen Vertreter pro Gruppe zu benennen. Die Stakeholder müssen eingeladen sowie über die Agenda und den von ihnen erwarteten Beitrag zum Prozess informiert werden.

Anwendung

- Der Moderator definiert das Thema und legt eine Agenda mit einem Zeitplan fest.
- Der Moderator präsentiert die Informationen zum Konflikt und erklärt mögliche Lösungstechniken.
- Die Stakeholder präsentieren ihre Argumente ohne Unterbrechungen, die anderen Parteien können im Anschluss Fragen stellen.
- Der Moderator folgt den Diskussionen und schlägt eine geeignete Lösung vor. Wenn dies aufgrund des Themas, der Konfliktart oder aufgrund neu auftretender Konflikte nicht möglich erscheint, sollte der Moderator die Diskussion beenden und eine andere Lösungstechnik vorschlagen.

Ergebnisverarbeitung

Einer der Teilnehmer kann das Ergebnis für die anderen Teilnehmer zusammenfassen. Der Moderator sollte direkt auf eine informelle Zustimmung von allen Parteien hinwirken. Wenn die Optionen zu komplex sind, werden das Ergebnis und die betreffenden Anforderungen später vervollständigt und separaten Reviews unterzogen.

Typische Arbeitsprodukte

Argumente für und wider die verschiedenen Alternativen; die ausgewählte Lösung (einschließlich der von den Konfliktparteien vereinbarten Anforderungen).

Chancen

Bei einer Einigung besteht die Möglichkeit, dass sich die Konfliktparteien untereinander und auch ihre Anforderungen besser verstehen. Ein positives Ergebnis kann eine zusätzliche Motivation für die Gruppe darstellen, und es besteht eine gute Chance, dass das Ergebnis von Dauer ist.

Herausforderungen

Die Herausforderung besteht darin, die Essenz des Konflikts so darzulegen, dass jeder versteht, was zu diskutieren ist. Für den Moderator kann es eine Herausforderung sein, zu verhindern, dass sich die Diskussionskultur negativ auf das Ergebnis auswirkt und dass keine Einigung erreicht wird obwohl sie als wahrscheinlich erschien. Es ist eine Herausforderung,

die Stakeholder auf Faktenebene zu halten, die Diskussionszeit gleich aufzuteilen und im Zeitplan zu bleiben.

Varianten

Nicht zutreffend.

4.3.2 Kompromiss

Was ist das?

Ein Kompromiss ist das Ergebnis einer Diskussion, in der Aspekte einer vorhandenen Menge von Anforderungen oder Konfliktlösungen sowie neue Aspekte kombiniert werden, um neue Optionen zu schaffen. Um eine solche Konfliktlösung zu erreichen, ist es wichtig, dass genügend Zeit zur Verfügung steht, um ein gutes Verständnis der jeweiligen Positionen und aller Aspekte des Problems zu erreichen, damit eine Lösung ausgehandelt werden kann, die den Bedürfnissen aller gerecht wird.

Die Rolle der Teilnehmer

Der Moderator kann die Rolle übernehmen, die Stakeholder daran zu erinnern, dass sie die vereinbarten Regeln einhalten und auf konstruktive und effiziente Weise diskutieren. Er sollte den Optionen neutral gegenüberstehen und von der Gruppe in dieser Rolle akzeptiert werden. Die beteiligten Stakeholder sollten sich auf Fakten konzentrieren und erlauben, dass andere Teilnehmer Fragen stellen, damit sie alle, den Konfliktparteien wichtigen Aspekte umfassend verstehen.

Vorbereitung

Zunächst hätte eine Einigung in Betracht gezogen und als unmittelbare Option ausgeschlossen werden müssen. Beachten Sie, dass eine Einigung per definitionem nicht möglich ist, wenn es sich um einen Sachkonflikt handelt. Bei einem solchen Konflikt ist ein Kompromiss die erste Option, die in Betracht gezogen werden sollte.

Bei einer sehr großen Gruppe ist es ratsam, einen Vertreter pro Gruppe zu benennen. Die Stakeholder müssen eingeladen sowie über die Agenda und den von ihnen erwarteten Beitrag zum Prozess informiert werden.

Anwendung

- Der Moderator definiert das Thema und legt eine Agenda mit einem Zeitplan fest.
- Der Moderator präsentiert die Informationen zum Konflikt und erklärt mögliche Lösungstechniken.
- Die Stakeholder präsentieren ihre Argumente ohne Unterbrechungen, die anderen Parteien können im Anschluss Fragen stellen.
- Der Moderator rät den Konfliktparteien, zunächst über ihre wichtigsten Aspekte zu verhandeln.
- Sobald sich die Konfliktparteien auf einen bestimmten Aspekt verständigt haben, sollte der Moderator sicherstellen, dass diese Einigung dokumentiert wird.

- Der Moderator verfolgt die Diskussion und regt die Anwendung von Kommunikationsregeln an.

Ergebnisverarbeitung

Einer der Teilnehmer kann das Ergebnis für die anderen Teilnehmer zusammenfassen. Der Moderator sollte direkt auf eine informelle Zustimmung von allen Parteien hinwirken. Sind die Optionen zu komplex, so kann das Ergebnis (und die betreffenden Anforderungen) später vervollständigt und einem individuellen Review unterzogen werden.

Typische Arbeitsprodukte

Argumentation, weshalb eine Einigung keine geeignete Lösungstechnik war. Verschiedene diskutierte Aspekte, einschließlich der von den Konfliktparteien vereinbarten Anforderungen.

Chancen

Bei einem Kompromiss besteht die Möglichkeit, dass sich die Konfliktparteien selbst und auch ihre Anforderungen besser verstehen, ohne dass eine vollständige Einigung nötig ist.

Es können neue Aspekte integriert werden und mehr Stakeholder zur erstellten Lösung beitragen. Ein positives Ergebnis kann eine zusätzliche Motivation für die Gruppe darstellen, und es besteht eine gute Chance, dass das Ergebnis von Dauer ist.

Herausforderungen

Die Herausforderung besteht darin, die Essenz des Konflikts so darzulegen, dass jeder versteht, worum es bei der Diskussion geht. Für den Moderator kann es eine Herausforderung sein, zu verhindern, dass sich die Diskussionskultur negativ auf das Ergebnis auswirkt, beispielsweise wenn die dominante Partei ein besseres Ergebnis für sich, aber nicht für das gesamte Projekt erzielt. Der Moderator sollte daher dafür sorgen, dass die Stakeholder auf Faktenebene bleiben, die Diskussionszeit gleich verteilt ist und der Zeitplan eingehalten wird.

Varianten

Nicht zutreffend.

4.3.3 Abstimmung

Was ist das?

Eine Abstimmung ist das Ergebnis einer Auswahl aus einer bestehenden Konstellation von Anforderungen. Es ist entscheidend, dass die möglichen Optionen oder die Konstellation von Anforderungen von den Entscheidungsträgern gut verstanden werden. Damit Abhängigkeiten oder ein unausgeglichenes Machtverhältnis das Ergebnis nicht beeinflussen, sollte die Abstimmung geheim erfolgen.

Die Rolle der Teilnehmer

Die Entscheidungsträger sollten die Konsequenzen ihrer Wahl und die verfügbaren Optionen verstehen. Ein Moderator kann sie durch den Prozess führen und die Schritte des

Abstimmverfahrens, die verfügbaren Optionen und das Ergebnis der Abstimmung kommunizieren.

Vorbereitung

Stakeholder mit Entscheidungsbefugnis für den Sachverhalt müssen die Entscheidungsgewalt an die ausgewählten Stakeholder übertragen und zustimmen, das Ergebnis zu akzeptieren. Um zu verdeutlichen, worüber die Stakeholder abstimmen, sollten für jede Option Vorschläge erstellt werden. Ein neutrales Gremium für die Abstimmung muss gewählt werden, und es sind Stimmzettel mit entsprechenden Auswahloptionen vorzubereiten.

Anwendung

- Wenn entschieden wurde, dass ein Konflikt anhand der Lösungstechnik „Abstimmung“ gelöst werden soll, sollte jede in den Konflikt involvierte Partei einen Vorschlag vorbereiten, in der sie ihre Position und die möglichen Konsequenzen der vorgeschlagenen Lösung für die Entscheidungsträger beschreibt. Nachdem die Vorschläge vorbereitet und verteilt wurden, kann das Abstimmungsmeeting stattfinden.
- Im Abstimmungsmeeting sollte der Moderator den Entscheidungsträgern das Abstimmverfahren und die entsprechenden Positionen erklären.
- Optional kann jede Partei ihren Vorschlag präsentieren und die Fragen der Entscheidungsträger beantworten.
- Die Stimmzettel werden eingesammelt und ausgewertet.

Ergebnisverarbeitung

Die Ergebnisse der Abstimmung sollten dokumentiert werden.

Typische Arbeitsprodukte

Vorschläge für die unterbreiteten Optionen einschließlich Pro und Kontra sowie die ausgefüllten Stimmzettel.

Chancen

An Abstimmungen können sich zahlreiche Stakeholder beteiligen, was zu einer höheren Zufriedenheit der Stakeholder führen kann.

Herausforderungen

Unter Umständen ist es schwierig, komplexe Optionen so zu erklären, dass sie von jedem Entscheidungsträger verstanden werden. Die Entscheidung kann durch politische Trends beeinflusst werden.

Varianten

Bei einfachen Entscheidungen kann die Abstimmung spontan, ohne große Vorbereitung durchgeführt und rasch abgeschlossen werden. Es gibt verschiedene Werkzeuge zur Unterstützung der Entscheidungsfindung und der Beteiligung von weltweit verteilten Stakeholdern. Mit solchen Werkzeugen kann auch ein nicht synchrones Abstimmverfahren unterstützt werden.

4.3.4 Variantenbildung

Was ist das?

Die Variantenbildung ist das Ergebnis einer Integration von relevanten, abweichenden Anforderungen in eine Lösung, in der das System so konfiguriert werden kann, dass es jede Option unterstützt. Der Konfigurator oder der Benutzer des Systems kann dann während der Konfiguration oder zur Laufzeit ein Feature auswählen, das die gewünschte Konstellation von Anforderungen darstellt. Häufig gibt es zusätzliche Anforderungen, die implementiert werden müssen, damit ein Wechsel zwischen den verschiedenen Optionen möglich ist. Da jede Option – sowie der Wechsel zwischen den Optionen – ermittelt und gepflegt werden muss, könnte sich die Variantenbildung als kostspielig erweisen und zu komplizierten Systemen, mit selten genutzten Features führen.

Hinweis 4.3.1:

Der Requirements Engineer muss beurteilen, ob die Variantenbildung nicht eigentlich eine Flucht vor einem angemessenen Konfliktlösungsprozess ist, und ob sie den zusätzlichen Aufwand tatsächlich wert ist.

Die Konfiguration des Systems muss so implementiert werden, dass die Varianten unabhängig voneinander verwendbar sind und der Konflikt wirklich gelöst wird. Die Definition von Varianten ist oft die einzige praktikable Lösung für einen Sachkonflikt.

Die Rolle der Teilnehmer

Der Requirements Engineer sollte die Unterschiede zwischen den Varianten hervorheben, damit klar ist, weshalb eine einzige Lösungsoption nicht erreicht werden konnte.

Vorbereitung

Zuerst sollte man Techniken wie Einigung, Kompromiss und Abstimmung in Erwägung gezogen haben. Alle Parteien sollten darin übereinstimmen, dass die Erstellung einer Variante, die in Zukunft zusätzlichen Aufwand verursacht, die einzig mögliche Lösung ist.

Anwendung

- Mit jeder Partei könnten separate Meetings organisiert werden, um die betreffenden Anforderungen für die verschiedenen Optionen zu ermitteln, damit sie sich nicht gegenseitig beeinträchtigen.

- Nachdem die einzelnen Optionen festgelegt wurden, führen die Parteien ein Review der Optionen durch, um sie besser aufeinander abzustimmen. Wenn die Abstimmung besser ist, sollten die verschiedenen Optionen kombiniert werden.
- Ist keine weitere Abstimmung möglich, sollten die Anforderungen für den Wechsel zwischen den Varianten festgelegt werden.

Ergebnisverarbeitung

Die Ergebnisse der Variantenbildung sollten dokumentiert werden.

Typische Arbeitsprodukte

Im Idealfall sind die Arbeitsergebnisse die betreffenden Anforderungen für die jeweilige Option und die Anforderungen für den Mechanismus zur Auswahl der Optionen.

Chancen

Durch die Definition von Varianten kann die Integration der speziellen Bedürfnisse unterschiedlicher Stakeholder erreicht werden. So lässt sich die Beteiligung der Stakeholder verbessern und ihre Zufriedenheit steigern.

Herausforderungen

Es kann schwierig sein, Optionen zu entwickeln, die wirklich notwendig sind, und die den zusätzlichen Aufwand im Entwicklungsprozess minimieren.

Varianten

Die Definition der Abweichung kann sehr früh oder spät im Produktlebenszyklus erfolgen. Sie kann sich von der Konfiguration während der Laufzeit zur Konfiguration während der Vorbereitung unterscheiden oder zu völlig anderen Produkten oder Produktfamilien. Es ist wichtig, die richtige Abweichung zu finden, um ein gutes Gleichgewicht zwischen Aufwand (überwiegend Kosten) und individuellen Bedürfnissen von Stakeholdern zu erreichen.

4.3.5 Ober-sticht-Unter

Was ist das?

Ober-sticht-Unter ist das Ergebnis einer Auswahl aus einer bestehenden Konstellation von Anforderungen. Es ist entscheidend, dass die möglichen Optionen oder die Konstellation von Anforderungen vom Entscheidungsträger gut verstanden werden. Da Entscheidungsträger alle beteiligten Konfliktparteien überstimmen können (Situation „Ober sticht Unter“), beeinflusst ein unausgeglichenes Machtverhältnis das Ergebnis nicht.

Die Rolle der Teilnehmer

Entscheidungsträger sollten die Konsequenzen ihrer Wahl und die verfügbaren Optionen verstehen. Sie sollten die Entscheidung auch rechtfertigen, damit sie von allen beteiligten Konfliktparteien akzeptiert wird. Es können Experten konsultiert werden, um Argumente für die verschiedenen Optionen zu sammeln und die Entscheidungsträger mit dem nötigen Wissen zu unterstützen.

Vorbereitung

Es sollte ein Entscheidungsträger gefunden werden, der von allen beteiligten Konfliktparteien akzeptiert wird. Konfliktparteien oder Experten bereiten Vorschläge mit der Erklärung aller Optionen vor, die der Ober–sticht–Unter–Situation unterliegen.

Anwendung

- Der Entscheidungsträger muss die Vorschläge lesen und hat die Möglichkeit, Fragen zu stellen.
- Optional kann jede Partei ihren Vorschlag persönlich präsentieren und die Fragen der Entscheidungsträger beantworten.
- Der Entscheidungsträger verkündet die Entscheidung.

Ergebnisverarbeitung

Die Ergebnisse und die Gründe sind zu dokumentieren.

Typische Arbeitsprodukte

Ein Dokument mit den Vorschlägen und der ausgewählten Option sowie die Begründung für die Entscheidung.

Chancen

Wenn es keine andere Möglichkeit gibt, kann die Hierarchie einer Organisation zur Lösungsfindung genutzt werden.

Herausforderungen

Es kann herausfordernd sein, einen Vorschlag zu erstellen, der sämtliche Fakten, Bedürfnisse und Konsequenzen der verschiedenen Optionen beinhaltet.

Varianten

Ober–sticht–Unter kann von einem Gremium aus Entscheidungsträgern durchgeführt werden. Anstelle des Managers der Konfliktparteien kann ein neutraler Experte, der von allen Parteien akzeptiert wird, die Entscheidung treffen.

4.3.6 Unterstützende Techniken

Zusätzlich gibt es verschiedene unterstützende Techniken, zum Beispiel:

- Gewaltfreie Kommunikation [Rosenberg2015]
- Verhandlungstechniken [FiUP2012]
- Consider–All–Facts [DeBono2006]
- Plus–Minus–Interesting [DeBono2006]
- Entscheidungsmatrix [BiAB2006] [IsNe2013]

4.3.7 Wahl einer geeigneten Konfliktlösungstechnik

Ausgehend von den Konflikteigenschaften, sollten passende Konfliktlösungstechniken ausgewählt werden.

	Einigung	Kompromiss	Variantenbildung (Konfigurierbarkeit)	Abstimmung	Ober-sticht-Unter	Legende:	
						+ empfohlen 0 anwendbar, die Entscheidung sollte auf Basis der anderen Randbedingungen getroffen werden - nicht empfohlen	
1	Große Anzahl von Stakeholdern und/oder unterschiedlichen Meinungen	-	-	0	+	+	Das Anhören aller Argumente könnte zu zeitaufwendig sein und wäre schwierig zu erreichen.
2	Hohe Kritikalität der Situation ¹⁰	+	-	-	-	0	Entscheidungen sind von allen Stakeholdern gründlich zu durchdenken und zu akzeptieren.
3	Weite Verteilung der Stakeholder	-	-	+	+	+	Siehe 1. Es könnte eine kluge Entscheidung sein, individuelle Varianten zu erstellen, um verschiedenen (lokalen) Bedürfnissen gerecht zu werden. Abstimmung und Ober-sticht-Unter können virtuell erfolgen.
4	Hoher Zeitdruck bei der Konfliktlösung	-	-	-	+	+	Die Definition von Varianten nimmt zu viel Zeit in Anspruch, da zwei Varianten und der Konfigurationsmechanismus definiert werden müssen.

¹⁰ D. h. das falsche Lösungsergebnis hat ein hohes Risiko für die Implementierung zur Folge.

		Einigung	Kompromiss	Variantenbildung (Konfigurierbarkeit)	Abstimmung	Ober-sticht-Unter	Legende:
							+ empfohlen 0 anwendbar, die Entscheidung sollte auf Basis der anderen Randbedingungen getroffen werden - nicht empfohlen
5	Die Klarheit des Ergebnisses ist wichtig	+	+	-	0	+	Durch das Definieren einer Variante ist es nicht leicht, den Produktinhalt zu erklären.
6	Geringe soziale Kompetenz der Stakeholder ¹¹	-	-	+	+	+	Einigungen und Kompromisse erfordern soziale Kompetenz, um Meinungen auszutauschen.
7	Komplizierte Situation ¹²	-	+	-	-	0	In komplizierten Situationen muss sich eine Konfliktpartei auf Aspekte konzentrieren, die sie nachvollziehen kann. Das Kombinieren von Kompetenzen und das Akzeptieren verschiedener Aspekte führt zu Kompromissen. Falls ein Experte mit der notwendigen Kompetenz Entscheidungsträger ist, ist Ober-sticht-Unter möglich.
8	Lange Lebensdauer der Ergebnisse erforderlich	+	-	+	-	-	Faule Kompromisse, Abstimmungen und Ober-sticht-Unter können dazu führen, dass Konfliktparteien das Ergebnis nicht akzeptieren und es im Laufe der Zeit ändern möchten.

¹¹ Stakeholder sind nicht in der Lage oder willens, einander zuzuhören und nicht fähig, die Meinungen anderer zu akzeptieren.

¹² Der Sachverhalt ist so kompliziert, dass nicht alle Stakeholder die Konsequenzen verstehen können.

	Einigung	Kompromiss	Variantenbildung (Konfigurierbarkeit)	Abstimmung	Ober-sticht-Unter	Legende: + empfohlen 0 anwendbar, die Entscheidung sollte auf Basis der anderen Randbedingungen getroffen werden - nicht empfohlen
9 Geringe Motivation der Stakeholder (zur aktiven Beteiligung)	-	-	+	+	+	Einigungen und Kompromisse erfordern Zeit für Diskussionen und Beteiligung, daher ist eine hohe Motivation nötig.
10 Schlechte zeitliche Verfügbarkeit der Stakeholder	-	-	0	+	+	Einigungen und Kompromisse erfordern Zeit für Diskussionen und Beteiligung. Je nach Beschaffenheit der Variante kann es weniger zeitintensiv sein, nur die eigene Variante anstelle des gesamten Systems zu definieren.
11 Widersprüchliche Sachbedürfnisse	-	0	+	-	-	Ein Kompromiss, sofern er überhaupt möglich ist, mag einfacher zu entwickeln sein, wird aber in den meisten Fällen mit höheren laufenden Kosten verbunden sein. Während der Lebensdauer des Systems können Varianten die bessere Lösung sein.
12 Datenkonflikt	+	+	0	+	-	Je nach Inhalt, können Varianten eine geeignete Technik sein.
13 Interessenskonflikt	-	0	+	0	+	Der Konflikt lässt sich nicht wirklich durch eine Abstimmung lösen, da immer die Mehrheit gewinnt.
14 Wertekonflikt	-	-	+	-	0	Da Werte nicht kurzfristig änderbar sind, ist das Erstellen von Varianten die einzige Möglichkeit, um alle Wertmaßstäbe zu erfüllen. Wenn das Erstellen von Varianten keine geeignete Technik darstellt, ist Ober-sticht-Unter die einzige Option.

	Einigung	Kompromiss	Variantenbildung (Konfigurierbarkeit)	Abstimmung	Ober-sticht-Unter	Legende:
						+ empfohlen 0 anwendbar, die Entscheidung sollte auf Basis der anderen Randbedingungen getroffen werden - nicht empfohlen
15 Strukturkonflikt	-	0	-	+	+	<p>Die negativen Effekte des strukturellen Konflikts verhindern eine ehrliche Einigung und eine faire Diskussion, die eine Einigung oder einen Kompromiss unwahrscheinlich machen. Der Kompromiss könnte jedoch die einzige verbleibende Konfliktlösungstechnik sein.</p> <p>Da Abstimmungen anonym ablaufen und für Ober-sticht-Unter eine Instanz nötig ist, die über beiden Konfliktparteien steht, wird die Verteilung der Macht minimiert.</p> <p>Liegt der Konflikt in der Verteilung der Mehrheit, stellt eine Abstimmung keine geeignete Technik dar.</p>
16 Beziehungskonflikt	-	-	-	-	-	<p>Es gibt keine geeignete Technik, oder die Verantwortung für die Lösung dieser Art von Konflikt liegt nicht beim Requirements Engineering.</p>

4.4 Dokumentation der Konfliktlösung

Nach der Konfliktauflösung sollte der Konflikt genau dokumentiert werden. Abgesehen von den in Abschnitt 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, ein darunterliegender sozialer Konflikt jedoch nicht.

5 Fähigkeiten eines Requirements Engineers

In diesem Kapitel erhalten Sie einen Überblick über wichtige Fähigkeiten, die ein Requirements Engineer mitbringen sollte. Da Kommunikation ein zentrales Element des Requirements Engineering ist, werden wichtige Kommunikationstheorien vorgestellt, die ein tieferes Verständnis der Mechanismen ermöglichen, die Kommunikation antreiben. Da persönliche Entwicklung ein laufender Prozess ist, werden in den letzten drei Abschnitten schließlich die Themen Selbstreflexion, Möglichkeiten der persönlichen Weiterentwicklung und lebenslanges Lernen behandelt.



Der zunehmende Einsatz generativer Künstlicher Intelligenz (KI) und sogenannter Large Language Models (LLM) im Requirements Engineering erfordert vom Requirements Engineer neue grundlegende Fertigkeiten. Dazu zählt insbesondere das Verständnis der Funktionsweise und Einsatzmöglichkeiten generativer KI sowie deren praktische Anwendung in der täglichen Arbeit. KI-Systeme können vergangene Projektdaten und Kommunikationsmuster systematisch analysieren, wodurch individuelle Stärken und mögliche Optimierungsfelder sichtbar werden. Requirements Engineers, die den Umgang mit generativer KI sicher beherrschen, sind in der Lage, aus diesen datenbasierten Erkenntnissen konkrete Maßnahmen zur persönlichen Weiterentwicklung abzuleiten. Dabei ergänzt KI die menschliche Kompetenz, ersetzt sie aber keinesfalls. Die Erfahrung, Intuition und ethische Verantwortung des Requirements Engineers bleiben unverzichtbar, um KI-generierte Ergebnisse in den richtigen Zusammenhang zu setzen und fundierte Entscheidungen zu treffen. Die Beherrschung generativer KI wird somit zu einer essenziellen Grundfertigkeit, die es dem Requirements Engineer erlaubt, effizienter und zielgerichteter zu arbeiten, während er gleichzeitig die Einhaltung rechtlicher Vorgaben und ethischer Prinzipien gewährleistet.

5.1 Benötigte Fähigkeiten im Bereich Ermittlung

Die in den vorangehenden Kapiteln erläuterten Ansätze und Techniken bilden das grundlegende Kompetenzspektrum oder Skillset für jeden Requirements Engineer. Die Beherrschung dieser (Hard-) Skills ist eine Voraussetzung für Practitioner und Specialists. Doch damit nicht genug: Um erfolgreich zu sein, muss der Requirements Engineer auch über eine Reihe von Soft Skills verfügen (siehe z.B. [GleA2020], [Klaus2007]), wie z.B. Kommunikationsfähigkeit, analytisches Denken, Einfühlungsvermögen, Konfliktlösungsfähigkeit, Moderationsfähigkeit, Selbstvertrauen und Überzeugungsfähigkeit.

Marcel Robles [Robles2012] bietet einen Überblick über die zehn wichtigsten Soft Skills:

- Kommunikation – mündliche und schriftliche Ausdrucksfähigkeit, Redefähigkeit, Präsentieren, Zuhören
- Höflichkeit – Umgangsformen, Etikette, Geschäftsetikette, kultiviert, sagt Bitte und Danke, respektvoll
- Flexibilität – Anpassungsfähigkeit, Veränderungsbereitschaft, lebenslanger Lerner, akzeptiert Neues, stimmt sich ab, ist gelehrig
- Integrität – ehrlich, ethisch, hohe Moralvorstellungen, hat persönliche Werte, tut das Richtige

- Zwischenmenschliche Fähigkeiten – nett, sympathisch, humorvoll, freundlich, fürsorglich, empathisch, verfügt über Selbstkontrolle, geduldig, Kontaktfreude, Herzlichkeit, soziale Kompetenz
- Positive Einstellung – optimistisch, enthusiastisch, aufmunternd, zufrieden, zuversichtlich
- Professionalismus – sachlich, gut gekleidet, Erscheinungsbild, gelassen
- Verantwortung – verantwortlich, zuverlässig, tatkräftig, einfallsreich, selbstdiszipliniert, möchte gute Arbeit abliefern, gewissenhaft, gesunder Menschenverstand
- Teamwork – kooperativ, kommt mit anderen zurecht, angenehm, unterstützend, hilfsbereit, teamorientiert
- Arbeitsethik – fleißig, arbeitsfreudig, loyal, initiativ, selbstmotiviert, rechtzeitig, gutes Anwesenheitsverhalten

Während alle oben genannten Eigenschaften für jeden Berufstätigen relevant sind, sind in Zusammenhang mit der Anforderungsermittlung zudem die folgenden Eigenschaften besonders wichtig:

- Kontextbewusstsein – den Kontext kennen, in dem man arbeitet und den Ansatz entsprechend daran anpassen
- ethisches Gewissen – jede Technologie kann sich negativ auf Menschen, die Gesellschaft und die Umwelt auswirken. Erkennt der Requirements Engineer eine solche negative Auswirkung, dann sollte er sie klar formulieren, um eine Diskussion zur Minderung der Auswirkung anzuregen.
- interkulturelle Kompetenz – die Fähigkeit, in unterschiedlichen (Geschäfts-, anwendungsgebietsbezogenen, regionalen usw.) Kulturen und mit Menschen verschiedener Kulturen zu arbeiten
- Führungspersönlichkeit – die Fähigkeit, Stakeholder zu einem bestimmten Ziel zu führen
- motivierendes Wesen – die Fähigkeit, Stakeholder für ein bestimmtes Ziel zu inspirieren
- Neutralität – die Fähigkeit, sämtliche (relevanten) Stakeholder und deren Interessen gleichermaßen zu bedienen, ohne persönliche Interessen zu verfolgen
- Reflexion – die Fähigkeit, Feedback anzunehmen und Situationen und das eigene Verhalten zu bewerten
- Ichbewusstsein – die eigene Position in Bezug auf die Stakeholder kennen und den eigenen Ansatz entsprechend anpassen

Abhängig von dem betreffenden Geschäftsfeld, dem Projektkontext und den Stakeholdern, sind gewisse Fähigkeiten mehr gefragt als andere. Trotzdem sind alle wichtig: Kontextbewusstsein bezieht sich in der Tat auf die Fähigkeit, die richtigen Fähigkeiten zur richtigen Zeit einzusetzen, wenn eine bestimmte Situation sie verlangt.

Von all diesen Soft Skills ist die Kommunikationsfähigkeit die Schlüsseleigenschaft des Requirements Engineers – es ist kein Zufall, dass Robles diese Fähigkeit an erster Stelle nannte.

Jede Interaktion zwischen Requirements Engineer und Stakeholdern (als der Hauptquelle für Anforderungen) ist eine Form von Kommunikation, in der die meisten, wenn nicht gar alle zuvor genannten Fähigkeiten, eine Rolle spielen. Einige Fähigkeiten, beispielsweise Integrität und Neutralität, sind für die Kommunikation an sich wichtig; andere, wie Führungspersönlichkeit und Teamwork, können nur durch Kommunikation umgesetzt werden.

5.2 Kommunikationstheorie und Kommunikationsmodelle

Kommunikation kann als eine Möglichkeit für Einzelpersonen gesehen werden, Nachrichten auszutauschen und sinngebend zu wirken. Sie beinhaltet jedes Verhalten einer Person mit dem Ziel, einer anderen Person Informationen zu verdeutlichen, welche die Person aufnimmt und interpretiert. Darüber hinaus wird Kommunikation als ein komplexer zwischenmenschlicher Prozess betrachtet, bei dem jede beliebige Kombination aus gesprochener Sprache, Schreiben und anderen Signalen als Basis für den Austausch von Konzepten, Gedanken, Meinungen und Informationen verwendet wird.

Kommunikation ist effektiv, wenn der Austausch dazu führt, dass die beabsichtigte und die aufgenommene Bedeutung übereinstimmen. Effektive Kommunikation ist nicht notwendigerweise effizient in dem Sinne, dass das gewünschte Ergebnis mit einem Minimum an Aufwand, Zeit, Komplexität und investierten Ressourcen erreicht wird. Da gute Kommunikation faktisch sehr stark auf Redundanz angewiesen ist, kann mehr Effizienz ganz schnell zu weniger Effektivität führen. Der Requirements Engineer sollte sicherstellen, dass die Kommunikation mit Stakeholdern innerhalb angemessener Grenzen von Effizienz effektiv ist.

Häufig ist Kommunikation zielgerichtet: Der Sender hat die Absicht, auf Empfängerseite ein bestimmtes Verhalten hervorzurufen. Bei der Anforderungsermittlung und Konfliktlösung versucht der Requirements Engineer, Stakeholder so zu motivieren, dass sie sich offen und teamorientiert verhalten und alle relevanten Informationen uneingeschränkt äußern.

Erfolgreiche Anforderungsermittlung und Konfliktlösung sind in einem hohen Maß vom richtigen Verständnis der Grundlagen der Kommunikation abhängig. Nachfolgend werden verschiedene Modelle aus dem Gebiet der Kommunikationstheorie dargestellt, jeweils mit ihrer Relevanz für die Anforderungsermittlung.

Das Shannon–Weaver–Modell [ShWe1971] wird häufig als „Mutter aller Kommunikationsmodelle“ betrachtet. Der Fokus liegt dabei auf der *Kodierung* (Verschlüsselung) einer *Nachricht* von einem *Sender* an einen *Empfänger*, der sie nach der *Übertragung* über einen bestimmten *Kanal dekodiert* (entschlüsselt), wobei das Risiko einer *Störung* besteht, welche die Nachricht verfälschen kann (siehe Abbildung 25).

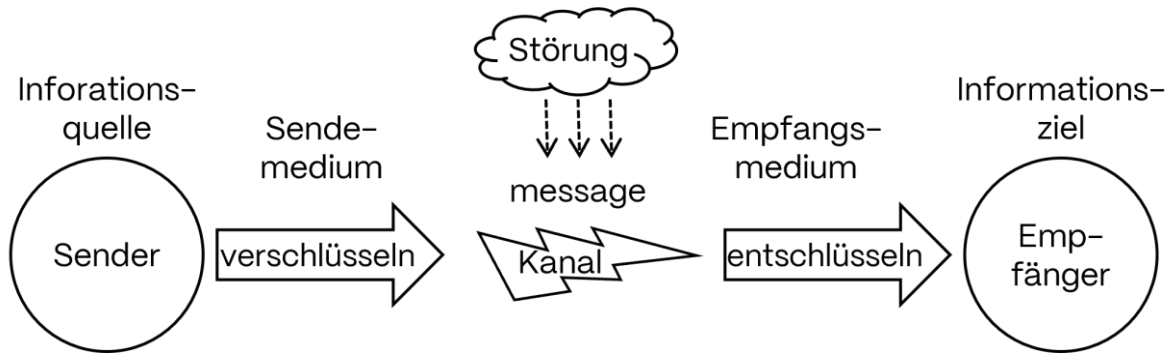


Abbildung 25: Das Shannon-Weaver-Kommunikationsmodell

Beim Requirements Engineering sind es oftmals „virtuelle“ Störungen, die Aufmerksamkeit verlangen, da diese zu falschen Anforderungen und Konflikten führen können: Voreingenommenheit, Gerede, Gerüchte, geheime Agenden. Der Requirements Engineer sollte in der Lage sein, diese Art von Störungen zu erkennen, die Kompatibilität von Kodierung und Dekodierung („Verstehen wir einander tatsächlich?“) zu prüfen und die geeigneten Kanäle (z. B. Meetings, Präsentationen, Veröffentlichungen) für die Kommunikation mit den Stakeholdern auszuwählen.

Schramm [Schramm1971] hat zwei Modelle zur Kommunikationstheorie beigetragen.

Kommunikation wird als eine *soziale Interaktion* zwischen einem Sender und einem Empfänger betrachtet. In Schramms erstem Modell (siehe Abbildung 26) wird deutlich, dass eine Nachricht nur dann erfolgreich kommuniziert werden kann, wenn sie in einen geteilten Erfahrungsbereich passt. Schramm hat auch darauf hingewiesen, dass man die Auswirkungen (sowohl gewünschte als auch unerwünschte) nachprüfen sollte, die eine Nachricht auf den Adressaten der Nachricht hat.

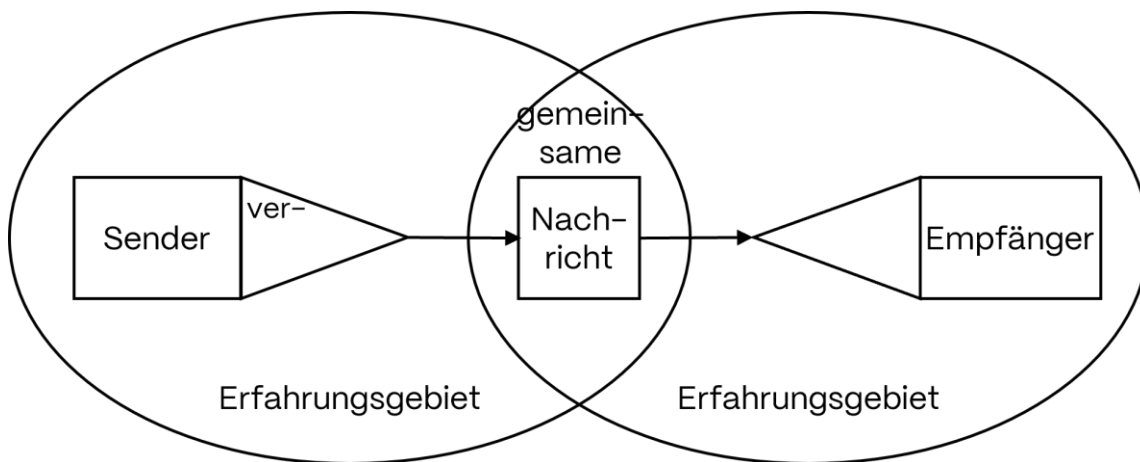


Abbildung 26: Das Kommunikationsmodell nach Schramm über die Bereiche geteilter Erfahrungen

Der Requirements Engineer sollte prüfen, ob das notwendige Teilen in der Beziehung zu den Stakeholdern vorhanden ist. Fehlt dies, könnte das zu einem Misserfolg führen. Der Requirements Engineer kann den Bereich geteilter Erfahrung vergrößern, beispielsweise

indem er sich Fachkenntnisse im Selbststudium aneignet oder indem er Stakeholder zu Problemen mit Anforderungen schult.

Schramm hat außerdem das *Zirkulationsmodell* der Kommunikation entwickelt (siehe Abbildung 27). In diesem Modell verschlüsselt der Sender eine Nachricht, die vom Empfänger entschlüsselt und interpretiert wird. Der Empfänger reagiert dann darauf, indem er eine andere Nachricht verschlüsselt und übergibt. Dieses Muster lässt sich bei der Anforderungsermittlung beobachten, beispielsweise wenn der Requirements Engineer einem Stakeholder Fragen stellt, sich die Antworten anhört und diese anschließend zusammenfasst. In dieses Modell passt die Idee des Feedbacks.

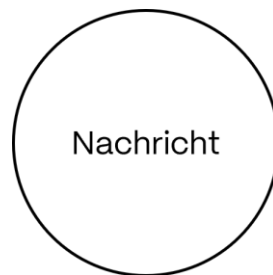


Abbildung 27: Schramms Zirkulationsmodell der Kommunikation

Das Vier-Seiten-Modell der Kommunikation von Schulz von Thun [ScTh1981] stellt die *Nachricht* in den Mittelpunkt und beschreibt *vier Aspekte*, die beachtet werden sollten (siehe Abbildung 28).

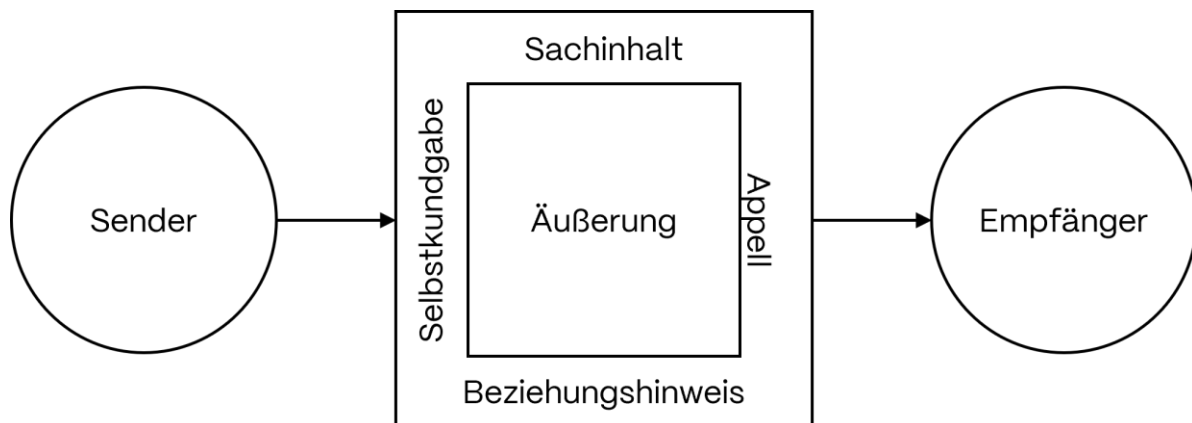


Abbildung 28: Kommunikationsmodell nach Schulz von Thun

Der Requirements Engineer muss sich all dieser Aspekte bewusst sein:

- *Sachinhalt*: Der Sachinhalt einer Nachricht ist die Grundlage für die Ermittlung und Konsolidierung von Anforderungen.
- *Selbstkundgabe*: Dies bezieht sich auf das Maß, in dem ein Stakeholder einer bestimmten Anforderung verpflichtet ist; der Requirements Engineer sollte hingegen strikte Neutralität wahren.

- *Beziehung*: Der Requirements Engineer sollte sich um eine offene und konstruktive Beziehung zu den Stakeholdern bemühen und die Perspektive der Stakeholder verifizieren.
- *Appell*: Der Aspekt „Appell“ kann Aufschluss darüber geben, was der Stakeholder vom Requirements Engineer erwartet, und er bietet Feedback zum kommunizierten Sachinhalt.

All diese Modelle tragen zu einem besseren Verständnis der Kommunikation zwischen Requirements Engineer und Stakeholdern bei. Eine gängige Idee ist, dass es bei Kommunikation darum geht, *aussagekräftige Konzepte* zwischen Individuen zu *teilen*. Der entscheidende Punkt ist, dass während dieser Kommunikation Informationen verloren gehen, hinzugefügt, verzerrt oder fehlinterpretiert werden können. Diese sogenannten *Transformationseffekte* (siehe 3.3.3) führen dazu, dass die betreffenden Individuen Konzepte unterschiedlich erfassen.

Der Requirements Engineer sollte die Qualität seiner eigenen Kommunikation sicherstellen, indem er diese Effekte abmildert. Es ist unverzichtbar, dem Kodieren und Dekodieren von Nachrichten die entsprechende Aufmerksamkeit zu schenken, indem man die richtigen Kanäle auswählt, Störungen vermeidet und die korrekte Interpretation durch das Einholen von *Feedback* überprüft.

Redundanz ist die primäre Lösung für Kommunikationsprobleme. Das bedeutet in der Regel, dass dieselben Informationen mehrmals übertragen werden, oftmals über verschiedene Kanäle. Beispiele dafür sind ein von einer Erklärung begleitetes Diagramm oder eine Präsentation, die durch nonverbale Gesten untermalt wird. Wenn redundante Nachrichten konsistent sind, und nur dann, unterstützen sie die Qualität der Kommunikation.

Hinweis 5.2.1:

Machen Sie in Ihrer gesamten Kommunikation von Redundanz Gebrauch.

Bitten Sie immer um Feedback.

Achten Sie auf Transformationseffekte.

Versuchen Sie, die Stakeholder dazu anzuregen, fragwürdige Aussagen in anderen Worten widerzugeben, anhand von konkreten Beispielen oder mithilfe sonstiger Medien wie Zeichnungen oder Metaphern anders auszudrücken.

5.3 Selbstreflexion zu den eigenen Fähigkeiten in der Anforderungsermittlung

„Ich weiß, dass ich nichts weiß.“ (Sokrates)

„Ich weiß nichts, ich komme aus Barcelona“ (Manuel, in der britischen Serie „Fawlty Towers“)

Der Lehrplan, dieses Handbuch und jedes begleitende Training legen die Grundlage für die erfolgreiche Anwendung der vorgestellten Techniken. Die Entwicklung und Verbesserung persönlicher Fähigkeiten in der Anforderungsermittlung ist jedoch ein langfristiger Lernprozess.

Die Grundlage für Verbesserungen bildet die Selbstreflexion. Es gibt drei Arten der Reflexion:

1. Das Nachdenken darüber, was Sie geplant haben und diesen Plan kritisch zu überdenken: Könnte man es auch anders machen? Was erwarte ich von den geplanten Aktivitäten? Was möchte ich lernen? Kann ich mich auf diese Weise verbessern? (prospektive Reflexion)
2. Das Nachdenken darüber, wie ich aktuell meine Arbeit verrichte: Verhalte ich mich professionell? Wende ich an, was ich gelernt habe? Welche Ängste und Hoffnungen habe ich derzeit in Bezug auf Ermittlungs- und Konfliktlösungstätigkeiten, an denen ich beteiligt bin? (begleitende Reflexion)
3. Das Nachdenken darüber, was ich getan habe und wie ich die Ermittlungs- und Konfliktlösungsaufgaben in der Vergangenheit ausgeführt habe (retrospektive Reflexion)



Abbildung 29: Drei Arten der Reflexion

Selbst wenn das Requirements Engineering während eines Entwicklungsprojekts als Erfolg gewertet wird, gibt es normalerweise verschiedene Möglichkeiten der Verbesserung.

Hinweis 5.3.1:

Hier ist eine kurze Checkliste mit Fragen, die Ihnen dabei helfen, Ihren Prozess der Selbstreflexion zu beginnen:

- Hat eine Technik die erwarteten Ergebnisse geliefert oder zur Entwicklung beigetragen? Falls ja, was war der ausschlaggebende Erfolgsfaktor? Falls nein, was war der Grund für den Misserfolg? Was hätte ich anders machen können? Was sollte ich beim nächsten Mal besser machen?
- Haben die Stakeholder die angewendeten Ermittlungs- oder Konfliktlösungstechniken akzeptiert? Falls ja, was war der Grund für ihre Kooperation? Falls nein, worin bestand das Problem? Was hätte ich anders machen können? Was sollte ich beim nächsten Mal besser machen?
- War der Aufwand für eine gewählte Technik im Vergleich zu ihrem Beitrag zum Projekterfolg gerechtfertigt? Falls nicht, warum nicht? War die Technik in unserem Kontext ein Problem (d. h. ich hätte eine besser geeignete Technik auswählen müssen) oder was ist mir bei der richtigen Anwendung der Technik misslungen?
- Welche Technik hätte es erlaubt, Anforderungen, die erst spät im Projekt aufkamen, früher zu erheben? Was war der Grund dafür, dass sie nicht früher identifiziert wurden?
- Welche alternativen Techniken hätten außerdem angewendet werden können? Wäre es förderlich, bestimmte Ermittlungs- oder Konfliktlösungstechniken bei einem künftigen Projekt in einer ähnlichen Situation zu berücksichtigen? Müssen dazu zusätzliche Kenntnisse und Erfahrungen mit diesen Techniken erworben werden (vgl. Abschnitt 5.4)?

Als fortgeschrittener Requirements Engineer werden Sie nach zahlreichen Sitzungen zur Selbstreflexion immer mehr Fragen entwickeln und sich auf bestimmte Aspekte usw. konzentrieren, bis Sie Ihre eigene, persönliche Checkliste erarbeitet haben. Oder – wenn Sie zum Experten in Selbstreflexion geworden sind – Sie können die Themen identifizieren, die im Verlauf Ihrer Arbeit reflektiert werden müssen. Als Anfänger (in der Selbstreflexion unerfahren) sollten Sie sich jedoch vor einer zu starken Vereinfachung in Acht nehmen: Wenn Sie auf dem Bett liegen, um über sich nachzudenken und professionelle Selbstreflexion zu betreiben, könnte dies in ein entspannendes Mittagsschläfchen münden, anstatt Ihnen Erkenntnisse über sich selbst und Ihre Kompetenz in Anforderungsermittlung zu bringen!

Hinweis 5.3.2:

Daher finden Sie nachfolgend einige Tipps zur Reflexion für Einsteiger:

- Beginnen Sie mit Ihrer Aktivität der Selbstreflexion bei einer guten Tasse Kaffee oder Tee (oder Ihrem bevorzugten Getränk); Ihre Notizen und Inputdaten haben Sie zur Hand (siehe Hinweis 5.3.3).
- Planen Sie Ihren Selbstreflexionsprozess: Über welche Frage möchten Sie zuerst nachdenken? Verwenden Sie zu Beginn eine vorhandene Checkliste mit Reflexionsfragen, beispielsweise diejenige aus Hinweis 5.3.1.
- Machen Sie sich Notizen zum Ergebnis Ihres Denkvorgangs.
- Bleiben Sie beim gewählten Thema! Vermeiden Sie es, das Thema zu wechseln, bevor Sie nicht alle zugehörigen Aspekte betrachtet haben. Berücksichtigen Sie systematisch jeden Aspekt der ausgewählten Frage.

- Denken Sie nicht nur über Probleme und Fehler nach und wie Sie diese in Zukunft vermeiden können; erfassen Sie auch Kriterien und gewonnene Erkenntnisse in Bezug auf Erfolge/positive Ergebnisse.
- Setzen Sie Ihren Denkvorgang in Beziehung zu bekannten Konzepten und etablierten Theorien. Halten Sie, wenn nötig, fest, dass Sie anschließend mehr über eine bestimmte Methode lernen müssen. Es ist auch möglich, dass Sie ein Lehrbuch (oder das Internet) zurate ziehen, um eine durchgeführte Anforderungsermittlungs- oder Konfliktlösungsstätigkeit zu beurteilen. Achten Sie darauf, dass Sie sich nicht in den Details einer schriftlichen Quelle verlieren. Kehren Sie schnell zur Reflexionsfrage zurück, die der Grund für die Exkursion in die Literatur war.

Hinweis 5.3.3:

Als Input für Ihre Selbstreflexion können Sie zunächst Daten über sich und Ihr berufliches Verhalten sammeln:

- Nehmen Sie, wenn möglich, ein Video oder eine Audioaufzeichnung einer von Ihnen durchgeführten Ermittlungsaktivität zur Hand (z. B. ein Interview).
- Bitten Sie Ihre Peergroup um Feedback zu bestimmten Aspekten Ihrer beruflichen Leistung. Ein Kollege könnte Sie beispielsweise bei einem Interview mit einem Stakeholder oder beim Moderieren eines Workshops beobachten. – Oder ein Experte analysiert die Notizen zu Ihren Beobachtungen oder beurteilt einen Fragebogen, den Sie erstellt haben. Durch das Lesen der Erkenntnisse und deren Diskussion mit dem Beobachter/Sachverständigen können Sie wertvollen Input für die Selbstreflexion erhalten.
- Bitten Sie Ihre Stakeholder um Feedback. Dies kann durch ein kurzes Interview (beispielsweise unmittelbar nach einem Interview zur Anforderungsermittlung) oder anhand eines Fragebogens erfolgen. Achten Sie jedoch darauf, dass Sie die Ermittlungs- und Konfliktlösungsaktivitäten nicht vermischen, wenn Sie um Feedback zu Ihrem Verhalten bitten. Im Falle von Konflikten oder mangelnder Bereitschaft, Informationen zum Ermittlungsthema zu liefern, sind Stakeholder möglicherweise auch nicht bereit, Ihren Selbstreflexionsprozess zu unterstützen.
- Wenden Sie einen personenzentrierten Ansatz auf Ihre eigene Arbeit an: Wenn Sie beispielsweise einen Fragebogen testen, den Sie mit einigen vorläufigen Stakeholdern entwickelt haben, kann Ihnen dies nicht nur Einblicke gewähren, die den Fragebogen an sich besser machen, sondern Ihnen auch als Input für die Reflexion über Ihre Leistung dienen.
- Bitten Sie einen erfahrenen Requirements Engineer, Sie zu coachen.
- Wenden Sie die bekannte Tagebuch-Technik auf Ihre eigene Arbeit an. Wenn Sie sich Notizen zu Problemen, erfolgreichen Aktivitäten, informellem Feedback, gewonnenen Erkenntnissen usw. machen, dient Ihnen das als Erinnerungsstütze, wenn Sie den Reflexionsprozess durchführen. Ein weiteres sinnvolles Messinstrument ist ein Einschätzungsbogen für die zuvor festgelegten Fähigkeiten [SmMa2011].

Das Sammeln von Feedback von Menschen aus Ihrer Umgebung wird manchmal auch als 360°-Feedback bezeichnet [LeLu2009].

5.4 Möglichkeiten der persönlichen Weiterentwicklung

Fehlende praktische Erfahrung wird häufig als Grund dafür angeführt, warum eine bestimmte Ermittlungs- oder Konfliktlösungstechnik nicht angewandt wurde. Solch eine Haltung ist zwar mit Blick auf den Projekterfolg verständlich (der Requirements Engineer wendet die Techniken an, die er am besten kennt, um den Erfolg des Projekts sicherzustellen); in Bezug auf die persönliche Weiterentwicklung ist diese Haltung jedoch wenig hilfreich, da der Requirements Engineer unbekannte Techniken so niemals kennenlernt und daher sein Instrumentarium nicht erweitern kann. Im folgenden Abschnitt stellen wir zwei bewährte Verfahren vor, welche die Anwendung neuer Techniken in laufenden Projekten erlauben: die Anwendung in einem Umfeld mit geringem Risiko und parallel zu einer vertrauten Technik.

Eine neue Technik kann in einem Setting mit geringem Risiko angewendet werden, um negative Auswirkungen zu minimieren, falls die Technik nicht das erwartete Ergebnis erbringt. Was ein Umfeld mit geringem Risiko ausmacht, hängt natürlich vom Projektkontext ab. Typische Merkmale eines solchen Umfelds wären etwa:

- Die Anwendung einer Technik auf eine begrenzte Teilgruppe von Stakeholdern. Der Requirements Engineer wendet das Apprenticing beispielsweise nur auf eine geringe Anzahl von Stakeholdern an.
- Die Anwendung einer Technik für einen begrenzten Zeitraum. Der Requirements Engineer plant beispielsweise eine sehr kurze Feldbeobachtung (z. B. eine Stunde) mit bestimmten Stakeholdern.
- Die Anwendung einer Technik in einem wohlgesonnenen Umfeld. Ein Requirements Engineer, der beispielsweise lange in einem Projekt gearbeitet und eine gute Beziehung zu seinen Stakeholdern aufgebaut hat. In einem solchen Umfeld ist die Anwendung nicht vertrauter Techniken in der Regel einfacher.
- Die Anwendung einer Technik auf einen Aspekt des Systems, der als nicht kritisch betrachtet wird. Die Bearbeitung eines Benutzerprofils in einem Onlineshop kann etwa als unkritisch bezeichnet werden. Um die Anforderungen für ein solches Feature zu ermitteln, könnte der Requirements Engineer eine nicht vertraute Technik einsetzen.

Hinweis 5.4.1:

Sie sollten bei Anwendung einer Technik, mit der Sie nicht vertraut sind, offen zu Ihren Stakeholdern sein. Erklären Sie ihnen, dass die angewandte Technik neu für Sie ist, und bitten Sie sie im Nachgang um ihr Feedback. Dieser ehrliche Umgang mit den Stakeholdern verringert für Sie den Druck. Stakeholdern fällt es normalerweise auf, wenn Sie bei einer neuen Technik unsicher sind, und sie sind oftmals bereit, Ihnen Feedback zu geben, damit Sie Ihre Fähigkeiten in diesem Bereich verbessern können.

Weiterhin ist es möglich, eine unbekannte Technik parallel zu einer bekannten Technik einzusetzen, um das Risiko eines Fehlschlags zu reduzieren. Die parallele Anwendung kann auf verschiedene Weise stattfinden:

- **Synchrone Parallelität:** Wenn bei der Anforderungsermittlung mit einer großen Gruppe von Stakeholdern gearbeitet werden muss, kann eine kleinere Gruppe von diesen Stakeholdern eine unbekannte Technik einsetzen. Ein Beispiel: 30 Stakeholder sollen an einem Kreativitätsworkshop teilnehmen. In diesem Fall könnte man eine Teilgruppe von sechs Stakeholdern zur Anwendung einer unbekannteten Technik einladen (z. B. die sechs Denkhüte), wobei die Mehrheit eine vertraute Technik anwendet (z. B. ein Brainstorming).
- **Asynchrone Parallelität:** Wenn die Aufteilung einer Gruppe von Stakeholdern nicht möglich (oder wünschenswert) ist, können stattdessen zwei Techniken sequenziell durchgeführt werden. Ein Beispiel: Fünf Endanwender eines neuen Systems sollen im Rahmen von Interviews an der Anforderungsermittlung teilnehmen. In einer solchen Situation könnte der Requirements Engineer direkt nach jedem Interview eine zusätzliche Feldbeobachtung einplanen.

Hinweis 5.4.2:

Wenn Sie eine unbekannte Technik parallel zu einer vertrauten Technik anwenden, sollten Sie mit Ihren Stakeholdern einen expliziten Vergleich der Ergebnisse aus beiden Techniken einplanen. Auf diese Weise können auch Ihre Stakeholder von den neuen Ergebnissen profitieren.

5.5 Von vorherigen Erfahrungen lernen – lebenslanges Lernen

Ein persönlicher Weiterbildungsprozess, der Lernen aus vorherigen Erfahrungen fördert, beinhaltet die folgenden grundlegenden Bestandteile:

- *Das regelmäßige Messen des eigenen Fähigkeitsprofils:* Ein Bewusstsein für Ihre eigenen Stärken und Schwächen in Bezug auf das Fähigkeitsprofil ist die Basis für eine erfolgreiche Weiterentwicklung. Regelmäßige Analysen (z. B. durch Fragebögen zur Selbsteinschätzung oder Gespräche mit Kunden oder Kollegen) Ihres Fähigkeitsprofils fördern Ihr Bewusstsein für Ihre eigenen Stärken und Schwächen.
- *Schulungsmaßnahmen:* Um Ihr Fähigkeitsprofil zu verbessern, sollten Sie Weiterbildungsmaßnahmen, Schulungen oder Coachings in einer oder mehreren Ermittlungs-/Konfliktlösungstechniken oder den erforderlichen Fähigkeiten (z. B. Mitarbeiterführung oder Motivationstraining) durchführen.
- *Verbesserung in der alltäglichen Arbeit:* Gezielte Schulungsmaßnahmen sind ein erster Schritt, um Ihre Fähigkeiten zu verbessern. Substanzielle Fortschritte kann man jedoch nur durch die Anwendung und Praxis in der täglichen Arbeit erreichen. Die kontinuierliche Verbesserung Ihrer Fähigkeiten muss daher ein integraler Teil Ihrer praktischen Arbeit sein. Gute Fortschritte lassen sich erreichen, wenn die Verbesserung von individuellen Fähigkeiten (z. B. die Verbesserung der Führungskompetenz in Workshops) über einen längeren Zeitraum (mindestens vier Wochen) gefördert wird.
- *Mentoring-Maßnahmen:* Eine Alternative zu einem Intensivtraining ist das Erlernen einer Methode mithilfe eines erfahrenen Mentors. Sie können den Mentor in der Regel

zunächst bei der Anwendung einer Technik unterstützen und durch Beobachten seines Verhaltens dazulernen. Später delegiert Ihr Mentor Verantwortungsbereiche an Sie, beobachtet Ihre Leistung und gibt Ihnen Feedback dazu.

Hinweis 5.5.1:

Entwickeln und kultivieren Sie Ihren persönlichen Weg der kontinuierlichen Verbesserung Ihrer Requirements-Engineering-Fähigkeiten. Die Fakten der meisten Techniken sind einfach zu erlernen; die Weiterentwicklung hin zu exzellenter Kompetenz ist jedoch noch einmal etwas ganz anderes. Es gibt keine vorgegebene „Himmelsleiter“ (oder RE-Exzellenz)! Nutzen Sie die in diesem Abschnitt beschriebenen Elemente als Instrumentarium, um Ihren ganz persönlichen Weg der Verbesserung zu finden.

6 Literaturverzeichnis und weitere Lektüre

- [A4qu2018] Alliance for Qualification: A4Q Design Thinking Foundation Syllabus. <https://isqi.org/en/a4q-design-thinking-foundation-level>, 2018. Zuletzt besucht im Oktober 2021.
- [AlIS1977] Alexander, S. Ishikawa, M. Silverstein: A Pattern Language – Towns – Buildings – Construction. Oxford University Press, New York, 1977.
- [Alexander2005] I. F. Alexander: A Taxonomy of Stakeholders – Human Roles in System Development. International Journal of Technology and Human Interaction, Vol 1, 1, 2005, pages 23–59.
- [BaGL1996] V.R. Basili, S. Green, O. Laitenberger, et al.: The Empirical Investigation of Perspective-Based Reading. Empirical Software Engineering 1: 133. <https://doi.org/10.1007/BF00368702>, 1996. Zuletzt besucht im Februar 2019.
- [BaCC2015] K. Baxter, C. Courage, K. Caine: Understanding Your Users – A Practical Guide to User Research Methods, Morgan Kaufmann; 2. Auflage 2015.
- [BaGr2005] R. Bandler, J. Grinder: Metasprache und Psychotherapie: Die Struktur der Magie I., 12. neu übers. Aufl., 2011.
- [BeHo1998] H. Beyer, K. Holtzblatt: Contextual Design: Defining Customer-Centered Systems. Morgan Kaufmann, 1998.
- [Beveridge 1957] W. I. Beveridge: The Art of Scientific Investigation. The Blackburn Press, Cambridge, 1957.
- [Design Council 2007] Design Council: 11 lessons: managing design in 11 global brands – A study of the design process. <https://www.designcouncil.org.uk/search/?query=Double+Diamond+model>. Zuletzt besucht im März 2024.
- [BiAB2006] Stefan Biffel, Aybuke Aurum, Barry Boehm: Value-Based Software Engineering. Springer-Verlag, Berlin, 2006.
- [Bitkom2017] Bitkom: Role Model Digital Design, <https://www.bitkom.org/Bitkom/Publikationen/Role-Model-Digital-Design.html>. Zuletzt besucht im März 2024.
- [Boehm2006] B. Boehm: A View of 20th and 21st Century Software Engineering. 28th international conference on Software engineering (ICSE '06), Seiten 12–29, 2006.
- [Bourne2015] L. Bourne: Making Projects Work: Effective Stakeholder and Communication Management. CRC Press, 2015.
- [Brls2005] J. Brown, D. Isaacs: Das World Café: Kreative Zukunftsgestaltung in Organisationen und Gesellschaft, Carl-Auer-Verlag, Heidelberg, 2007.
- [Brown2009] T. Brown: Change by Design – How Design Thinking Transforms Organizations and Inspires Innovation. Verlag Franz Vahlen, München, 2016.

- [Buxton2007] B. Buxton: Sketching User Experiences – Getting the design right and the right design. Morgan Kaufmann, San Francisco, 2007.
- [BuBu2005] T. Buzan, B. Buzan: Das Mind-Map-Buch – Die beste Methode zur Steigerung Ihres geistigen Potenzials, Morgan Kaufmann; 5. aktualisierte Auflage, 2005.
- [BuHe2019] S. Bühne, A. Herrmann: Handbuch Anforderungsmanagement nach dem IREB-Standard (Version 1.1.0), IREB e.V., Karlsruhe, 2019.
<https://www.ireb.org/en/downloads/#cpre-advanced-level-requirements-management-handbook>. Zuletzt besucht im Mai 2022
- [Cambridge2017] Cambridge Dictionary
<https://dictionary.cambridge.org/dictionary/english/abstraction>. Zuletzt besucht im Februar 2019.
- [Carrol2003] JM. Carrol: Making Use. The MIT Press, 2003.
- [Chernak2012] Y. Chernak: Requirements reuse: the state of the practice. In: Proceedings of the 2012 IEEE international conference on software science, technology & engineering (SwSTE), Seiten 46–53
- [Cockburn2001] A. Cockburn: Use Cases effektiv erstellen. mitp, Heidelberg, 2008.
- [CRCN2014] A. Cooper, R. Reimann, D. Cronin, C. Noessel: About Face: The Essentials of Interaction Desig. 4. Auflage, John Wiley & Sons, Inc, Hoboken, 2014.
- [CoSh2007] T. Colburn, G. Shute: Abstraction in Computer Science, Minds & Machines, Vol. 17, Seiten 169–184, 2007.
- [Cooper2004] A. Cooper: The Inmates are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity. Que, Indianapolis, 2004.
- [Couger1996] J. D. Couger: Creativity and Innovation in Information Systems Organizations. Byod & Fraser, 1996.
- [CrOB2006] O. Creighton, M. Ott, B. Bruegge: Software Cinema: Video-based Requirements Engineering. 14th IEEE International Requirements Engineering Conference (RE'06), 2006.
- [CHQW2016] T. Cziharz, P. Hruschka, S. Queins, T. Weyer: Handbook of Requirements Modeling According to the IREB Standard (version 1.3), IREB e.V., Karlsruhe, 2016. <https://www.ireb.org/en/downloads/#handbook-cpre-advanced-level-requirements-modeling>. Zuletzt besucht im März 2024.
- [DeBono2006] E. De Bono: De Bonos neue Denkschule: Kreativer Denken, Effektiver Arbeiten, Mehr Erreichen. mvg Verlag, München, 2010.
- [DeDe2011] K. Dewalt, B. Dewalt: Participant observation – A guide for fieldworkers. 2nd Edition. AltaMitra Press, Plymouth, UK, 2011.
- [DDP2021] IREB e.V.: Lehrplan DDP Foundation Level, Version 1.0.1.
<https://www.digitaldesign.org/syllabus>. Zuletzt besucht im März 2024.

- [EIOs2017] R. Elamin, R. Osman: Towards Requirements Reuse by Implementing Traceability in Agile Development, in: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC) Italy, <http://ieeexplore.ieee.org/abstract/document/8029969/?part=1>, 2017. Zuletzt besucht im März 2024.
- [FiUP2012] R. Fisher, W. Ury, B. Patton: Das Harvard-Konzept: Die unschlagbare Methode für beste Verhandlungsergebnisse, Deutsche Verlags-Anstalt, München, 2018.
- [GHJV1994] E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns. Elements of Reusable Object-Oriented Software. mitp Verlag, Frechen, 2015.
- [Glasl1999] F. Glasl: Confronting Conflict – A first-aid kit for handling conflict. Hawthorn Press, Gloucestershire, 1999.
- [Glasl2004] F. Glasl: Konfliktmanagement: Ein Handbuch für Führungskräfte, Beraterinnen und Berater. 11. Auflage, Freies Geistesleben, Stuttgart, 2004.
- [GleA2020] M. Glinz et al.: Handbook for the CPRE Foundation Level nach IREB Standard (Version 1.0.0). IREB e.V., Karlsruhe, 2020. <https://www.ireb.org/de/downloads/#cpre-foundation-level-handbook>. Zuletzt besucht im März 2024.
- [Glin2020] M. Glinz: A Glossary of Requirements Engineering Terminology (Version 2.0.0). IREB e.V., Karlsruhe, 2020. <https://www.ireb.org/de/downloads/#cpre-glossary>. Zuletzt besucht im März 2024.
- [GoBe2015] L. Goldin, D. M. Berry: Reuse of requirements reduced time to market at one industrial shop: a case study, Requirements Engineering Journal 2013. In: Requirements Engineering–Volume 20, Issue 1, March 2015.
- [Goodwin2009] K. Goodwin: Designing for the Digital Age. Wiley, New York, 2009.
- [Gothelf2013] J. Gothelf: Lean UX – Applying Lean Principles to Improve User Experience. mitp Verlag, Frechen, 2015.
- [Gottesdiener2002] E. Gottesdiener: Requirements by Collaboration: Workshops for Defining Needs, Addison-Wesley Professional, 2002.
- [GoWo2005] T. Gorschek, C. Wohlin: Requirements Abstraction Model, Requirements Engineering Journal 2005. In Requirements Engineering Volume 11, Issue 1, Seiten 79–101, Dezember 2005.
- [Groen et al.2017] E. C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini, M. Stade: The Crowd in Requirements Engineering – The Landscape and Challenges, IEEE Software, vol. 34, no. 2, Seiten 44–52, 2017.
- [GrKo2016] E. C. Groen, M. Koch: How Requirements Engineering can benefit from crowds – Driving innovation with crowd-based techniques. Requirements Engineering Magazin Vol. 2. <https://re-magazine.ireb.org/issues/2016-2-take-the-broader->

[view/how-requirements-engineering-can-benefit-from-crowds](#). 2016. Zuletzt besucht im März 2024.

- [HaPy2012] R. Hartson, P.S. Pyla: The UX Book: Process and Guidelines for ensuring a Quality User Experience. Morgan Kaufmann, San Francisco, 2012.
- [IREB2020] IREB Certified Professional for Requirements Engineering Foundation Level Syllabus (version 3.0.1). IREB e.V., Karlsruhe, 2020.
<https://www.ireb.org/de/downloads/#cpre-foundation-level-syllabus-3-0>.
Zuletzt besucht im März 2024.
- [IsNe2013] A. Ishizaka, P. Nemery: Multi-criteria decision analysis. Methods and software. John Wiley & Sons, Ltd, Chichester, 2013.
- [ISO25010] ISO/IEC 25010:2011: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. International Organization for Standardization, Genf, 2011.
- [ISO29148] ISO/IEC/IEEE29148: Systems and software engineering — Life cycle processes — Requirements Engineering. International Organization for Standardization, Genf, 2011.
- [ISO9241.11] ISO9241: Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability. International Organization for Standardization, Genf, 1998.
- [ISO9241.210] ISO9241: Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems. International Organization for Standardization, Genf, 2010 (ersetzt ISO 14407-210: 1999).
- [JoDö2004] I. John, J. Dörr: Requirements Engineering, basierend auf existierenden Systemen, in: G. Böckle, P. Knauber, K Pohl, K Schmid. „Software-Produktlinien - Methoden, Einführung und Praxis“, dpunkt.Verlag, Heidelberg 2004, Seiten 153-163.
- [KaBP2002] C. Kaner, J. Bach, B. Pettichord: Lessons Learned in Software Testing – A Context-Driven Approach. Wiley, New York, 2002.
- [Katie2017] Katie: Pril Gets Pranked. Social Media for Business Performance, <https://smbp.uwaterloo.ca/2017/05/pril-gets-pranked/>, 2017. Zuletzt besucht im Februar 2019.
- [Klaus2007] P. Klaus: The Hard Truth About Soft Skills – Workplace Lessons Smart People Wish They'd Learned Sooner. HarperCollins Publishers, New York, 2007.
- [KnZK2016] J. Knapp, J. Zeratsky, B. Kowitz, B.: Sprint: Wie man in nur fünf Tagen neue Ideen testet und Probleme löst. Simon & Schuster. 2016.
- [Koelsch2016] G. Koelsch: Requirements Writing for System Engineering. Apress, 2016.
- [Koes1964] A. Koestler: The Act of Creation. Scherz Verlag, 1966.

- [Kumar2013] V. Kumar: 101 Design Methods – A Structured Approach for Driving Innovation in Your Organization. Wiley, 2013.
- [Kvale2008] S. Kvale: Doing Interviews. SAGE, 2008.
- [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 Magazin Vol. 1. <http://re-magazine.ireb.org/issues/2014-1-learning-to-fly/what-does-it-mean-to-say-requirement>. Zuletzt besucht im März 2024.
- [LeLL2018] M. Lewrick, P. Link, L. Leifer: The Design Thinking Playbook: Mindful Digital Transformation of Teams, Products, Services, Businesses and Ecosystems. Vahlen, München, 2. überarbeitete Auflage 2018.
- [LeLu2009] R. Lepsinger, A.D. Lucia: The Art and Science of 360 Degree Feedback. 2. Auflage, Wiley, San Francisco, 2009.
- [LiHB2003] W. Lidwell, K. Holden, J. Butler: Design: Die 100 Prinzipien für erfolgreiche Gestaltung. Stiebner, Grünwald, 2004.
- [LiFi2012] S. L. Lim, A. Finkelstein: StakeRare: Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation, in IEEE Transactions on Software Engineering, vol. 38, no. 3, Seiten 707–735, 2012.
- [LiOg2011] J. Liedtka, T. Liedtka, T. Ogilvie. Designing for Growth: A Design Thinking Tool Kit For Managers. Columbia University Press, 2011.
- [LoSL2017] H. van Loenhoud, P. Steiger, K. Lauenroth: The goal is to solve the problem – Some thoughts on problems and goals in the context of Requirements Engineering. RE magazine 2017–02. <https://re-magazine.ireb.org/articles/the-goal-is-to-solve-the-problem>, 2017. Zuletzt besucht im Februar 2019.
- [Maiden et al.2010] N. Maiden, S. Jones, I. Karlsen, R. Neill, K. Zachos, A. Milne: Requirements Engineering as Creative Problem Solving: A Research Agenda for Idea Finding. 18th IEEE International Requirements Engineering Confereneec (RE), 2010.
- [MaGi2001] N. Maiden, A. Gizikis: Where Do Requirements Come From? IEEE Software Vol. 18, No. 5, 2001, S. 10–12.
- [McCo2006] S. McConnell: Software Estimation – Demystifying the Black Art, Microsoft Press, 2006.
- [McEl2017] K. McElroy: Prototyping for Designers: Developing the Best Digital and Physical Products. O’Reilly, 2017.
- [GrBa2017] J. McGrath, B. Bates: Der 5–Minuten–Manager: Die wichtigsten Management–Theorien auf den Punkt. Börsenmedien, Kulmbach, 2. Auflage 2014.
- [Mill2009] R.E. Miller: The Quest for Software Requirements. MavenMark Books, 2009.
- [Moor2014] C. W. Moore: The Mediation Process – Practical Strategies for Resolving Conflicts. 4. Auflage, John Wiley & Sons, Hoboken, 2014.

- [Osbo1948] A. F. Osborn: Your creative power: How to use imagination. C. Scribner's Sons, 1948. (Zugriff auf digitale Version: Read Books Ltd. (epub eBook), April 2013)
- [Owen2008] H. Owen: Open Space Technology: A User's Guide. Schäffer-Poeschel, Stuttgart, 2. aktualisierte und erweiterte Auflage, 2011.
- [Dsch2012] An Introduction to Design Thinking PROCESS GUIDE. Hasso Plattner Institute of Design. <https://hpi.de/school-of-design-thinking/design-thinking.html>. Zuletzt besucht im März 2024.
- [Pohl2010] K. Pohl: Requirements Engineering: Grundlagen, Prinzipien, Techniken. Springer, 2010.
- [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.
- [Rein1997] D. Reinertsen: Managing the Design Factory: A Product Developers Tool Kit. Hanser, München, Wien, 1998.
- [RiFl2014] M. Richter, M. Flückiger: Usability und UX kompakt: Produkte für Menschen. Springer, Berlin Heidelberg, 2014.
- [Robertson2001] S. Ian Robertson: Problem Solving. Psychology Press, 2001.
- [Robles2012] M.M. Robles: Executive Perceptions of the Top 10 Soft Skills Needed in Today's Workplace. Business Communication Quarterly 75(4) S. 453-465, 2012.
- [RoRo2013] S. Robertson, J. Robertson: Mastering the Requirements Process: Getting Requirements Right. Dritte Auflage, Pearson Education, London, 2013.
- [Rosenberg2015] M. B. Rosenberg: Gewaltfreie Kommunikation: Eine Sprache des Lebens. 3rdrev. ed., Puddle Dancer Press (US), Encinitas, 2015.
- [RuCh2008] J. Rubin, D. Chisnell: Handbook of Usability Testing- How to Plan, Design, and Conduct Effective Tests. Wiley; Idianapolis, 2008.
- [Rupp et al.2014] 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.)
- [Schramm1971] W.L. Schramm: How communication works. in W.L. Schramm, Ed., (1971). The Process and Effects of Mass Communication, Überarb. Auflage, University of Illinois Press, 1971.
- [Schulz von Thun 1981] F. Schulz von Thun: Miteinander reden 1 - Störungen und Klärungen. Psychologie der zwischenmenschlichen Kommunikation. Rowohlt, Reinbek, 1981.

- [Shackel1991] 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. Oldenbourg Verlag, München, Wien, 1976.
- [SiOp2005] G. Sindre, A. L. Opdahl: Eliciting security requirements with misuse cases. Requirements Engineering Journal, Vol. 10, No. 1, 2005.
- [SmMa2011] S. Smith, R. Mazin: The HR Answer Book: An Indispensable Guide for Managers and Human Resources Professionals. 2. Auflage, AMACOM, New York, 2011.
- [Snijders et al.2015] R. Snijders, F. Dalpiaz, S. Brinkkemper, M. Hosseini, R. Ali, A. Özüm: REfine: A Gamified Platform for Participatory Requirements Engineering, Proc. 1st Int'l Workshop Crowd–Based Requirements Eng. (CrowdRE 15), S. 1–6, 2015.
- [Snyder2003] C. Snyder: Paper Prototyping – The Fast and Easy Way to Design and Refine User Interfaces. Morgan Kaufmann, 2003.
- [TiSi2017] S. Tiwari, S. Singh Rathore: A Methodology for the Selection of Requirement Elicitation Techniques. In: arXiv e–prints, 2017.
<https://arxiv.org/abs/1709.08481>. Zuletzt besucht im März 2024.
- [UXQB2017] UXQB Advanced Level – Usability Testing and Evaluation (CPUX–UT), Version 1.07, Februar 2017.
- [Walten et al.2015] D.D. Walten, G.J. Roedler, K.J. Forsberg, R.D. Hamelin, T.N. Shortell: Systems Engineering Handbook – A Guide for System Life Cycle Process and Activities. GfSE, Gesellschaft für Systems Engineering e.V., München, 2017.
- [Warfel2009] T. Warfel: Prototyping – A Practitioner's Guide. Rosenfeld Media, 2009.
- [Wiki2017] Wikipedia Principle of abstraction:
https://en.wikipedia.org/wiki/Principle_of_abstraction. Zuletzt besucht im März 2024.
- [Withall2007] S. Withall: Software Requirements Patterns. Microsoft Press, 2007.
- [YoAs2015] M. Yousuf, M. Asger: Comparison of Various Requirements Elicitation Techniques. In: International Journal of Computer Applications, Vol. 116, No. 4, 2015.