



Chris Rupp

Requirements Engineering

Ein Überblick

3., aktualisierte und erweiterte Auflage

SOPHIST 

dpunkt.verlag

Chris Rupp
SOPHIST GmbH
Vordere Cramergasse 13
90478 Nürnberg
chris.rupp@sophist.de
www.sophist.de

3., aktualisierte und erweiterte Auflage 2012
Copy Editing: Ursula Zimpfer, Herrenberg
Herstellung: Birgit Bäuerlein
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck: Wörmann PRODUCTION CONSULT, Heidelberg

Artikel-Nr. 077.95731

Copyright © 2012 dpunkt.verlag GmbH
Ringstraße 19 b
69115 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die in der Broschüre verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in dieser Broschüre wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieser Broschüre stehen.

5 4 3 2 1 0

Chris Rupp

Requirements Engineering

Ein Überblick

3., aktualisierte und erweiterte Auflage



dpunkt.verlag

Liebe Leser,

diese Broschüre soll Ihnen einen kleinen Einstieg in die vielschichtige Disziplin des Requirements Engineering bieten. Sie basiert auf »Basiswissen Requirements Engineering«, dem offiziellen Lehrbuch für die Zertifizierung zum *Certified Professional for Requirements Engineering (CPRE)* – *Foundation Level*, verfasst von Klaus Pohl und Chris Rupp, den Autoren der auflagenstärksten Bücher zum Thema Requirements Engineering. Die Entscheidung der Autoren, jenes Buch gemeinsam zu verfassen, kam nicht von ungefähr. Es sollten langjährige Praxiserfahrungen mit Lehr- und Forschungserkenntnissen zum Thema Requirements Engineering zusammengeführt werden.

Die immer stärker an Bedeutung gewinnende Disziplin Requirements Engineering befasst sich mit Anforderungen in der Systementwicklung: mit ihrer Ermittlung, ihrer Dokumentation, ihrer Validierung und ihrer Verwaltung. Die vorliegende Broschüre möchte Ihnen vermitteln, was genau Requirements Engineering ist, und Ihnen seine verschiedenen Bereiche vorstellen. Sie erfahren Grundsätzliches über das Persönlichkeitsprofil und Berufsbild des Requirements Engineer und lernen einige Aspekte seiner täglichen Berufspraxis kennen.

Ich hoffe, dass Sie gerne mit mir und dieser Broschüre auf die Reise in die spannende Welt des Requirements Engineering gehen. Wenn Sie nach der Lektüre Interesse haben, das Thema zu vertiefen, finden Sie am Ende der Broschüre weiterführende Informationen und Literatur.

Chris Rupp
Nürnberg, im Dezember 2011

Requirements Engineering – was ist das?

Die Bedeutung des Requirements Engineering (RE) für die erfolgreiche, den Kundenwünschen entsprechende Entwicklung von Systemen ist mittlerweile kaum mehr zu übersehen. In der Praxis ist es üblich, einen entsprechenden Aufwand für das Requirements Engineering einzuplanen. Immer häufiger findet man zudem die Erkenntnis, dass der Requirements Engineer eine eigenständige Rolle mit anspruchsvollen Tätigkeiten ist.

Glaubt man den Zahlen im Chaos Report 2009 der Standish Group, so hat sich in den fünfzehn Jahren von 1994 bis 2009 bei der erfolgreichen Abwicklung von Softwareprojekten einiges zum Besseren gewendet.

Sind im Jahre 1994 noch gut 30 Prozent der untersuchten Softwareprojekte gescheitert, so waren es 2009 nur noch 24 Prozent. Die Anzahl der gefährdeten Projekte, die mit starken Zeit- oder Budgetüberziehungen und/oder nicht zur Zufriedenheit der Kunden abgeschlossen werden konnten, verringerte sich von 53 Prozent auf 44 Prozent.

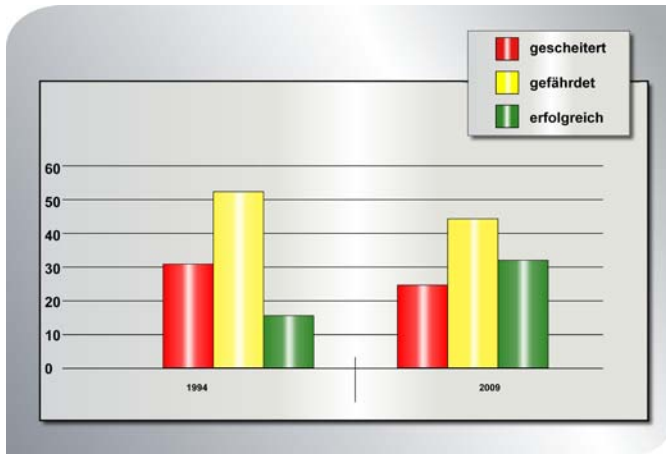


Abb. 1 Vergleich der Zahlen des Chaos Report von 1994 und 2009

Jim Johnson, Vorsitzender der Standish Group, nennt als einen von drei Gründen für die positive Entwicklung der Zahlen seit 1994 die Tatsache, dass Anforderungen besser kommuniziert würden als noch vor zehn Jah-

ren. Interessant sind diese Zahlen, da der Umgang mit Anforderungen eines Systems eine signifikante Ursache für Projektfehlschläge bzw. für Zeit- und Budgetüberschreitungen darstellt: Die Mehrheit der schweren Fehler passieren laut einer Studie von Barry Boehm [Boehm 1981] nicht in den späteren Phasen der Systementwicklung, sondern bereits in der Analysephase.

Definition »Anforderung«

Eine Anforderung ist:

- (1) Eine Bedingung oder Fähigkeit, die von einem Benutzer (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.
- (2) Eine Bedingung oder Fähigkeit, die ein System oder Teilsystem erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.
- (3) Eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft gemäß (1) oder (2).

Übersetzt aus [IEEE]

Je später ein Fehler in den Anforderungen im Verlauf des Entwicklungsprojekts behoben wird, desto höher sind die damit verbundenen Kosten. So wird beispielsweise für die Beseitigung eines Anforderungsfehlers, der erst beim Programmieren entdeckt wird, ein um zirka den Faktor 20 höherer Aufwand angenommen – für die Fehlerbeseitigung in der Abnahmephase geht man von dem Faktor 100 aus.

Ziel des Requirements Engineering ist es, möglichst vollständige Kundenanforderungen in guter Qualität zu dokumentieren und dabei Fehler möglichst frühzeitig zu erkennen und zu beheben.

Der Stakeholder (Projektbetroffener) ist einer der zentralen Begriffe im Requirements Engineering. Stakeholder sind neben weiteren Faktoren eine wichtige Quelle für Anforderungen. Das Übersehen von Stakeholdern hat häufig zur Konsequenz, dass Anforderungen an das System lückenhaft sind oder gänzlich fehlen. Stakeholder sind also alle diejenigen Personen oder Organisationen, die Anforderungen in irgendeiner Weise beeinflussen. Das können natürliche Personen sein, die das System später nutzen werden

(z. B. der Nutzer oder der Administrator), oder solche, die es nicht nutzen werden oder sollen (z. B. das Management oder ein Hacker, vor dem man das System schützen muss), aber auch juristische Personen, Institutionen usw., da diese letztlich durch natürliche Personen vertreten werden, die die Anforderungen des betrachteten Systems beeinflussen bzw. definieren können.

Definition »Stakeholder«

Ein Stakeholder eines Systems ist eine Person oder Organisation, die (direkt oder indirekt) Einfluss auf die Anforderungen des betrachteten Systems hat.

Requirements Engineering ist kaum noch wegzudenken, wenn es darum geht, für den Kunden zufriedenstellende Systeme zu entwickeln und dabei Budget- und Zeitpläne einzuhalten. In der heutigen Zeit kann sich kein Unternehmen mehr ein halbherziges Anforderungsmanagement bzw. eine fahrlässige Anforderungserhebung leisten. Moderne Systeme sollen schneller, besser, innovativer sein. Fehler müssen so früh wie möglich entdeckt und behoben werden.

Definition »Requirements Engineering«

Das Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit den folgenden Zielen:

- (1) Die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen, die Anforderungen konform zu vorgegebenen Standards zu dokumentieren und die Anforderungen systematisch zu managen.
- (2) Die Wünsche und Bedürfnisse der Stakeholder zu verstehen, zu dokumentieren sowie die Anforderungen zu spezifizieren und zu managen, um das Risiko zu minimieren, dass das System nicht den Wünschen und Bedürfnissen der Stakeholder entspricht.

Mittlerweile haben viele Unternehmen einen etablierten Requirements-Engineering-Prozess. Unter anderem hat die Möglichkeit der Zertifizierung zum *Certified Professional for Requirements Engineering* (CPRE) dazu beigetragen, dass der Requirements Engineer immer mehr als eine eigene, spezialisierte Rolle gesehen und mit eigenen Ressourcen für seine anspruchsvolle Arbeit ausgestattet wird.

Der Requirements Engineer und die Zertifizierung

Berufsbild und Persönlichkeitsprofil des Requirements Engineer

Der Requirements Engineer als Projektrolle steht häufig im Mittelpunkt des Geschehens. Er pflegt in der Regel als Einziger direkten Kontakt zu allen Stakeholdern und hat die Chance und Verantwortung, sich ausreichend in das Fachgebiet der Stakeholder einzuarbeiten sowie die Sprache in den verschiedenen Fachgebieten zu erlernen und zu verstehen. Er ist derjenige, der die Bedürfnisse hinter den Aussagen der Stakeholder erkennt und so aufbereiten muss, dass fachfremde Architekten und Entwickler sie verstehen und umsetzen können.

Man kann sich den Requirements Engineer also als eine Art Dolmetscher vorstellen, der sowohl das Fachgebiet und dessen Sprache ausreichend kennt als auch über genug IT-Know-how verfügt, um sich der Probleme der Entwickler bewusst zu sein und mit diesen gleichberechtigt kommunizieren zu können. Um allen Aufgaben gerecht werden zu können, benötigt der Requirements Engineer weit mehr als Methodenwissen. Viele der erforderlichen Fähigkeiten setzen entsprechende praktische Erfahrungen voraus.

Analytisches Denken

Der Requirements Engineer muss fähig sein, sich in ihm unbekannte Fachgebiete und Sachverhalte schnell einzuarbeiten und dabei komplizierte Probleme und Zusammenhänge zu verstehen und zu analysieren.

Da Stakeholder oft in konkreten Beispielen und (suboptimalen) Lösungen über das eigentliche Problem und die zugehörigen Anforderungen sprechen, muss der Requirements Engineer in der Lage sein, konkrete Aussagen der Stakeholder zu abstrahieren.

Empathie

Der Requirements Engineer hat die schwierige Aufgabe, zu erkennen, was ein Stakeholder tatsächlich benötigt. Hierfür ist ein ausgeprägtes Einfühlungsvermögen eine der zentralen Voraussetzungen.

Kommunikationsfähigkeit

Um die Anforderungen der Stakeholder zu erheben, richtig zu interpretieren und zu kommunizieren, muss der Requirements Engineer über hohe kommunikative Fähigkeiten verfügen. Er muss zuhören können, zur rechten Zeit die richtigen Fragen stellen, bemerken, wenn Aussagen nicht den gewünschten Informationsgehalt haben, und rechtzeitig erforderliche Rückfragen stellen.

Konfliktlösungsfähigkeit

Durch unterschiedliche Meinungen der Stakeholder kommt es im Requirements Engineering häufig zu Konflikten. Der Requirements Engineer muss Konflikte erkennen, zwischen den Parteien vermitteln und schließlich durch den Einsatz geeigneter Techniken den Konflikt auflösen.

Moderationsfähigkeit

Der Requirements Engineer muss zwischen unterschiedlichen Meinungen vermitteln und Diskussionen leiten können. Dies gilt sowohl für Einzelbesprechungen als auch in Gruppendiskussionen oder in Workshops.

Selbstbewusstsein

Da der Requirements Engineer häufig im Mittelpunkt steht und dabei gelegentlich auch Kritik ausgesetzt ist, benötigt er ein selbstbewusstes Auftreten und die Fähigkeit, sich auch durch hartnäckige Ablehnungen nicht aus dem Konzept bringen zu lassen. Er sollte Kritik niemals persönlich nehmen.

Überzeugungsfähigkeit

Der Requirements Engineer ist unter anderem eine Art Anwalt für die Anforderungen seiner Stakeholder. Er muss fähig sein, diese nach außen und in Besprechungen und Präsentationen zu vertreten.

Das International Requirements Engineering Board und die Zertifizierung zum Certified Professional for Requirements Engineering

Im Jahr 2007 wurde das International Requirements Engineering Board (IREB e.V.) gegründet. Es setzt sich aus unabhängigen, weltweit anerkannten Experten aus den Bereichen Industrie, Beratung, Forschung und Lehre zusammen.

Das International Requirements Engineering Board hat sich der Aufgabe verschrieben, das Requirements Engineering zu professionalisieren. Professionalität bedeutet hierbei, anerkannte Methoden und Verfahren einzusetzen, eine standardisierte Begriffsbildung und Begriffsverwendung zu fördern, aber auch den Nachweis zu führen, dass Methoden, Verfahren und Begriffe von verantwortlichen Personen korrekt eingesetzt werden.

Die Mitglieder des Boards erarbeiten gemeinsam eine Reihe von Lehrplänen unterschiedlicher Erfahrungsstufen (Foundation Level, Advanced Level und Expert Level) für den Bereich Requirements Engineering und entwickeln darauf aufbauende Zertifizierungen, die eine spezialisierte Ausbildung zum *Certified Professional for Requirements Engineering (CPRE)* ermöglichen und am Ende bescheinigen. Ziel ist es, eine qualitätsgesicherte Standardisierung der Aus- und Weiterbildung im Requirements Engineering und damit letztlich eine breite Verbesserung der täglichen Requirements-Engineering-Praxis zu erreichen.

Im Detail sieht das Zertifizierungsmodell des IREB eine Unterteilung der CPRE-Ausbildung in drei aufeinander aufbauende Zertifizierungsstufen mit den entsprechenden Lehrplänen vor. Die Erreichung einer Stufe ist dabei Voraussetzung für die Teilnahme einer höheren Zertifizierung. Der Lehrplan für den Foundation Level umfasst das Grundlagenwissen zum Requirements Engineering auf den Gebieten Ermittlung, Dokumentati-

on, Prüfung und Verwaltung von Anforderungen. Die Advanced-Level-Zertifizierung bildet die zweite Stufe des IREB-Modells und ist modular aufgebaut. Hier können anhand bestimmter wählbarer Schwerpunkte die passenden Module zur Vertiefung der im Foundation Level vermittelten Inhalte belegt werden. Nach Bestehen der Prüfung erhält man das entsprechende Advanced-Level-Zertifikat des jeweiligen Moduls. Teile des Advanced-Level-Zertifikats sowie der Expert Level als höchste Zertifizierungsstufe befinden sich derzeit noch im Aufbau. Eine Übersicht über das vollständige IREB-Ausbildungsmodell gibt Abbildung 2.

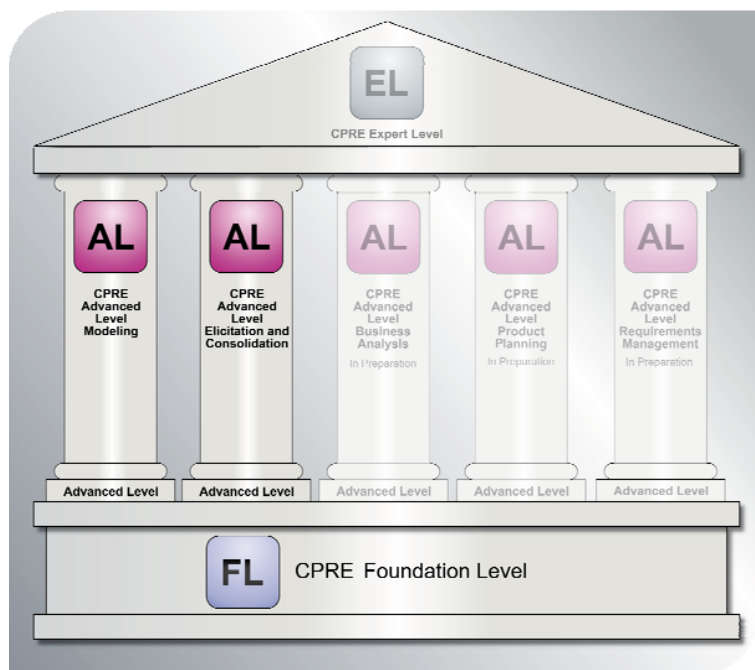


Abb. 2 Das vollständige Zertifizierungsmodell des IREB

Im Jahre 2007 begann das IREB sehr erfolgreich seine Arbeit in Deutschland, Österreich und der Schweiz. Seit der Bereitstellung der CPRE-Lehrpläne auf Englisch, Französisch, Spanisch und Portugiesisch (für Brasilien) kommen stetig neue Länder hinzu, die ebenfalls CPRE-Zertifizierungen anbieten, z. B. die USA, Israel, Indien, Malaysia, Australien, Brasilien, Polen und viele mehr. Am Zertifizierungsprozess sind vier Hauptakteure beteiligt: das International Requirements Engineering Board, die anerkannten Schulungsanbieter (Trainingsprovider), die Zertifizierungsstellen in den einzelnen Ländern und natürlich die Kursteilnehmer bzw. die zu prüfenden Personen. Das IREB erarbeitet die Lehrpläne, erstellt die zugehörigen Prüfungsfragen, definiert und regelt das Prüfungsverfahren und beauftragt Zertifizierungsstellen mit der Prüfungsabnahme. Um die operativen Aufgaben des IREB e.V. wie die Koordination von Arbeitsgruppen, die Kommunikation und Vertragsgestaltung mit Zertifizierungsunternehmen und Trainings Providern sowie das Marketing kümmert sich die im April 2011 gegründete IREB GmbH.

In den einzelnen Ländern führen vom IREB beauftragte Zertifizierungsstellen die Prüfungen für das Zertifikat durch.

Durch diese Lehrpläne gibt das IREB genau den Umfang, den Inhalt und die Zeit für die Erreichung der Lernziele sowie die Themen der praktischen Übungen vor. Des Weiteren können auf den Internetseiten des IREB (<http://certified-re.de>) auch das vollständige aktuelle Glossar und nähere Informationen zu den Prüfungsmodalitäten nachgeschlagen werden. Derzeit haben bereits über 7500 Teilnehmer weltweit CPRE-Zertifizierungsprüfungen abgelegt. Das begleitende Lehrbuch für den CPRE-Foundation Level existiert in deutscher und englischer Sprache und bald auch auf Portugiesisch (Brasilien).

Die Kerndisziplinen des Requirements Engineering

Um ein Entwicklungsprojekt zum Erfolg führen zu können, muss zunächst bekannt sein, was die Anforderungen an das System sind, und diese müssen geeignet dokumentiert sein.

Dem Requirements Engineering im Entwicklungsprozess kommt die Aufgabe zu, die Anforderungen der Stakeholder zu ermitteln, zweckmäßig zu dokumentieren, zu überprüfen und abzustimmen sowie die dokumentierten Anforderungen über den gesamten Lebenszyklus des Systems hinweg zu verwalten.

Die vier damit einhergehenden Haupttätigkeiten sind das Ermitteln, Dokumentieren, Prüfen und Abstimmen sowie das Verwalten von Anforderungen.



Abb. 3 Die Kerndisziplinen des Requirements Engineering

Anforderungen ermitteln

Die Disziplin »Anforderungen ermitteln« umfasst viele Tätigkeiten. Zuerst müssen Ziele festgelegt und das System von seiner Umwelt klar abgegrenzt werden. Nur so wissen alle Projektbeteiligten, was Teil des Systems und daher Gegenstand der detaillierten Systemanalyse sein wird und zu welchen Nachbarsystemen und menschlichen Nutzern Schnittstellen spezifiziert werden müssen.

Meist sogar noch vor Zieldefinition und Kontextabgrenzung muss ermittelt werden, welche Anforderungsquellen in welchem Rahmen zur Verfügung stehen.

Eine wichtige Anforderungsquelle sind Stakeholder. Dies sind nicht nur die zukünftigen Nutzer des Systems, sondern auch z. B. die Marketingabteilung, die Absatzziele im Kopf hat und den Kunden als Stakeholder intern vertritt, wenn der Kunde ein momentan noch anonymer großer Markt ist. Das Management als Entscheider und Sponsor ist ebenso Stakeholder wie

die Personen, die später das System warten und administrieren sollen, das Schulungspersonal, der Betriebsrat, der Ergonomieexperte und viele mehr.

Aber auch Dokumente sind wichtige Anforderungsquellen, von denen viele Anforderungen abgeleitet werden können. Beispiele hierfür sind allgemeingültige Dokumente wie z. B. Normen/Standards oder Gesetzestexte sowie branchen-/organisationsspezifische Dokumente, z. B. Anforderungsdokument, Benutzungshandbuch oder Fehlerberichte des Altsystems.

Auch Systeme in Betrieb kommen als Anforderungsquelle infrage. Dies können sowohl Alt- bzw. Vorgängersysteme als auch Konkurrenzsysteme sein.

Ermittlungstechniken

Um das Wissen und die Anforderungen der Stakeholder zu ermitteln, kann man sich unterschiedlicher Ermittlungstechniken bedienen. Denn so unterschiedlich Projektrahmenbedingungen, Systeme sowie die Persönlichkeiten der Stakeholder sind, so unterschiedlich sind die Techniken, die bei der Ermittlung von Anforderungen Erfolg versprechen. Daher ist eine bewusste und situationsbezogene Entscheidung für eine bestimmte Kombination von Ermittlungstechniken nötig. Ermittlungstechniken werden nach Befragungstechniken, Kreativitätstechniken, dokumentenzentrierten Techniken und Beobachtungstechniken unterschieden.

Für die Anforderungsermittlung ist das Wissen, welche Bedeutung die Anforderungen für die Zufriedenheit der Stakeholder haben, sehr hilfreich. Diese Zufriedenheit wird mit den jeweiligen Merkmalen eines Produkts, von denen sie abhängen, nach dem Modell von Kano in drei Kategorien eingeteilt (siehe auch Abb. 4):

Basisfaktoren sind selbstverständlich vorausgesetzte Systemmerkmale (unterbewusstes Wissen).

Leistungsfaktoren sind die explizit geforderten Systemmerkmale (bewusstes Wissen).

Begeisterungsfaktoren sind Systemmerkmale, die der Stakeholder nicht kennt und erst während der Benutzung als angenehme und nützliche Überraschungen entdeckt (unbewusstes Wissen).

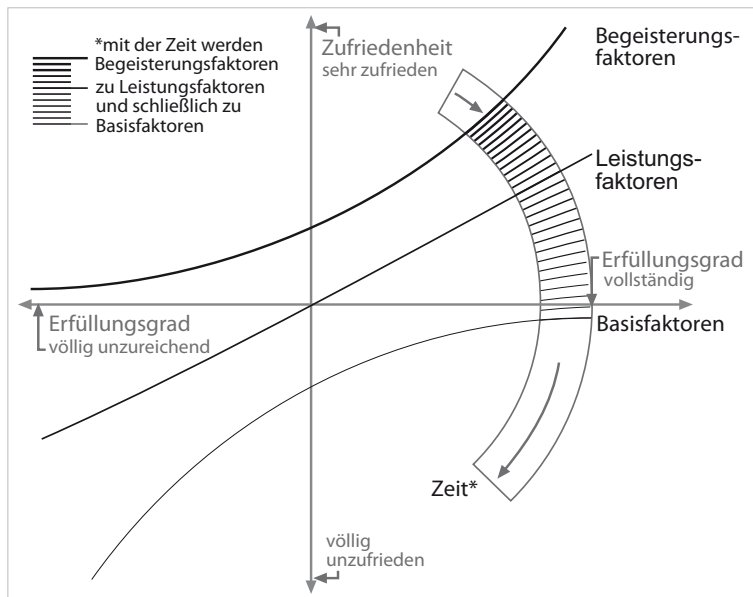


Abb. 4 Das Kano-Modell

Während bei der Anforderungsermittlung Befragungstechniken vor allem zum Abfragen von explizitem Wissen geeignet sind, also der Leistungsfaktoren nach Kano, können Beobachtungstechniken und einige dokumentenzentrierte Techniken besonders gut für die Erhebung von implizitem Wissen, den sogenannten Basisfaktoren, eingesetzt werden. Um an die unbewussten Wünsche der Stakeholder zu gelangen, die häufig die Begeisterungsfaktoren im Kano-Modell ausmachen, müssen Wünsche erraten oder durch Kreativitätstechniken an die Oberfläche geholt werden.

Befragungstechniken

Mit Befragungstechniken wird versucht, direkt vom Stakeholder eine möglichst genaue und unverfälschte Aussage über seine Anforderungen an das System zu erhalten. Alle Befragungstechniken setzen voraus, dass der Befragte sein Wissen explizit ausdrücken kann und dass er bereit ist, Zeit und Engagement in die Ermittlung zu investieren. Befragungstech-

niken sind tendenziell vom Requirements Engineer getrieben, da dieser die Fragen vorgibt. Dadurch können Anliegen der Stakeholder eventuell verdrängt, vergessen oder vernachlässigt werden.

Klassische Befragungstechniken sind das Interview und der Fragebogen.

Kreativitätstechniken

Kreativitätstechniken dienen dazu, innovative Anforderungen zu entwickeln, die erste Vision eines neuen Systems festzulegen und Begeisterungsfaktoren zu ermitteln. Kreativitätstechniken eignen sich allerdings in der Regel nicht dazu, detaillierte Anforderungen an das Systemverhalten herauszuarbeiten.

Klassische Kreativitätstechniken sind das Brainstorming sowie eine Modifikation desselben, bei der Ergebnisse gesammelt werden, die nicht erreicht werden sollen, um so in der anschließenden Diskussion Maßnahmen zur Risikovermeidung abzuleiten, das sogenannte Brainstorming paradox.

Techniken, die mit Perspektivenwechseln arbeiten, betrachten das Problem von mehreren Seiten oder aus unterschiedlichen Sichten, während durch Analogietechniken ähnliche Problemstellungen (und Lösungen) anhand analoger Strukturen in der Natur (Bionik) oder auch außerhalb (Bisoziation) gesucht werden.

Dokumentenzentrierte Techniken

Dokumentenzentrierte Techniken verwenden Lösungen und Erfahrungen bestehender Systeme wieder. Im Falle der Ablösung eines Altsystems stellt diese Technik sicher, dass die gesamte Funktionalität des Altsystems identifiziert werden kann. Dokumentenzentrierte Techniken sollten mit anderen Ermittlungstechniken kombiniert werden, um die Gültigkeit der ermittelten Anforderungen zu bestimmen und um neue Anforderungen an das zu entwickelnde System herauszufinden.

Beobachtungstechniken

Für Situationen, in denen Fachspezialisten nicht die Zeit besitzen, das benötigte Wissen an den Requirements Engineer weiterzugeben, oder nicht fähig sind, dieses Wissen zu formulieren, eignen sich Beobachtungstechniken. Dabei werden die entsprechenden Stakeholder vom Requirements

Engineer bei ihrer Arbeit beobachtet, ihre Arbeitsschritte dokumentiert und daraus die vom System zu unterstützenden Arbeitsabläufe, aber auch potenzielle Fehler, Risiken und offene Fragen ermittelt und die Anforderungen abgeleitet. Als positiven Nebeneffekt lernt der Requirements Engineer dabei den jeweiligen Fachjargon, was ihm weitere Befragungen erleichtert. Beispiele für Beobachtungstechniken sind Feldbeobachtung und Apprenticing (»in die Lehre gehen«).

Anforderungen dokumentieren

Im Requirements Engineering werden unterschiedliche Informationen dokumentiert, die in den einzelnen Aktivitäten anfallen bzw. erarbeitet werden. Hierzu gehören z. B. Protokolle von Interviews, Berichte zu Überprüfungs- und Abstimmungsaktivitäten oder auch Änderungsanträge. Die Hauptaufgabe der Dokumentation im Requirements Engineering ist es jedoch, die Anforderungen an das zu entwickelnde System geeignet zu dokumentieren.

Die wesentlichen Gründe für die Dokumentation von Anforderungen sind:

- *Anforderungen sind Basis für die Systementwicklung:* Anforderungen werden in jeglicher Form im Requirements Engineering, beim Entwurf, in der Realisierung oder im Test direkt oder indirekt auf das Projektgeschehen einwirken. Die Qualität einer Anforderung bzw. des Anforderungsdokuments hat entscheidenden Einfluss auf den späteren Projektverlauf und somit auf den Projekterfolg.
- *Anforderungen sind rechtlich relevant:* Durch die schriftliche Dokumentation der Anforderungen können im Streitfall rechtliche Konflikte zwischen den Beteiligten zügig geklärt werden.
- *Anforderungsdokumente sind komplex:* Systeme, die Tausende von Anforderungen besitzen, die wiederum vielschichtig miteinander in Beziehung stehen, stellen in der Praxis keine Ausnahme dar. Ohne geeignete Dokumentation wird es für alle Beteiligten schwierig, den Überblick zu bewahren.
- *Anforderungen sollten allen Beteiligten zugänglich sein:* Projekte durchwandern mit der Zeit eine Entwicklung – sowohl inhaltlich als

auch personell. Durch die Gewährleistung einer permanenten Verfügbarkeit des aktuellen Informationsstands vermeidet man Unklarheiten und ermöglicht Personen, die neu in Projekt kommen, eine schnelle Einarbeitung.

Anforderungen in natürlicher Sprache dokumentieren

Die natürliche Sprache, insbesondere Prosa, ist die in der Praxis am häufigsten genutzte Dokumentationsform für Anforderungen. Gegenüber anderen Notationsformen hat Prosa einen entscheidenden Vorteil: Keiner der Stakeholder muss eine neue Notation erlernen. Weiterhin ist Sprache sehr vielseitig einsetzbar – der Requirements Engineer kann mittels natürlicher Sprache alle Arten von Anforderungen ausdrücken. Allerdings birgt die natürliche Sprache auch das Risiko von Missverständnissen und falschen Interpretationen sowie Auslassungen und impliziten Annahmen. Diese sogenannten sprachlichen »Effekte« können jedoch weitgehend vermieden werden, wenn natürlichsprachige Anforderungen systematisch formuliert und hinterfragt werden. Im folgenden Kapitel »Aus der Praxis des Requirements Engineer« stellen wir Ihnen eine bewährte Möglichkeit zur Formulierung von natürlichsprachigen Anforderungen vor.

Um Probleme zu vermeiden, die aus einem uneinheitlichen Begriffsverständnis resultieren, ist es notwendig, dass alle am Entwicklungsprozess beteiligten Personen eine konsistente Terminologie verwenden. Hierzu sind alle relevanten Begriffe in einem *Glossar* zu definieren.

Anforderungen durch konzeptuelle Modelle dokumentieren

Konzeptuelle Modelle stellen die dokumentierten Anforderungen im Vergleich zum Einsatz natürlicher Sprache kompakter und somit für den geübten Leser verständlicher dar. Zudem bieten konzeptuelle Modelle aufgrund ihrer Formalität einen höheren Grad der Eindeutigkeit (d. h. weniger Möglichkeiten zur Interpretation) als natürliche Sprache.

Allerdings setzt der Einsatz konzeptueller Modelle eine spezielle Modellierungssprache zur Dokumentation von Anforderungen und damit besondere Modellierungskenntnisse voraus [Rupp et al. 2007]. Eine solche Modellierungssprache besteht aus den Zeichen bzw. Elementen zur Dokumentation des spezifischen Modells, einer exakten semantischen Definition jedes dieser Elemente sowie eindeutigen Regeln zu deren Verknüpfung.

Zu den häufig im Requirements Engineering eingesetzten konzeptuellen Modellen zählen Zielmodelle (z. B. in Form von Und-Oder-Bäumen) und Use-Case-Diagramme sowie konzeptuelle Modelle zur Dokumentation von verfeinerten Anforderungen. Sie lassen sich drei unterschiedlichen Blickwinkeln zuordnen, die ein zu entwickelndes System aus bestimmten Sichtweisen beleuchten. Die Funktionsperspektive, Verhaltensperspektive und Strukturperspektive. Für jede dieser drei Perspektiven existieren jeweils geeignete Modellierungssprachen, um die in der Perspektive jeweils betrachteten Informationen zweckmäßig zu dokumentieren.

Funktionsperspektive

Die Modelle der Funktionsperspektive dokumentieren, welche Funktionen ein System bereitstellen muss. In diesem Zusammenhang wird dargestellt, welche Daten und Informationen vom System als Ein- und Ausgaben genutzt (Systemschnittstellen) und in welcher Art und Weise diese konkret verarbeitet werden.

Auf der obersten Ebene der Funktionsperspektive eignet sich ein Use-Case-Diagramm als konzeptuelles Modell. Use Cases stellen ideale Einstiegspunkte in eine Systementwicklung dar, weil sie bestimmte Nutzungssituationen schildern, in denen das System einem oder mehreren Nutzern (Akteuren) eine bestimmte Funktionalität zur Verfügung stellt, ohne dabei detaillierte Ablaufschritte zu beschreiben. Der Nutzer verfolgt hierbei ein bestimmtes Ziel, wünscht also ein konkretes Ergebnis, und das System liefert dieses Ergebnis durch die Ausführung der jeweiligen Systemfunktion. Abbildung 5 zeigt ein Use-Case-Diagramm mit den Use Cases sowie einem Benutzer als menschlichen Akteur. Die restlichen Akteure stellen Nachbarsysteme dar, die mit dem betrachteten System in Verbindung stehen.

Use Cases in einem Diagramm alleine reichen zur Beschreibung der Funktionsperspektive allerdings nicht aus. Mit einer Use-Case-Beschreibung können in Form einer Tabelle die einzelnen Schritte zur Umsetzung der im Diagramm verzeichneten Use Cases schriftlich dokumentiert werden, dennoch ist eine Betrachtung der Prozesse auf tieferer Ebene notwendig. Um die genaueren Abläufe und tiefer liegenden Prozesse hinter einem Use Case als Modell abbilden zu können, werden häufig Aktivitätsdiagramme eingesetzt. Ein Aktivitätsdiagramm visualisiert die Abläufe von Uses Cases, indem es einen einzelnen Use Case in eine Viel-

zahl darunterliegender Aktionen aufspaltet. Es kann jedoch auch dazu verwendet werden, eine Aktion in einem eigenen Aktivitätsdiagramm zu beschreiben.

Aktivitätsdiagramme können zur weiteren Verfeinerung um zusätzliche natürlichsprachige Anforderungen ergänzt werden.

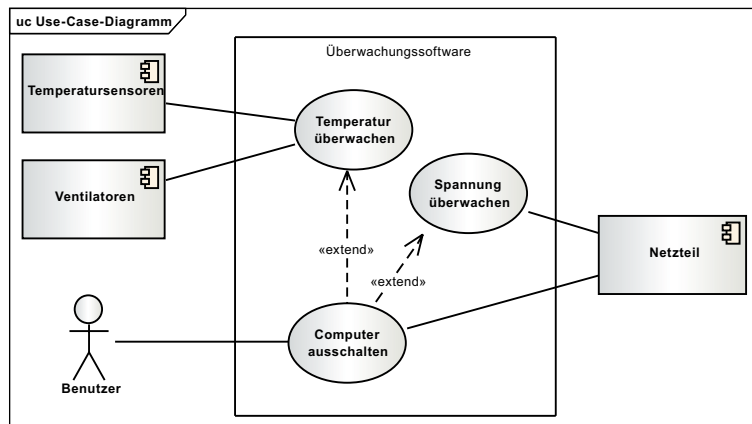


Abb. 5 Beispiel eines vollständigen Use-Case-Diagramms

Verhaltensperspektive

Der Blick auf ein System aus der Verhaltensperspektive heraus betrachtet gezielt die Zustände bzw. Zustandswechsel, die ein System, dessen Komponenten und Objekte einnehmen können. Als Diagrammtyp zum Entwurf eines solchen zustandsbasierten Verhaltens eignet sich ein UML-Zustandsautomat hervorragend. Ein Zustandsautomat kann ähnlich dem Aktivitätsdiagramm an einem Use Case anknüpfen, indem es die für einen Use Case relevanten Systemzustände abbildet. Er beschreibt dabei nicht die funktionale Sicht in Form von Abläufen und Aktionen, sondern vielmehr die Reaktionen des Systems auf bestimmte Ereignisse und Bedingungen. Verhaltens- und Funktionssicht stehen hierbei in enger Beziehung zueinander, denn der Zustandsautomat beschreibt durch seine Zustände, wann eine bestimmte Funktion zur Verfügung steht und welcher Zustand im Anschluss dieser Funktion eintritt.

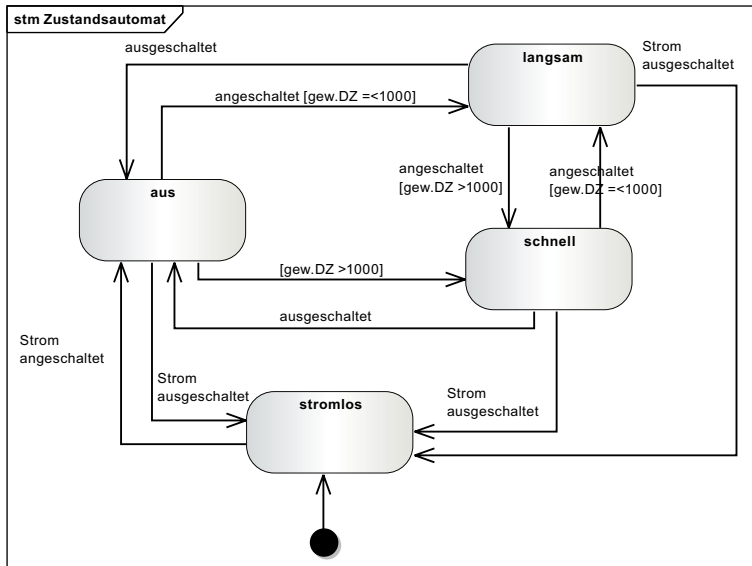


Abb. 6 Ein Zustandsautomat bildet die einzelnen Systemzustände ab.

Strukturperspektive

Dieser Perspektive beschreibt eine statische Sicht auf das System. Hier werden die Grundlagen geschaffen, auf denen die anderen beiden Perspektiven aufbauen. Modelle der Strukturperspektive formen dabei ein rein statisches Abbild des Systems, ohne auf dynamische Aspekte wie Abläufe oder Zustandsänderungen einzugehen.

Ein typisches Diagramm zur Abbildung dieser Perspektive stellt das Klassendiagramm dar. Hier werden anhand von Klassen, deren Eigenschaften und Beziehungen zueinander die fachlichen Begriffe und Daten beschrieben. Die einzelnen Klassen können dabei unterschiedliche Objekte innerhalb eines Systems darstellen, z.B. einzelne Ein- bzw. Ausgabeobjekte, die das System verarbeiten muss. Die Beziehungen, die zwischen den Klassen modelliert werden können, geben Aufschluss darüber, wie die einzelnen Klassen untereinander in Zusammenhang stehen. Hier werden jedoch keine konkreten Abläufe dargestellt, sondern lediglich statische Nutzungs- und Abhängigkeitsbeziehungen.

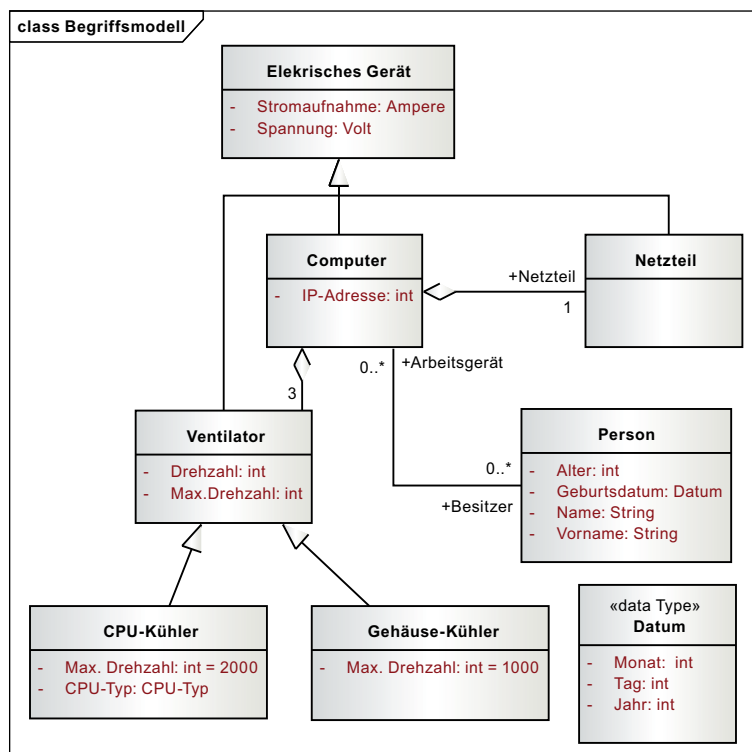


Abb. 7 Ausschnitt eines Klassendiagramms der Strukturperspektive

Speziell im Requirements Engineering kann ein Klassendiagramm mehrere Aspekte erfüllen. Zum einen dient es der Kommunikation und Verständigung, z.B. hinsichtlich der in einem Projekt oder in einer speziellen Domäne verwendeten Begriffe und deren fachlichen Zusammenhänge. Zum anderen lassen sich gezielt formalisierte Anforderungen modellieren, die in späteren Phasen einer Systementwicklung die Grundlage für das Systemdesign oder die Systemarchitektur bilden. Das obige Beispiel in Abbildung 7 zeigt den Einsatz eines Klassendiagramms als Begriffs- bzw. Informationsmodell. Bestandteile des Systems, benötigte Ein- und Ausgaben sowie mit dem System in Verbindung stehende Akteure sind

entsprechend ihren Beziehungen zueinander angeordnet. Die an den Verbindungslinien angebrachten Beschriftungen kennzeichnen die gegenseitigen Abhängigkeiten einzelner Elemente.

Konzeptuelle Modelle als Dokumentationsform für Anforderungen zu verwenden, bringt viele Vorteile mit sich, kann jedoch auch hinderlich sein. So müssen die Adressaten dieser Modelle gute Kenntnisse der jeweiligen Modellierungssprache vorweisen. Sind die Voraussetzungen für das Verständnis der Modelle gegeben, lassen sich die darzustellenden Sachverhalte wesentlich eindeutiger beschreiben als über prosaischen Text. Um die Vorteile beider Dokumentationsformen nutzen zu können, wird meist eine Kombination aus konzeptuellen Modellen und natürlicher Sprache angestrebt.

Anforderungen prüfen und abstimmen

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

Anforderungen prüfen

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

Das Ziel der Überprüfung von Anforderungen ist es somit, Fehler in den dokumentierten Anforderungen zu entdecken. Typische Beispiele für Fehler in Anforderungen sind Mehrdeutigkeit, Unvollständigkeit und Widersprüche.

Anforderungsdokumente sind Referenzdokumente für alle weiteren Entwicklungsaktivitäten. Daher beeinträchtigen Fehler in Anforderungen alle weiteren Entwicklungsaktivitäten. Ein Anforderungsfehler, der erst im Betrieb des erstellten Systems identifiziert wird, erfordert die Überar-

beitung aller Artefakte, die von dem Fehler betroffen sind, wie beispielsweise Quellcode, Testartefakte oder Architekturbeschreibungen. Die Beseitigung von Anforderungsfehlern verursacht somit erhebliche Kosten.

Zur Überprüfung der Anforderungen existieren verschiedene Techniken, die abhängig von den aktuellen Projektgegebenheiten und Zielsetzungen ausgewählt und zweckmäßig kombiniert werden sollten. Zu den verbreiteten Prüfungstechniken für Anforderungen gehören dabei die verschiedenen Ausprägungsformen des Reviews von Anforderungen (z. B. Stellungnahme, Inspektion, Walkthrough) sowie das perspektivenbasierte Lesen und der Einsatz von Prototypen und Checklisten.

Anforderungen abstimmen

Das Ziel der Abstimmung von Anforderungen ist es, unter den Stakeholdern ein gemeinsames und übereinstimmendes Verständnis bezüglich der Anforderungen an das zu entwickelnde System herbeizuführen.

Besteht hinsichtlich der Anforderungen unter den Stakeholdern ein Widerspruch in der Art, dass die Anforderungen nicht gemeinsam in einem System umgesetzt werden können, so entsteht ein Konflikt zwischen den widersprüchlichen Anforderungen und ebenso zwischen den Stakeholdern, die diese Anforderungen wünschen. Zum Beispiel könnte ein Stakeholder fordern, dass das zu erstellende System im Fehlerfall abschaltet, wohingegen ein anderer Stakeholder verlangen könnte, dass das System im Fehlerfall neu startet.

Die Akzeptanz eines geplanten Systems wird durch unaufgelöste Konflikte gefährdet, da diese dazu führen, dass die Anforderungen mindestens einer Gruppe von Stakeholdern nicht umgesetzt werden. Im schlimmsten Fall kann ein unbeachteter Konflikt der Grund dafür sein, dass die Entwicklung eines Systems nicht weiter durch die betroffenen Stakeholder unterstützt wird und dadurch die Entwicklung gänzlich scheitert.

Neben den genannten Risiken sind Konflikte allerdings auch eine Chance für das Requirements Engineering, da Konflikte zwischen Stakeholdern eine Lösung erfordern, die unter Umständen auch zur Entwicklung von neuen Ideen beitragen oder mögliche Optionen in der Entwicklung aufzeigen kann.

Anforderungen verwalten

Ziel des Verwaltens von Anforderungen (*Requirements Management*) ist es, die dokumentierten Anforderungen sowie andere relevante Informationen bis hin zu vollständigen Anforderungsdokumenten über den gesamten Lebenszyklus des Systems bzw. Produkts hinweg persistent verfügbar zu machen, sinnvoll zu strukturieren sowie den selektiven Zugriff auf diese Informationen zu gewährleisten (z. B. durch geeignete Attributierung und Sichtenbildung). Die Verwaltung von Anforderungen umfasst dabei Techniken der folgenden Kategorien:

- *Verwalten verschiedener Informationen und deren Beziehungen untereinander:* z.B. natürlichsprachige Anforderungen, konzeptuelle Modelle, Skizzen, Testpläne, Änderungswünsche usw.
- *Anforderungen mit Informationen versehen:* Um Anforderungen sinnvoll verwalten und organisieren zu können, müssen sie mit entsprechenden Metainformationen versehen werden. Dies geschieht in Werkzeugen häufig durch Anforderungsattribute wie z. B. ein eindeutiger Identifikator, Autor, Quelle, Kritikalität. Auf Grundlage einer solchen Attributierung können Anforderungen nach spezifischen Informationen gefiltert und in sogenannten Sichten abgebildet werden (Sichtenbildung).
- *Workflow-Management:* Anforderungen durchlaufen im Zuge eines Entwicklungsprozesses unterschiedliche Zustände. Diese Zustände müssen ersichtlich und nachvollziehbar abgebildet und durch ein angemessenes Rechte- und Rollenkonzept gesteuert werden können. Anforderungen werden von unterschiedlichen Rollen erstellt, geprüft und freigegeben bzw. überarbeitet. Eine Festlegung der Verantwortlichkeiten ist für die Verwaltung von Anforderungen daher unumgänglich.
- *Strukturieren von Anforderungen:* Anforderungen müssen ihren Abhängigkeiten und Eigenschaften nach verwaltet werden können. Im Falle von Anforderungen unterschiedlicher Ebenen müssen diese hierarchisch angeordnet werden. Eine Strukturierung von Anforderungen gemäß ihres Kontextes ist ebenfalls wichtiger Teil der Verwaltung.

- *Priorisierung von Anforderungen:* Anforderungen werden zu verschiedenen Zeitpunkten in verschiedenen Aktivitäten nach unterschiedlichen Kriterien priorisiert. Je nach Priorisierungsziel und Priorisierungsgegenstand sind für die Priorisierung unterschiedliche Techniken einzusetzen.
- *Verfolgbarkeit von Anforderungen:* Im Rahmen der Verfolgbarkeit, auch Traceability genannt, werden Verfolgbarkeitsinformationen von Anforderungen aufgezeichnet, organisiert und gepflegt, um Informationen über Querbezüge und Abhängigkeiten von Anforderungen untereinander oder von Anforderungen zu anderen Entwicklungsartefakten nutzen zu können. Hierbei ist wichtig, dass jede Anforderung eine individuelle Kennung in Form einer eindeutigen ID erhalten muss. Eine solche ID kann für jede Anforderung z.B. über Attributierung vorgenommen werden.
- *Versionierung von Anforderungen:* Die Versionierung und Konfiguration von Anforderungen ermöglicht es, über den Lebenszyklus eines Systems oder Produkts hinweg spezifische Entwicklungsstände von Anforderungen und Anforderungsdokumenten verfügbar zu halten.
- *Änderungsmanagement von Anforderungen:* Für die Bearbeitung der Änderungsanträge ist typischerweise das sogenannte Change Control Board zuständig. Das Change Control Board entscheidet über die Annahme bzw. die Ablehnung von Änderungsanträgen. Im Vorfeld der Entscheidung priorisiert es die Änderungsanträge und schätzt die Auswirkungen der Änderung auf alle Entwicklungsartefakte sowie die zur Umsetzung der Änderung benötigten Ressourcen ab.

Aus der Praxis des Requirements Engineer

Nachdem wir Ihnen in den bisherigen Kapiteln das theoretische Fundament des Requirements Engineering vorgestellt haben, gehen wir in diesem Kapitel auf zwei praktische Aspekte ein.

Zunächst zeigen wir Ihnen eine bewährte Methode, natürlichsprachige Anforderungen anhand einer Satzschablone so zu formulieren, dass jede Anforderung alle für ihre Umsetzung relevanten Informationen transportiert.

Danach stellen wir Ihnen die Eigenschaften und Vorzüge spezialisierter Werkzeuge für das Requirements Management vor und zeigen Ihnen auf, warum klassische Büroanwendungen für die Verwaltung von Anforderungen eigentlich gänzlich ungeeignet sind.

Die Satzschablone

Die oft genannten Vorteile der Dokumentation in natürlicher Sprache sind die Lesbarkeit und die universelle Anwendbarkeit für verschiedenste Sachverhalte ohne besonderes Vorwissen bezüglich der Notation. Demgegenüber stehen aber die Nachteile, die aus der fehlenden Formalisierung stammen, z. B. die Mehrdeutigkeit. Da Projektbeteiligte aufgrund unterschiedlichen Wissens, sozialer Prägung und Erfahrung die dokumentierten Anforderungen verschieden interpretieren, kommt es in der Praxis immer wieder zu Missverständnissen. Diese Nachteile können durch die Verwendung einer Satzschablone und das Prüfen auf sprachliche Effekte vermindert werden.

Eine Satzschablone (*Requirement Template*) ist ein Bauplan für die syntaktische Struktur einer einzelnen Anforderung. Der Einsatz der Satzschablone unterstützt den Autor einer Anforderung darin, die syntaktische Eindeutigkeit der Anforderung zu erreichen und Anforderungen hoher Qualität in einem optimalen Zeit- und Kostenrahmen zu verfassen.

Die Satzschablone sollte dann eingesetzt werden, wenn sich in den Projekten die Bereitschaft der Mitarbeiter herauskristallisiert, einer stark normierten Vorgehensweise zu folgen, denn die stilistischen Freiheitsgrade der Schreibenden bei der Formulierung von Anforderungen werden dabei stark eingeschränkt. Den besten Erfolg erzielt man, wenn man die

Satzschablonen nicht als ein Muss vorschreibt, sondern die Methode schult und die Schablone als Hilfsmittel darstellt.

Die richtige Anwendung der Schablone erfordert fünf Schritte:

Schritt 1: Festlegen der rechtlichen Verbindlichkeit

Legen Sie als Erstes die rechtliche Verbindlichkeit der Anforderung fest. Eine Möglichkeit, die Verbindlichkeit einer Anforderung auszudrücken, stellen die Modalverben »muss«, »sollte« und »wird« dar.

Schritt 2: Den Kern der Anforderung bestimmen

Im Mittelpunkt jeder Anforderung steht die geforderte Funktionalität (z. B. »drucken«, »speichern«, »übertragen«, »berechnen«). In Schritt 2 wählen Sie das Verb (Prozesswort), das den benötigten Vorgang oder die Tätigkeit beschreibt.

Schritt 3: Charakterisieren der Aktivität des Systems

Für funktionale Anforderungen lässt sich die Systemtätigkeit in drei relevante Arten klassifizieren:

Typ 1: Selbstständige Systemaktivität:

Das System führt den Prozess selbstständig durch.

Typ 2: Benutzerinteraktion:

Das System stellt dem Nutzer die Prozessfunktionalität zur Verfügung.

Typ 3: Schnittstellenanforderung:

Das System führt einen Prozess in Abhängigkeit von einem Dritten (z.B. einem Fremdsystem) aus, ist an sich passiv und wartet auf ein externes Ereignis.

In Schritt 3 ist anhand der Art der geforderten Systemaktivität einer der drei Typen der Satzschablone zu wählen.

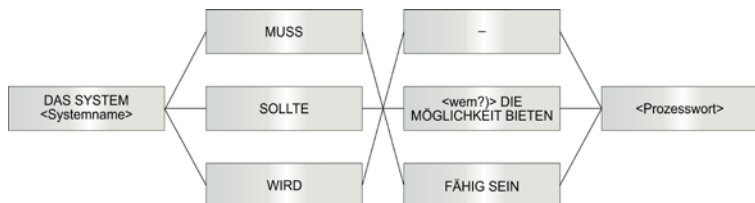


Abb. 8 Die Satzschablone nach Schritt 3

Lesart der Schablonentypen:

Typ 1: DAS SYSTEM MUSS/SOLLTE/WIRD <Prozesswort>.

Typ 2: DAS SYSTEM MUSS/SOLLTE/WIRD <wem?> DIE
MÖGLICHKEIT BIETEN <Prozesswort>.

Typ 3: DAS SYSTEM MUSS/SOLLTE/WIRD FÄHIG SEIN
<Prozesswort>.

Schritt 4: Objekte einfügen

Manche Prozesswörter benötigen zur Ergänzung ein oder mehrere Objekte. In Schritt 4 werden daher die noch fehlenden Objekte und Ergänzungen der Objekte in der Anforderung identifiziert und ergänzt.

Schritt 5: Formulierung von logischen und zeitlichen Bedingungen

Für Anforderungen ist es typisch, dass die geforderte Funktionalität nicht fortwährend, sondern nur unter bestimmten zeitlichen oder logischen Bedingungen ausgeführt oder zur Verfügung gestellt wird. Um zeitliche von logischen Bedingungen klar unterscheiden zu können, wählen wir für zeitliche Bedingungen als temporale Konjunktion »sobald«, für logische Bedingungen als konditionale Konjunktion »falls« und stellen die Bedingung an den Satzanfang.

Werkzeuge in der Anforderungsverwaltung

Die verschiedenen Aktivitäten im Requirements Engineering werden durch geeignete Werkzeuge unterstützt, die idealerweise integriert sind und die jeweils abgelegten Informationen nutzen und weiterverarbeiten können. Dies können Informationen sein, die im Requirements Engineering erzeugt wurden (z. B. natürlichsprachige oder modellbasierte Anforderungen) oder als Basis für die Anforderungen genutzt wurden (z. B. Besprechungsprotokolle, Zieldokumentationen, Stakeholderlisten). Viele Werkzeuge, die eigentlich ein anderes Einsatzgebiet haben, können auch im Requirements Engineering Verwendung finden.

Die in der Praxis bekannteste Form von Werkzeugen für das Requirements Engineering sind solche, die auf die Verwaltung von Anforderungen (*Requirements-Management-Werkzeuge*, kurz RM-Werkzeuge) ausgelegt sind.

Jedoch werden in vielen Projekten Standard-Büroanwendungen für die Verwaltung von Anforderungen genutzt, auch wenn ihre Nachteile gegenüber spezialisierten RM-Werkzeugen nicht zu unterschätzen sind.

Allgemeine Werkzeugunterstützung im Requirements Engineering

Eine Vielzahl von Werkzeugen, die im Rahmen der Systementwicklung eingesetzt werden, kann neben dem eigentlichen Einsatzgebiet auch für Aufgaben des Requirements Engineering genutzt werden. So bieten Testverwaltungs-, Fehlerverfolgungs- oder Konfigurationsmanagement-Werkzeuge oftmals Funktionen zur Anforderungsverwaltung oder können mithilfe von Anpassungen dazu eingesetzt werden. Ein Vorteil beim Einsatz dieser Werkzeuge z. B. zur Verwaltung von Anforderungen ist die Integration, die zwischen Anforderungen und den Entwicklungsartefakten erreicht wird, für die diese Werkzeuge ursprünglich konzipiert wurden (wie Testfälle oder Änderungsanträge). Werden z. B. die Anforderungen im Testverwaltungswerkzeug gepflegt und nicht separat in einem RM-Werkzeug, so fällt eine Werkzeugschnittstelle weg, und die Verknüpfung von Anforderung und zugehörigen Testfällen kann leichter erfolgen.

Wiki-Technologien können zum kooperativen Aufbau von Glossaren genutzt werden, im Brainstorming entworfene Mindmaps können als Gliederungsstruktur dienen, mit Präsentationswerkzeugen kann man ein grobes Analysekonzept erzeugen.

Des Weiteren werden im Requirements Engineering auch Werkzeuge, die die alltäglichen Arbeitsabläufe im Office-Betrieb unterstützen, eingesetzt: von Mailsystemen über Chatsoftware, Adressbücher, Terminplaner, Groupware-Plattformen bis hin zu Werkzeugen für das Projektmanagement bzw. für die Projektplanung und das Projektcontrolling. Sie unterstützen die Stakeholder im Requirements Engineering bei der Kommunikation sowie der Planung und Koordination von Aufgaben.

Neben den natürlichsprachigen Informationen werden im Requirements Engineering auch Informationen in Form von Modellen dokumentiert, die mithilfe von Modellierungswerkzeugen erstellt werden. Diese Werkzeuge bieten nicht nur die Möglichkeit, Modelle zu erstellen, sondern verfügen oftmals auch über Funktionen zur syntaktischen Analyse dieser Modelle.

Eine Fragestellung, die sich im Zusammenhang mit dem ergänzenden Einsatz verschiedener Werkzeuge ergibt, ist die Integration und Verfolgbarkeit zwischen Artefakten der verschiedenen Werkzeuge (z. B. Use Cases, Verhaltensmodelle und Testfälle). Die Wahl des Modellierungswerkzeugs oder des Werkzeugs für das Requirements Management ist so zu treffen, dass eine Schnittstelle zwischen den beiden Werkzeugen besteht oder geschaffen werden kann, die es ermöglicht, Verfolgbarkeitsbeziehungen zwischen Artefakten (Anforderungen, Modellelementen, Anforderungsdokumenten, Testfällen usw.) zu setzen und zu verwalten. Wenn sich Anforderungen ändern, ist es unerlässlich, dass die entsprechenden Änderungen auch an den betroffenen Modellelementen vorgenommen werden können. In gleicher Weise gilt dies für Änderungen in Anforderungsmodellen, die ggf. auch in die entsprechenden natürlichsprachigen Anforderungen integriert werden müssen.

Requirements-Management-Werkzeuge

Einschlägige Requirements-Management-Werkzeuge bieten Unterstützung für die Verwaltung von Anforderungen. Ein gutes RM-Werkzeug kann die im Abschnitt »Anforderungen verwalten« beschriebenen Prinzipien durch seine angebotenen Funktionen umsetzen.

Dem angebotenen Funktionsumfang nach und der damit verbundenen Abdeckung der grundlegenden Requirements-Management-Prinzipien lassen sich Werkzeuge, die für das Requirements Management eingesetzt werden können, in zwei Kategorien gruppieren: spezialisierte Werkzeuge sowie Standard-Büroanwendungen.

Spezialisierte RM-Werkzeuge wurden speziell zur Unterstützung der verschiedenen Techniken zur Verwaltung von Anforderungen entwickelt und unterstützen die Aufgaben in der Verwaltung von Anforderungen am besten. Die charakteristischen Eigenschaften solcher Werkzeuge sind:

- Vergabe von Attributen für unterschiedliche Informationsarten (Anforderungen, Modelle, Ziele usw.), um die zur Verwaltung notwendigen Metainformationen darstellen zu können.
- Setzen von Verlinkungen zur Gewährleistung der Verfolgbarkeit (Traceability) zwischen unterschiedlichen Informationsarten untereinander (z.B. zwischen Anforderungen und Testfällen)
- Organisation untereinander abhängiger Anforderungen (mittels Hierarchieebenen und Gruppierungen bzw. Collections)
- Konfigurations- und Versionsmanagement auf Anforderungsebene durch automatische Versionierung und Änderungsprotokollierung
- Definition von Anforderungsbasislinien (Baselining)
- Mehrbenutzerzugriff und -verwaltung zur Umsetzung eines Rechte- und Rollenkonzeptes (z. B. Zugriffskontrolle)
- Analysemöglichkeiten: Eine automatische Erkennung von Änderungen entlang zuvor gesetzter Verlinkungen (Impact Analysis) ist zwischen einzelnen miteinander verknüpften Informationsarten möglich. So lassen sich Auswirkungen auf Anforderungen, die mit zu ändernden Anforderungen in Beziehungen stehen, genau lokalisieren.
- Konsolidierung der erfassten Anforderungen mit unterschiedlichen Stakeholdergruppen und Rollen (z. B. Sichtenbildung)
- Unterstützung des Änderungsmanagements (Änderungskontrolle) mithilfe eines entsprechenden Workflow-Mechanismus (Change Requests, Defect Reporting)
- Erstellen von Reports, Auswertungen oder umfangreichen Ergebnisdokumenten von den verwalteten Informationen (z. B. Reports über Änderungsanträge für Anforderungen, Statistiken oder Anforderungsdokumente für spezifische Systemreleases)

Die verschiedenen am Markt verfügbaren RM-Werkzeuge besitzen einen ähnlichen Aufbau. So verfügen die einschlägigen Werkzeuge über eine Benutzungsoberfläche, über die der Anwender mit dem Werkzeug arbeitet und alle zur Verfügung gestellten Funktionen nutzen kann. Die verwalteten Daten werden in einer Datenbank gespeichert und über einen integrierten Editor eingegeben und geändert. Verschiedene Import- und Exportmechanismen für Dokumente und Reports stellen sicher, dass Daten von externen Systemen in das Werkzeug eingelesen sowie aus dem Werkzeug in externe Systeme ausgeleitet werden können.

Requirements-Management-Werkzeuge erreichen damit eine sehr hohe Abdeckung der grundlegenden Funktionen.

Eine vergleichende Auflistung der gängigsten Requirements-Management-Werkzeuge finden Sie unter:

<http://www.incose.org/productspubs/products/rmsurvey.aspx>

Standard-Büroanwendungen

In einer Vielzahl von Projekten werden jedoch trotz der wachsenden Verbreitung spezialisierter RM-Werkzeuge nach wie vor Standard-Büroanwendungen (z. B. Textverarbeitung und Tabellenkalkulation) zur Verwaltung von Anforderungen eingesetzt. Die hauptsächlichen Gründe hierfür liegen einerseits in der weiten Verbreitung dieser Werkzeuge und andererseits darin, dass zur Nutzung kein zusätzlicher Schulungs- bzw. Einarbeitungsaufwand notwendig ist. In Verbindung mit dem Einsatz von Schablonen, wie z. B. Schablonen zur Anforderungsdokumentation (siehe vorheriger Abschnitt), eignen sich diese Werkzeuge für die Dokumentation und eingeschränkt für die Verwaltung von Anforderungen (z. B. können Verfolgbarkeitsbeziehungen durch Hyperlinks etabliert werden). Viele wichtige Grundeigenschaften eines professionellen RM-Werkzeugs von Versionsverwaltung bzw. Baselineing über Workflow-Management-Funktionen bis hin zur Benutzerverwaltung fehlen allerdings.

Literatur

- [Boehm 1981] B. Boehm: Software Engineering Economics, Prentice Hall, Englewood Cliffs 1981.
- [IEEE 1990] Institute of Electric and Electronic Engineers: IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990), IEEE Computer Society, New York 1990.
- [Rupp et al. 2007] C. Rupp, S. Queins, B. Zengler: UML 2 Glasklar, Hanser Fachbuchverlag, München, Wien 2007.

Weiterführende Informationen

Die vorliegende Broschüre basiert auf *Klaus Pohl, Chris Rupp: »Basiswissen Requirements Engineering«,* offizielles Lehrbuch für die Zertifizierung zum Certified Professional for Requirements Engineering – Foundation Level, 3. korrigierte Aufl., dpunkt.verlag, Heidelberg 2011.

Das Buch »Basiswissen Requirements Engineering« basiert auf den beiden auflagenstärksten Büchern zum Thema Requirements Engineering: *Chris Rupp und die SOPHISTen: Requirements-Engineering und -Management – Professionelle, iterative Anforderungsanalyse für die Praxis*, Hanser Fachbuchverlag, München 2009.

sowie

Klaus Pohl: Requirements Engineering – Grundlagen, Prinzipien, Techniken, 2. korrigierte Aufl., dpunkt.verlag, Heidelberg 2008.

Informationen zum International Requirements Engineering Board e.V. sowie zur Zertifizierung zum *Certified Professional for Requirements Engineering* finden Sie unter: <http://www.certified-re.de>.

Informationen zum Thema Requirements Engineering finden Sie unter www.sophist.de.

Außerdem bei Facebook unter <http://de-de.facebook.com/SOPHIST.GmbH> sowie bei XING unter <http://www.xing.com/companies/SOPHISTGMBH>.

Durchtrainiert?

Wir machen Sie wirklich fit!

Mit Offenen Trainings, Beratung und Coaching in den Bereichen Requirements Engineering, UML sowie Geschäftsprozessanalyse.



Eine Auswahl unserer Trainingsangebote:

- CPRE Foundation Level
- CPRE Advanced Level – Elicitation and Consolidation
- CPRE Advanced Level – Requirements Modeling
- Requirements Engineering in der Praxis
- Requirements Engineering & Scrum

Alle Trainings können auch individuell auf Ihre Anforderungen angepasst werden. Sprechen Sie mit uns. Gemeinsam entwickeln wir Ihren optimalen Trainingsplan.

SOPHIST GmbH
Vordere Cramergasse 13
90478 Nürnberg, Deutschland

fon: +49 (0)9 11 40 900-0
fax: +49 (0)9 11 40 900-99
e-mail: heureka@sophist.de

Kleben Sie sich die RE- & OO-Experten an die Wand!

Der legale Spickzettel für den Requirements Engineer und Modellierer.

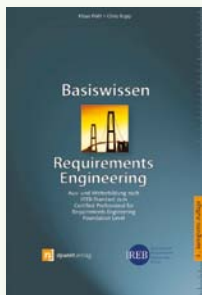


Schicken Sie uns eine E-Mail an heureka@sophist.de mit dem Codewort „CPRE-Booklet“ und erhalten kostenlos das Plakat zugesendet.

Übrigens, wenn Sie noch tiefer in die UML einsteigen wollen:
Das Buch „UML 2 Glasklar“ können wir Ihnen wärmstens empfehlen!

SOPHIST GmbH
Vordere Cramergasse 13
90478 Nürnberg, Deutschland

fon: +49 (0)9 11 40 900-0
fax: +49 (0)9 11 40 900-99
e-mail: heureka@sophist.de



Klaus Pohl · Chris Rupp

Basiswissen Requirements Engineering

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

3., korrigierte Auflage
2011, 192 Seiten, Festeinband
€ 29,90 (D)
ISBN 978-3-89864-771-7

Der »Certified Professional for Requirements Engineering« hat sich als international standardisiertes Aus- und Weiterbildungsprogramm etabliert. Dieses Lehrbuch ist das erste für die Zertifizierung zum Foundation Level, geschrieben von Mitgliedern des International Requirements Engineering Board (IREB), die den Lehrplan mit entwickelt haben. Es umfasst Grundlagenwissen in den Gebieten Ermittlung, Dokumentation, Prüfung und Verwaltung von Anforderungen und eignet sich zum Selbststudium sowie als Begleitliteratur zu Schulungen.



Klaus Pohl

Requirements Engineering

Grundlagen, Prinzipien, Techniken

2., korrigierte Auflage
2008, 764 Seiten, Festeinband
€ 49,00 (D)
ISBN 978-3-89864-550-8

Requirements Engineering ist der Schlüssel zur erfolgreichen Entwicklung von Software und softwareintensiven Systemen. Das moderne Requirements Engineering gewinnt neue, innovative Anforderungen an solche Systeme und überführt sie in detaillierte, möglichst vollständige Anforderungsspezifikationen.

Das Buch bietet eine umfassende Einführung in die Grundlagen, Prinzipien und Techniken des Requirements Engineering. Die kompakte Darstellung der Inhalte sowie deren Strukturierung anhand eines Rahmenwerks machen es zugleich zu einem Nachschlagewerk für Praxis, Lehre und Forschung.

Mit zahlreichen Beispielen, Checklisten und praktischen Hinweisen.



Andreas Spillner · Tilo Linz

Basiswissen Softwaretest

Aus- und Weiterbildung zum Certified Tester –
Foundation Level nach ISTQB-Standard

4., überarbeitete und aktualisierte Auflage

2010, 308 Seiten, Festeinband

€ 39,90 (D)

ISBN 978-3-89864-642-0

Andreas Spillner · Thomas Roßner · Mario Winter · Tilo Linz

Praxiswissen Softwaretest – Testmanagement

Aus- und Weiterbildung zum Certified Tester –
Advanced Level nach ISTQB-Standard

3., überarbeitete und erweiterte Auflage

2011, 464 Seiten, Festeinband

€ 42,90 (D)

ISBN 978-3-89864-746-5



Graham Bath · Judy McKay

Praxiswissen Softwaretest – Test Analyst und Technical Test Analyst

Aus- und Weiterbildung zum Certified Tester –
Advanced Level nach ISTQB-Standard

2., durchgesehene Auflage

2011, 432 Seiten, Festeinband

€ 42,90 (D)

ISBN 978-3-89864-735-9

Sachar Paulus

Basiswissen Sichere Software

Aus- und Weiterbildung zum ISSECO Certified
Professionell for Secure Software Engineering

2011, 286 Seiten, Festeinband

€ 39,90 (D)

ISBN 978-3-89864-726-7





Chris Rupp

Requirements Engineering

Requirements Engineering ist eine Schlüsseldisziplin der Systementwicklung und entscheidet maßgeblich über den Erfolg oder Misserfolg eines Projekts.

Mit dieser Broschüre erhalten Sie einen Einstieg in die vielschichtige Disziplin des Requirements Engineering. Sie erfahren, was genau Requirements Engineering ist, und lernen dessen verschiedene Tätigkeitsfelder kennen: die Ermittlung, Dokumentation, Validierung und Verwaltung von Anforderungen. Dabei wird Grundsätzliches über das Berufsbild und das Persönlichkeitsprofil des Requirements Engineer vermittelt, und Sie bekommen einen Einblick in einige Aspekte seiner täglichen Berufspraxis.

Die Broschüre basiert auf »Basiswissen Requirements Engineering«, dem offiziellen Lehrbuch für die Zertifizierung zum *Certified Professional for Requirements Engineering (CPRE) – Foundation Level nach dem IREB-Standard*, verfasst von Klaus Pohl und Chris Rupp, den Autoren der auflagenstärksten deutschsprachigen Bücher zum Thema Requirements Engineering.