

Soziologisch/Psychologische Aspekte des Requirements Engineering

Mathias Landhäußer
Universität Karlsruhe

19. Juli 2007

Zusammenfassung

Zunächst wird „Requirements Engineering“ und Qualitätssicherung für Anforderungen motiviert. Dann beschäftigt sich dieser Artikel mit den Problemen der Anforderungsermittlung und deren psychologischen Ursachen. Der Schwerpunkt liegt dabei auf der Kommunikation zwischen den Anforderungsermittlern und den Beteiligten, sowie den Fehlern, die aufgrund der Eigenschaften der natürlichen Sprache entstehen. Abschließend werden vorbeugende und korrigierende Maßnahmen betrachtet, die von Anforderungsermittlern direkt angewendet werden können.

1 Einleitung

„The motivation for RE is simple: to reduce the high cost of misunderstanding between users and designers, so that computer systems are built to do what the users want, on time and at a reasonable cost.“ [Sut02, S. 4]

1.1 Requirements Engineering ist grundlegend

Die Anforderungsermittlung ist der grundlegende Abschnitt der Softwareentwicklung. Selbstverständlich ist es wichtig, dass die Anforderungen bei der Implementierung vollständig und korrekt umgesetzt werden; was jedoch wird umgesetzt, wenn es schlechte, falsche oder unvollständige Anforderungen gibt? Diese simple Tatsache ist nicht zu verachten, werden wir doch meist darauf aufmerksam gemacht, wenn ein prestigeträchtiges Projekt zu spät fertig gestellt wird, nachgebessert werden muss oder es ganz scheitert.

In den Anfangszeiten des Requirements Engineering (RE) wurde ein besonderes Augenmerk auf ein ingenieurmäßiges Herangehen an die Anforderungsermittlung gelegt [Hof00]. Der Fokus wurde dabei auf die Konsistenz der natürlichsprachlichen Anforderungen mit der Spezifikation und die Konsistenz des Produktes mit der Spezifikation gelegt; Praktiker hingegen wurden mit den Problemen der Anforderungsermittlung weitestgehend alleine

gelassen. Man befindet sich in der Zwickmühle, dass die natürliche Sprache mehrdeutig sein kann, eine rein formale Spezifikation der Anforderungen jedoch Kommunikationsschwierigkeiten nach sich zieht. Eine mögliche Lösung ist die Strukturierung der natürlichsprachlichen Anforderungen.¹ Mittlerweile schloss die Forschung diese Lücke und gibt allen Beteiligten verschiedene Werkzeuge und Methoden an die Hand, mit denen Anforderungen systematisch erfasst und dokumentiert werden können.

1.2 Warum gute Anforderungen erheben?

Offenbar kann nur durch das Erheben von Anforderungen überhaupt ein System erstellt werden, das die Ziele des Kunden erfüllt. Wenn eine hohe Anforderungsqualität erreicht werden kann, profitieren alle an der Softwareentwicklung Beteiligten vom RE. Nach Hood und Wiebel [HW05, S. 4]

- führen gute Anforderungen zu größerem (wirtschaftlichem) Erfolg,
- setzen die Entwicklungszeiten herab,
- senken gleichzeitig das unternehmerische Risiko,
- setzen die Testzeiten herab und erhöhen die Testqualität,
- erhöhen somit die Gesamtqualität der produzierten Software und
- ziehen schließlich bessere Systeme nach sich.

Bei der Softwareentwicklung wird viel Zeit auf Änderungs- und Nacharbeiten verwendet, welche aufgrund schlecht erfasster Anforderungen notwendig werden. Fehlerhafte Anforderungen werden umso teurer, je später sie entdeckt werden. Die Kostenexplosion der Fehlerbehebung führt nicht selten dazu, dass ein Projekt scheitert oder verzögert wird. Dies kann nicht nur dazu führen, dass das Vertrauen in die Entwicklungskompetenz eines Unternehmens sinkt, sondern dass unentdeckte Anforderungsfehler zu lebensbedrohlichen Situationen oder Unfällen führen können: Beispielsweise war es möglich den Airbus A320 zu starten, obwohl sich die Hydrauliksteuerung der Klappen im Wartungsmodus befand. Es gab keinen Punkt auf der Checkliste der Piloten, dies zu prüfen und es gab keine Signalleuchte im Cockpit. Nach dem Start stellten die Piloten fest, dass sich das Flugzeug nicht stabilisieren ließ und vermuteten eine Fehlfunktion. Die Notfallprozeduren konnten über die Menüführung der Cockpitterminals nicht aufgerufen werden, lediglich in der („alten“) Papierform der Anleitung waren die Anweisungen zu finden.

¹Beispielsweise mit Schemata wie IEEE 830 oder Volere oder mithilfe von Software wie Rational Analyst Studio (IBM), Rational RequisitePro (IBM), DOORS oder CRADLE.

Die zentrale Frage scheint somit zu sein, wie die unterschiedlichen Vorstellungen der beteiligten Personen vom Produkt konfliktbereinigend zusammengeführt werden können, ohne Anforderungen zu vergessen oder zu übersehen. Oftmals werden persönliche Auffassungen oder (konkrete) Vorstellungen vom Endprodukt nicht mitgeteilt oder vom Designer stillschweigend angenommen; jeder der am Entwicklungsprozess beteiligt ist, legt seiner Tätigkeit seine persönlichen Erfahrungen und Einstellungen zugrunde – und dies meist unbewusst. Viele Anforderungen werden jedoch erst entdeckt, wenn bereits mit der Entwicklung begonnen wurde. Wie also können die – teilweise un- oder unterbewussten – Anforderungen ermittelt werden? Entscheidend ist die Frage, wie wir Menschen *richtig verstehen* und welche Methoden uns dabei unterstützen können.

2 Probleme bei der Anforderungsermittlung

2.1 Einführung

Zunächst werden wir uns damit beschäftigen, wie Anforderungen ermittelt werden und an welcher Stelle welche Probleme auftreten. Dann werden wir einige Methoden betrachten, mit denen man den Problemen vorbeugen kann. Zuletzt werden wir sehen, wie wir welche Probleme im Nachhinein beheben können.

Die Ermittlung von Anforderungen ist naturgemäß schwierig: Der Ermittler muss sich mit Menschen befassen – Diese Menschen haben eine Vorstellung von dem, was gebaut werden soll; oft ist es jedoch so, dass sie dies erst ausdrücken (können), wenn sich jemand mit den Bedürfnissen auseinandersetzt und diese hinterfragt. Als Ermittler muss man sich darüber hinaus mit den Problemen der zwischenmenschlichen Kommunikation auseinandersetzen. Das bedeutet insbesondere, dass man wissen muss, welche Auswirkungen implizites Wissen, Mehrdeutigkeit, und Gefühle bzw. Meinungen auf die Beantwortung von Fragen haben. Hinzu kommt, dass Menschen nicht wissen, was sie wollen, bis sie es bekommen; wenn sie es bekommen, sehen sie vor allem, was verbesserungswürdig ist, oder sie mögen das Ergebnis im ganzen nicht [Sut02, S. 10] (siehe „Ja-Aber“-Syndrom auf Seite 6).

Kommunikation ist wichtig

„Anforderungen müssen ermittelt werden.“ Dieser kurze Satz trifft den Kern des RE [BGB01]: Anforderungen können nicht einfach beim Kunden in einer schönen, vollständigen und richtigen Liste abgeholt werden, sie müssen systematisch ermittelt und verifiziert werden. Diese Aufgabe ist nur dann zu bewältigen, wenn ein reger Austausch zwischen allen Beteiligten (*stakeholder*) und dem Ermittler stattfindet. Boehm legt dar [BI99], wie zwischen den 70er und 80er Jahren der Übergang vom vertragsbasierten Ansatz zum benutzerorientierten Ansatz dazu führte, dass teure, nicht wartbare Systeme gebaut wurden. Oftmals wurde aufgrund der unzureichenden Anforderungen der Beteiligten Systeme entwickelt, die zwar „nette“ Funktionen bereit-

stellten, jedoch nicht skalierbar oder portierbar waren. Boehm führte diese Probleme darauf zurück, dass die Beteiligten nicht genau wüssten, was sie benötigten.

Alle Beteiligten sind mitverantwortlich dafür, dass die relevanten Fragen diskutiert und die wichtigen Probleme gelöst werden. Die Hauptverantwortung liegt jedoch beim Ermittler. Er ist dafür zuständig, die Anforderungen der Beteiligten in softwaretechnisch korrekte Anforderungen zu überführen und neue Ideen oder Vorschläge einzubringen (Lösungsansätze, an die die Beteiligten nicht gedacht haben bzw. von denen sie nicht wussten, dass sie überhaupt möglich sind). Im Zweifel muss er die Arbeit der Beteiligten analysieren und bewerten – diese Aufgabe wird besonders schwer, wenn es sich bei Teilen der Arbeitsabläufe um implizites Wissen (für nähere Informationen siehe [NT97]) handelt. Oftmals werden die im Arbeitsablauf der Mitarbeiter auftretenden Probleme einfach behoben, ohne dass hinterher eine Begründung oder eine nachvollziehbare Entscheidungsgrundlage vorhanden ist [Dut06]. Eine besonders schwere Aufgabe ist es also, den Beteiligten ihr implizites Wissen zu entlocken.

Die Hauptaufgabe des Ermittlers ist es, das festzuhalten was die Beteiligten wirklich meinen [Rup02], nicht das was sie mitteilen. Dies kann auf informale (Prosatexte), semi-formale oder formale Art geschehen. Dawson und Swatman haben 1999 festgestellt [DS99], dass sehr häufig informale Modelle von den Ermittlern entgegengenommen und ad-hoc in semi-formale oder formale überführt werden. Die informalen Dokumente werden bei der Kommunikation mit den Beteiligten zu Rate gezogen und bestimmen die Verhandlungen mit dem Kunden. Im internen Entwicklungsprozess wird jedoch auf die formalisiertesten Anforderungen zurückgegriffen (beispielsweise bei der Kommunikation zwischen dem Ermittler und dem Designer). Dies hat zur Folge, dass Änderungen (aufgrund von Fehlern, Inkonsistenzen usw.) nur in die formalen Beschreibungen einfließen, nicht jedoch in die informalen Anforderungsdokumente; ihre Studien legen zudem nahe, dass Projektverantwortliche davon ausgehen, dass die beim Kunden Beteiligten die formalen Modelle nicht verstehen können. Abhilfe könnte nach Dawson und Swatman UML oder ähnliche OO-Modelle schaffen, da OO-Modelle von Nutzern verstanden und gleichzeitig von Designern und Entwicklern verwendet werden können; so entfielen die ad-hoc Transformation.

Im Ermittlungsprozess selbst sind psychologische Kenntnisse des Ermittlers unentbehrlich, um gruppenspezifische Prozesse zu verstehen und daraus resultierende Probleme vermeiden zu können. So muss beispielsweise vermieden werden, dass in einer Gruppe eine dominante Person die Leitung übernimmt und die übrigen Gruppenmitglieder nur noch der Meinung ihres Anführers folgen. Ebenso sollten große Hierarchiesprünge innerhalb einer Gruppe vermieden werden, wenn mittels Brainstorming, Workshops oder Ähnlichem Anforderungen ermittelt und Vereinbarungen ausgehandelt werden sollen. Große Hierarchiesprünge können zum Beispiel dazu führen,

dass untergeordnete Mitarbeiter nicht zur Sprache kommen oder sich nicht trauen, ihren Vorgesetzten zu kritisieren und so für den Ermittler wertvolle Informationen verloren gehen.

Neben diesen Problemen sprechen Horner und Atwood von vier so genannten fundamentalen Barrieren, welche ursächlich für die unvollständigen Anforderungen seien [Dut06][S.78]:

1. Kognitive Beschränktheit: Die kognitiven Fähigkeiten der Beteiligten sind beschränkt, was sich in einem unvollständigen oder falschen Problemverständnis niederschlägt. Da die Rationalität an sich beschränkt ist, können weder von den Beteiligten, noch vom Ermittler alle möglichen Lösungen bedacht werden. In der Folge werden also Satisfizierungsziele anstatt optimaler Lösungen gewählt.
2. Beschränkung der Aufnahmefähigkeit: Ein Ermittler kann nicht alles verarbeiten, was die Beteiligten vorbringen – er kann insbesondere nichts aufnehmen, was die Beteiligten stillschweigend annehmen; hierzu zählt auch das implizite Wissen der Beteiligten.
3. Erhebungsprobleme: Rund 60% der Fragen, die bei der Anforderungsermittlung auftreten sollten, werden von den Anforderungsdokumenten nicht gestellt bzw. beantwortet. Dieser große Unterschied zwischen Anspruch und Wirklichkeit kann auf einige grundsätzliche Probleme heruntergebrochen werden. Hierzu zählen von den Ermittlern nicht gestellte Fragen, nicht betrachtete Möglichkeiten, die falsche Verwendung von Erhebungsmethoden oder die Verwendung falscher Erhebungsmethoden und nicht zuletzt mangelndes Verständnis seitens der Ermittler.
4. Falscher Transfer: Teilweise werden Lösungen für bereits bekannte, ähnliche Probleme auf das aktuelle Problem übertragen ohne die jeweils problemspezifischen Eigenheiten zu betrachten. Außerdem verfügen Designer ebenfalls über implizites Wissen und lassen dieses selten explizit in den Prozess einfließen; sie wenden es an um die Entwürfe anzufertigen, begründen dies jedoch nicht und gefährden so das gemeinsame Verständnis des Problems und der Lösung.

Leffingwell und Widrig beschäftigen sich ebenfalls mit den Problemen der Anforderungsermittlung [LW00]. Sie merken an, dass die Anforderungsermittlung und die Formulierung der Anforderungen in hinreichend genauer Art eigentlich Aufgabe des Kunden sei, in aller Regel könne der Kunde dies aber nicht (gut). Man könne also darauf warten, bis die Kunden der IT-Unternehmen professionelle Anforderungsermittlung erlernt haben oder man müsse selbst aktiv werden und lernen, wie man den Kunden bei der Anforderungsermittlung unterstützen kann. Leffingwell und Widrig klassifizieren die Probleme anhand von drei so genannten Syndromen:

3 Syndrome

- „Ja, aber“-Syndrom: Der Kunde wartet unter Umständen sehr lange auf das fertige Produkt; im Gegensatz zu materiellen Produkten gibt es keine (oder keine anschaulichen) Prototypen und die Entwickler beschreiben das Produkt während der Entwicklungsphase in den buntesten Farben. Dann sieht der Kunde das vermeintlich fertige Produkt zum ersten Mal – und es ist nicht das, was er sich vorgestellt hat.
- „Unentdeckte Ruinen“: Leffingwell und Widrig vergleichen Anforderungen mit unentdeckten Ruinen: Je mehr bei Ausgrabungen gefunden werden, desto mehr scheinen noch verborgen zu sein. Die Kernfrage wird folgende: „Wann soll ich aufhören zu suchen?“
- „Benutzer und Entwickler“-Syndrom: Die Benutzer zu verstehen ist das größte Problem der Ermittler, dieses Verständnis an die Entwickler weiterzugeben, ein nicht minder schweres. Benutzer und Entwickler kommen sprichwörtlich aus unterschiedlichen Welten – viele Fachbegriffe des einen sind unverständlich für den anderen. Hinzu kommt das unterschiedliche Domänenwissen der Parteien und die Betriebsblindheit² auf beiden Seiten.

Lösungsansätze für die beschriebenen Syndrome werden ebenfalls ausgearbeitet, sollen an dieser Stelle jedoch noch nicht behandelt werden.

Interviews sind die verbreitetste Art der Anforderungsermittlung, jedoch führen sie selten zu vollständigen Anforderungen [Rup02]. Obwohl gewissenhaft gearbeitet wird, werden keine guten Anforderungen erzeugt und die wirklichen Bedürfnisse des Kunden werden nicht entdeckt. Selten versteht der Ermittler die Domänensprache der übrigen Beteiligten auf Kundenseite und/oder spricht die Domänensprache. Diese Tatsache führt dazu, dass Anforderungen falsch verstanden und weitergegeben werden. Einen Fehler dieser Kategorie zu erkennen ist nur dann möglich, wenn alle verwendeten Begriffe klar definiert werden (siehe „Vokabular festlegen“ auf Seite 14).

2.2 Linguistische Analyse

Wir betrachten zunächst, wie Menschen ihre Umwelt wahrnehmen und wie sie ihr „Modell von der Welt“ an andere Menschen weitergeben können. Dann beschäftigen wir uns mit den aus diesen Prozessen entstehenden Problemen und den möglichen Defekten. Die neurolinguale Programmierung (NLP) bietet nicht nur Techniken, die die Kommunikation zwischen Menschen analysieren, sondern auch einfache Grundregeln, die eine zielgerichtete

²Die 3 Regeln des Fachbereichs:

- „Das machen wir schon immer so.“
- „Das haben wir noch nie so gemacht.“
- „Da könnte ja jeder kommen.“

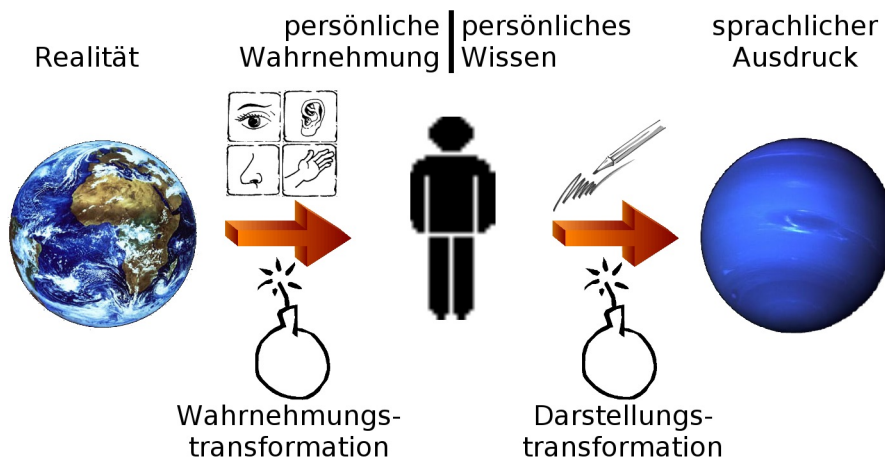


Abbildung 1: Wissenstransformation frei nach [Rup06, S. 141]

Kommunikation fördern (für weitere Informationen zu NLP siehe beispielsweise [Sta92]). Diese Techniken richten sich an natürlichsprachliche Kommunikation, weswegen sie überall dort eingesetzt werden können, wo natürliche Sprache verwendet wird. Chris Rupp und die SOPHISTen³ haben in den vergangenen Jahren ein Regelwerk entwickelt (die so genannten SOPHIST REgeln), welches die Techniken der NLP für die Informatikdomäne adaptiert und speziell auf die Anforderungsermittlung zugeschnitten ist [Rup06, S. 123ff]; es greift somit die verschiedenen Disziplinen der Linguistik, Informatik, Psychologie und die Psychotherapie auf und fügt sie zu einer praktischen Hilfe zusammen.

NLP und die
SOPHIST REgeln

Der Zusammenhang zwischen Linguistik und natürlichsprachlichen Anforderungen ist einleuchtend – doch wie kommt die Psychologie ins Spiel? Die Antwort auf diese Frage gibt die Psychologie selbst: Sie beschreibt anhand des Transformationsmodells wie Menschen ihre Umwelt wahrnehmen und ihr geistiges Modell mithilfe der Sprache mitteilen (siehe Abbildung 1). Als Einstieg in diese Materie soll ein Beispiel dienen:

Zwei Kellner berichten von einer Speisekarte – Sie als Ermittler können nur auf die Informationen zugreifen, die Ihnen von Ihren Gesprächspartnern mitgeteilt werden; Ihr Ziel ist es, eine Nachbildung der beschriebenen Speisekarte anzufertigen.

- Der erste berichtet von einer sehr schönen Karte, die auf dickem, beigen Papier gedruckt sei. Besonders beeindruckt sei er von den farbigen Abbildungen, die es fremdsprachigen Gästen ermögliche, die Speisen zu erkennen; eine Übersetzung der Speisekarte gebe es leider nicht.

³Die SOPHISTen sind eine Gruppe von Unternehmensberatern im IT-Bereich. Sie haben sich auf Software- und Systementwicklung spezialisiert und legen ein besonderes Augenmerk auf die Analyse von natürlichsprachlichen Anforderungen.

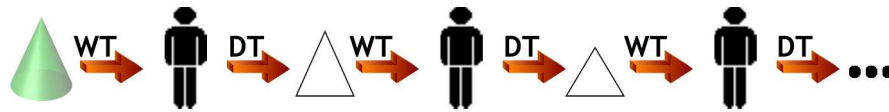


Abbildung 2: „Flüsterpost“

- Der zweite erklärt Ihnen, dass die Speisekarte „sinnvoll sortiert“ sei – zuerst die Vorspeisen und Suppen, dann die Hauptgerichte (wiederum sortiert nach der Fleischsorte), am Ende die Nachspeisen; danach stünden die alkoholfreien Getränke und zuletzt die Alkoholika.

Dieses einfache Beispiel verdeutlicht zweierlei: Beide Personen haben ein unterschiedliches Bild vom System (der Speisekarte) erworben und verinnerlicht; die Informationen, die sie weitergegeben haben, unterscheiden sich ebenfalls gravierend. Sie müssen also *beide Personen befragen*, um ein wirklichkeitsgetreues Modell anfertigen zu können. Um der Realität gerecht zu werden, müssen Sie noch dazu den Restaurantchef befragen – er könnte Ihnen mitteilen, dass vor allem die erste Seite (die Begrüßungsseite) von „immenser Wichtigkeit“ sei; dort sei darüber hinaus ein Photo von seiner Frau und Ihm abgedruckt.

Wahrnehmungs-
transformation

Darstellungs-
transformation

Die Begründung für das unterschiedliche Bild (das persönliche Wissen) liegt in der persönlichen Wahrnehmung der Personen (siehe Abbildung 1). Die hierbei stattfindende Wahrnehmungstransformation (WT) ist essentiell für das menschliche Gehirn; ohne selektive Wahrnehmung und Generalisierung wäre es von der Informationsflut überfordert. Das Interesse der Personen lenkt die Aufmerksamkeit auf bestimmte Teile der Realität – nur diese Teile werden bewusst wahrgenommen. Die WT ist irreversibel – wir können nicht beeinflussen, wie Menschen etwas aufgenommen haben. Das persönliche Wissen wird nach Chomsky auch als Tiefenstruktur bezeichnet. Diese Tiefenstruktur kann vom Menschen nicht direkt weitergegeben werden – das geistige Modell ist zu komplex, als dass wir es mithilfe der Sprache ausdrücken könnten; bevor es an eine andere Person übertragen werden kann, muss es auf die so genannte Oberflächenstruktur abgebildet werden. Dies ist die Darstellungstransformation (DT). Die wiedergegebene DT wird wiederum von einer Person aufgenommen, mit der WT in persönliches Wissen überführt und so weiter (siehe Abbildung 2).

Bei beiden beschriebenen Transformationen kann es zu Defekten kommen. Dass die WT fehlerbehaftet ist, liegt in der Natur der Sache und da sie irreversibel ist, können wir als Ermittler nur hoffen, dass sich die Defekte über der Menge der Befragten „ausmitteln“. Die DT folgt jedoch gewissen Regeln, die bei allen Menschen mehr oder minder gleich sind (ein Regelwerk ist die Transformationsgrammatik nach Chomsky; für nähere Informationen siehe [Cho73] und [Gar73]). Wenn wir als Ermittler nun diese Regeln kennen

und verstehen, können wir mögliche Defekte erkennen und direkt nachfragen, wenn sich ein Problem ergibt oder ergeben könnte. Das Ziel ist es, an die unter der ausgesprochenen Oberflächenstruktur gelegenen Tiefenstruktur zu gelangen. Hinterfragen sollte man jedoch nicht jede einzelne Aussage seines Interviewpartners, sondern nur die relevanten (siehe Abschnitt 3.3 auf Seite 12).

2.3 Sprachliche Effekte – Defekte

Sprachliche Effekte

Die möglichen Defekte werden von Bandler und Grinder in [GB89] in drei Kategorien eingeteilt und durch so genannte sprachliche Effekte (Defekte) ergänzt ([Rup06, S. 146ff]):

- **Tilgung:** „Tilgung ist das sprachliche Pendant zur Fokussierung der Wahrnehmung“ [Rup02]. Ohne die Tilgung würden wir mehr mitteilen, als unser Gegenüber aufnehmen könnte (man beachte die Analogie zu Horner und Atwoods zweiter fundamentalen Barriere auf Seite 5). Erkennbar wird die Tilgung am besten (aber nicht nur dort) bei *unvollständig spezifizierten Prozessworten*: So ist beispielsweise die Anforderung „Das System *meldet* Abstürze der Datenbank.“ schön aber nutzlos – es fehlen Angaben zur eigentlichen Meldung: Wann soll diese erfolgen, auf welchem Weg, an wen, mit welchen Informationen, wie lange (d.h. muss die Meldung quittiert werden)? Hinzu kommen *unvollständige Vergleiche oder Steigerungen*, *Modaloperatoren der Möglichkeit*, *Modaloperatoren der Notwendigkeit* und *implizite Annahmen*.
- **Generalisierung:** Menschen übertragen Erfahrungen auf ähnliche Situationen – ein notwendiger Bestandteil des Lernens. Werden bei der Spezifikation jedoch Spezialfälle vergessen oder weggelassen, führt dies im Echtbetrieb zu Störungen; genau dann, wenn diese Spezialfälle auftreten, verhält sich das System nicht so wie es sollte, sondern so wie es spezifiziert wurde. Generalisierungen lassen sich häufig an *Universalquantoren*, *unvollständig spezifizierten Bedingungen* (beispielsweise Partition des Intervalles [0 €; 100 €] in den Bereich „Preis kleiner 40 €“ und „teurer als 60 €“) und *Substantiven ohne Bezugsindex* (beispielsweise „die Daten“ oder „das System“) erkennen.
- **Verzerrung:** Sie führt dazu, dass Menschen ihr Wissen nicht ständig neu organisieren müssen; passt eine neue Information nicht zu bereits bekanntem, wird sie einfach verzerrt aufgenommen (d.h. Details angepasst oder „zurecht geschustert“). Verzerrungen treten nicht nur bei der WT auf, sondern auch bei der DT; erkennbar sind sie vor allem an *Nominalisierungen* und *Funktionsverbgefügen*. Nominalisierungen führen dazu, dass (möglicherweise komplexe) Vorgänge in ein Wort

gepackt werden; beispielsweise „Überweisung“ bei einer Bank, „Neustart“ eines Computersystems und so weiter; Funktionsverbgefüge bestehen aus einem Funktionsverb und einem sinngabenden Substantiv (beispielsweise „in Auftrag geben“) und können häufig durch ein Vollverb ersetzt werden (beispielsweise „beauftragen“).

Rupp sieht unter allen genannten Indizien die Nominalisierung, unvollständig spezifizierte Prozesswörter, Substantive ohne Bezugsindex, unvollständig spezifizierte Bedingungen, impliziten Annahmen und Modaloperatoren der Notwendigkeit als die am häufigsten auftretenden Indizien für Defekte in natürlichsprachlichen Anforderungen [Rup06, S. 147]. Lege man ein besonderes Augenmerk auf diese Kriterien, könne man schon gute Anforderungen erzeugen.

3 Mögliche Lösungen

3.1 Domänensprache verwenden

„*People can't share knowledge if they don't speak a common language.*“ [DP98, S. 98]

Gemeinsame Sprache

Man sollte das Problem verstanden haben, bevor man anfängt, die Lösung zu modellieren [LW00]. Da die Problemanalyse viel Kommunikation zwischen den Nutzern und dem Ermittler erfordert, muss die Sprache der Nutzer verwendet werden [RR06] – sonst reden Nutzer und Ermittler aneinander vorbei. Bei der Analyse geht es immer um einen Wissenstransfer zwischen Nutzer und Ermittler; dieser Transfer kann nur erfolgen, wenn die am Gespräch Beteiligten die gleiche Domänensprache sprechen bzw. verstehen. Hat der Ermittler den Nutzer verstanden, sollte er sein Verständnis des Sachverhaltes wieder ausdrücken um sicherzugehen, dass keine Verständnisprobleme entstanden sind (Feedback-Schleife). Nachdem das Problem erkannt wurde, sollte man versuchen, eine bereits bestehende Lösung (ein altes Computersystem oder den „Papierweg“) des Nutzers zu verstehen. Erst am Ende steht dann die Modellierung der Lösung.

3.2 Unterstützung durch Software und verteilte Prozesse

SW-Unterstützung

Wichtig ist es, die Aufgaben der Anforderungsermittlung vorzubereiten. Das bedeutet, dass der Ermittler nicht nur Personen für die Erhebung auswählen muss, sondern auch die anzuwendenden Techniken (Brainstorming, Fragebögen, Interviews etc.) [RR06]. Darüber hinaus sollten die einzelnen Dokumente für die Techniken vorbereitet werden. Dass Werkzeuge die Arbeit des Ermittlers unterstützen können, erkannte man bereits in den 70er Jahren. Damals ging man davon aus, dass eine Formalisierung der Anforderungen zu einer höheren Qualität führen würde und führte Systeme wie PSL/PSA

oder SREM ein, die nach der Erfassung der Anforderungen in einer speziellen Eingabesprache umfangreiche Auswertungen und Prüfungen zuließen. Diese waren jedoch kompliziert und in der Regel nicht von allen Beteiligten zu bedienen [SSTW80]. Insbesondere gaben die damals existierenden Systeme keine oder nur wenig methodische Hilfestellungen für die tatsächliche Ermittlung der Anforderungen, sondern nur für deren Verwaltung (Sicherstellung von Eindeutigkeit, Vermeidung von Inkonsistenzen etc.).

Boehm entwickelte an der University of Southern California (USC) das so genannte WinWin-System [BGB01]; es vereint methodische Ansätze und ein Groupware System (USC Easy WinWin), das die Beteiligten nicht nur bei der Verwaltung der Anforderungen, sondern auch bei der Erhebung unterstützt. Sein Ziel ist es, die Anforderungen der verschiedenen Beteiligten zusammenzuführen und zu harmonisieren. Boehm erkannte, dass die meisten Entwickler nichts von Gruppendynamik und Psychologie verstehen und entwickelte das Groupware-System um den Fokus weg von Konflikten wieder zurück auf die Anforderungen zu lenken. Die Beteiligten geben ihre Anforderungen in das System ein und diskutieren über die einzelnen Punkte nur dann, wenn nicht alle Beteiligten der Anforderung zustimmen. Einigen sie sich auf einen Sachverhalt, werden Argumente und deren Interpretationen sowie die Anforderung als *WinCondition* festgehalten. Der Prozess läuft so lange, bis keine Probleme (*Issues*) mehr registriert sind – Boehm stellte fest [BE98], dass die meisten Anforderungen keiner Diskussion bedürfen, sondern ohne weitere Diskussion festgehalten werden können. WinWin erhöhe nicht nur die Produktivität der Gruppe indem es Reibungspunkte vermindere, sondern führe auch dazu, dass einmal vereinbarte Anforderungen stabiler seien (schließlich werden alle Entscheidungen und Entwicklungsschritte dokumentiert), so Boehm. Auch dieses System verhindert nicht, dass die oben beschriebenen linguistischen Probleme auftreten – durch die Konzentration auf die Anforderungen und die Möglichkeit zeitversetzt und online auf WinWin zuzugreifen, unterstützt es verteilte Prozesse sehr stark [BE98].

Dass dezentralisierte Prozesse die Qualität von Ermittlungsprozessen erhöhen können, ist ein Ergebnis, zu dem auch Damian kommt [Dam01]: Sie führte eine Studie durch, in der die psychologischen Effekte betrachtet wurden, die in verteilten Prozessen entstehen. Da Anforderungen immer „nur“ ein Kompromiss zwischen den Nutzern seien, müsse man sich darauf konzentrieren die Kompromissbereitschaft und die Objektivität der Nutzer zu erhöhen. Verglichen wurde das „übliche“ Szenario mit Moderator, Ermittler und zwei Nutzern an einem Tisch (Face to Face – F2F) mit einigen verteilten Szenarien (Kombinationen, bei der die Gruppe auf zwei Standorte aufgeteilt und mit einer Videoschaltung verbunden wurde). Zwar unterscheiden sich die Ergebnisse der unterschiedlichen verteilten Anordnungen voneinander, jedoch ergibt sich aus allen die selbe Aussage: Die verteilte Situation schnitt in allen Fällen besser ab, da die Teilnehmer aufgrund der Videoschaltung zu langsamerem Sprechen und höherer Disziplin gezwungen waren. War entwe-

Dezentrale Prozesse

der der Moderator oder der Ermittler isoliert, empfand er die Distanz als positiv; ebenso empfanden die Nutzer die Distanz zum Ermittler als Positiv. Alle Beteiligten sagten aus, dass das verteilte Szenario durch die Distanzierung eine erhöhte Objektivität mit sich bringe und sich die erhöhte Disziplin positiv auf die Verhandlungen auswirke. Man könnte aus diesen Ergebnissen ableiten, dass es nutzenbringend sein kann, die Nutzer voneinander zu trennen um die Rivalitäten zu vermindern und den Ermittler zu distanzieren um eine möglichst hohe Objektivität der Beteiligten zu fördern.

Wie Davenport in [DP98, S. 96ff] kommt auch Damian zu dem Ergebnis, dass eine sinnvolle Verhandlung nur dann möglich ist, wenn ein gewisses Grundvertrauen zwischen den Beteiligten etabliert ist. Um dies zu erreichen ist es unabdingbar, dass sich die Beteiligten zumindest ein mal treffen – dieses Treffen kann auch ein informelles Treffen sein und muss nicht mit dem selben Projekt zusammenhängen; es geht einzig und allein darum, dass sich die Personen kennen lernen.

3.3 NLP und das REgelwerk der SOPHISTen

Kommen wir nun zu den möglichen Lösungen der sprachlichen Probleme. Im zweiten Abschnitt haben wir erfahren, wie Menschen geistige Modelle bauen und wie sie diese verfälscht werden. Nun wollen wir uns mit den Regeln beschäftigen, die dies verhindern oder eindämmen sollen.

Im oberen Abschnitt haben wir bereits gesehen, welche sprachlichen Anzeichen es für die drei Defektgruppen gibt. Aus diesen Anzeichen lassen sich einfache Regeln ableiten, die bereits während der Interviews angewendet werden können – der Ermittler überprüft die Aussagen des Interviewpartners in Echtzeit und kann so auf mögliche Defekte eingehen; wann immer eine der Regeln „anschlägt“ kann er nachfragen, muss es aber nicht. Auf keinen Fall sollte der Ermittler auf jeder Aussage seines Gesprächspartners „herumreiten“, sondern sich ggf. Fragen notieren und zu einem anderen Zeitpunkt noch einmal auf dieses Thema genauer eingehen. Einige der Regeln der SOPHISTen lauten [GR03]:

- Formuliere jede Aussage im Aktiv (dies erzwingt, dass der Akteur eindeutig benannt wird).
- Finde alle Prozesse, die als Substantive angegeben sind.
- Entdecke nicht vollständig spezifizierte Prozessworte.
- Vervollständige unvollständige Vergleiche.
- Prüfe alle Modaloperatoren auf Klarheit (warum gilt etwas bzw. muss etwas getan werden?).
- Benenne die impliziten Annahmen.

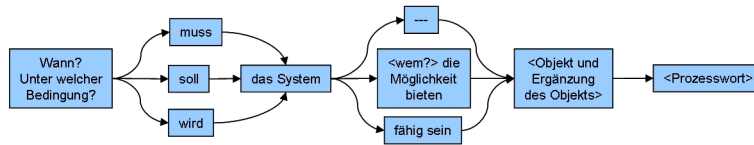


Abbildung 3: Syntaktische Schablone für Anforderungen nach [Rup06, S. 234]

- Finde und hinterfrage Allquantoren.

Diese Regeln können selbstverständlich auch auf Transkriptionen von Gesprächen oder Antworten auf Fragenbögen angewendet werden. Werden diese Regeln konsequent angewandt, kommt man zu semi-formalen Anforderungen – sie liegen immer noch in natürlicher Sprache vor, genügen jedoch (beinahe) unseren Qualitätsanforderungen.

Diese semi-formalen Anforderungen können noch weiter verbessert werden. Da es wesentlich einfacher ist, sofort gute Anforderungen zu schreiben, als beinahe gute zu verbessern, wollen wir uns an dieser Stelle noch damit befassen. Wir erstellen uns Schablonen für Anforderungen, nach denen wir alle Anforderungen aufbauen. Dies

Schablonenbasierter
Ansatz

- spart Zeit,
- ergibt einheitliche Anforderungen und
- ergibt eindeutige und testbare Anforderungen.

Mit Schablonen können wir sicherstellen, dass häufige Fehler vermieden werden und geben den Ermittlern und den anderen Beteiligten eine Anleitung an die Hand, mit der sie mit einfachen Mitteln gute Anforderungen erstellen können [Rup06, S. 228f]. In Abbildung 3 ist eine Schablone abgedruckt, die alle notwendigen Informationen aufnehmen kann. Hält sich der Ermittler oder der Nutzer an die Schablone, muss er lediglich die leeren Bereiche auffüllen und kommt zu einer guten Anforderung. Der große Vorteil der schablonenbasierten Anforderungen ist es, dass sie einheitlich aussehen; alle Anforderungen sind nach dem selben Schema gestrickt. Dies ist aber auch ein Nachteil, denn wenn wir sicherstellen wollen, dass die Anforderungen hinterher auch von jemandem gelesen werden, müssen wir das Anforderungsdokument ebenfalls strukturieren; am besten eignet sich eine Gliederung nach Funktionsgruppen, sodass ein Nutzer direkt finden kann, was ihn interessiert bzw. was er überprüfen muss.

Mit sechs Schritten kommen wir nach Rupp [Rup06, S. 235f] zu einer guten Anforderung (in Klammern steht das jeweilige Element der Schablone in Abbildung 3):

6 Schritte

1. Zuerst bestimmen wir den Prozess (Prozesswort),

2. anschließend wählen wir die Systemaktivität (Fähigkeit, Bereitstellen einer Funktion, oder —),
3. dann legen wir die rechtliche Verbindlichkeit (muss, soll, wird) fest,
4. anschließend fügen wir ein Objekt und ggf. Ergänzungen hinzu und stellen (Objekt und Ergänzung)
5. eine Bedingung (Wann? Unter welcher Bedingung?) voran.
6. Zuletzt prüfen wir die erstellte Anforderung mit dem weiter oben besprochenen Regelwerk.

Die Schablonenmethode sollte in der Praxis nur eingesetzt werden, wenn sie von den Beteiligten akzeptiert wird – Wenn sich die Nutzer zu sehr eingeschränkt fühlen, formulieren sie vielleicht nur ein Mindestmaß an Anforderungen, weil ihnen die Schablonen lästig sind. Bessere Erfahrungen hat man damit gemacht, die sechs Schritte als Methode zu schulen und die Schablone als Hilfsmittel darzustellen. Selbstverständlich lässt sich die Schablone einfach in ein Werkzeug integrieren, das die Nutzer beim Verfassen unterstützen kann.

Vokabular festlegen

Einen großen Schritt hin zu einer guten Anforderung haben wir bereits mit den Schablonen und dem Regelwerk getan; das große Problem der Mehrdeutigkeit der Sprache haben wir mit unseren Schablonen noch nicht behoben. Zu diesem Zweck definieren wir zur syntaktischen Schablone zusätzlich semantische Definitionen und legen die Formulierungen für logische Operatoren fest. Hierfür werden zunächst alle Wörter, die verwendet werden sollen, in ein Verzeichnis aufgenommen. In diesem Verzeichnis wird auch ihre Bedeutung festgehalten. Um dies zu vereinfachen sollen keine Synonyme in das Verzeichnis aufgenommen werden – wenn „einfügen“ und „eingeben“ synonym verwendet werden können, ist einer der Begriffe überflüssig. Gibt es jedoch einen Unterschied zwischen den Begriffen, soll klar vermerkt werden *worin* dieser Unterschied besteht und wann welcher Begriff verwendet werden soll. Dieses Vorgehen führt zu einer „common language“, die allen Beteiligten zugute kommt [ZJ97]. Dies ist ein schwieriger Prozess, der nicht vernachlässigt werden darf – er ist essentiell für die Bildung eines gemeinsamen Verständnisses und wie wir bereits wissen: ohne dies ist ein Informationsaustausch stark eingeschränkt.

Logische Operatoren definieren

Eine klare Definition muss auch für logische Operatoren angegeben werden – damit alle diese verstehen können, muss zwingend ein Beispiel für den beschriebenen Operator angegeben werden. So werden AND, OR, XOR usw. festgelegt und für alle zugänglich gemacht. Verwendet man nun die definierten Operatoren anstelle von Sprachkonstrukten, werden Bedingungen klar ausgedrückt.

Bei der Anforderung „An jedem Dienstag um 9:00 Uhr oder an jedem Freitag um 8:00 Uhr und unter der Bedingung, dass [...], soll xyz“ wird nicht

klar, ob die einschränkende Bedingung nur freitags oder an beiden Tagen gelten soll. Besser wäre eine Formulierung mit den definierten Operatoren: „(An jedem Dienstag XOR an jedem Montag) UND unter der Bedingung, dass [...], soll xyz“.

Bei dieser Formulierung muss man jedoch darauf achten, dass die Anforderungen noch lesbar bleiben. Wir haben nichts gewonnen, wenn wir wieder vollständig formale, aber dem Nutzer unverständliche Anforderungen formuliert haben.

Zum Ende dieses Abschnittes soll kurz auf einige Möglichkeiten hingewiesen werden, wie bereits gemachte Fehler aufgespürt und eliminiert werden können. Mit der NLP und den daraus abgeleiteten Regeln können wir Fehler noch bei bereits „fertigen“ Anforderungen finden – dann müssen wir noch einmal beim Nutzer nachfragen. Robertson beschreibt eine Feedback-Schleife [RR06], mit der Fehler entdeckt werden können: Der Ermittler formuliert zunächst die Anforderung und gibt diese dann an den Nutzer zurück; dieser bestätigt die Anforderung oder korrigiert sie. Dieser Kreislauf endet, wenn alle Anforderungen bestätigt wurden. Hierbei lässt sich darüber streiten, ob es sich um eine nachträgliche Korrektur handelt, oder ob Robertson einen iterativen Prozess beschreibt. Eine interessante Möglichkeit der nachträglichen Überprüfung des Anforderungsverständnisses besteht im Test Driven Development (TDD): Zeigt und erklärt der Entwickler/Tester die durchgeführten Tests, so können Defekte aufgrund der WT beim Entwickler aufgedeckt werden – da die Tests ohnehin geschrieben werden müssen, sollten sie unbedingt bei komplexen Anforderungen zu Rate gezogen werden um eine Abstimmung zwischen Ermittler und Entwickler zu erreichen.

Feedback-Schleife

TDD

4 Schlussfolgerung/Fazit

Es gibt viele Probleme bei der Anforderungsermittlung, deren Ursachen sich mithilfe der Psychologie nachvollziehen lassen. Mithilfe einfacher Gruppentechniken und Groupware kann die Zusammenarbeit im Team verbessert werden. Wird nach nach bestimmten Kriterien Ausschau gehalten, können Probleme der natürlichen Sprache aufgespürt und behoben werden. Mit wenigen Hilfsmitteln kann vielen Problemen vorgebeugt werden; werden einfache Grundregeln befolgt, können darüber hinaus Unklarheiten bereits in der frühen Anforderungsphase beseitigt werden. Nicht zuletzt sorgt eine offene Unternehmenskultur bei den Beteiligten dafür, dass Kommunikation reibungsloser vonstatten gehen kann – es müssen allerdings Anreize geschaffen werden, damit alle Beteiligten ihre wahren Bedürfnisse offenbaren [Dut06].

Abschließend kann klargestellt werden, dass es keinen allgemein gültigen Anforderungserhebungsprozess gibt; dieser muss für jede Organisation – vielleicht sogar für jedes Projekt individuell ausgewählt werden [DS99]. Jedes Projekt ist einzigartig und dies gilt ebenso für alle am Projekt beteiligten.

Literatur

- [BE98] BOEHM, Barry ; EGYED, Alexander: Software Requirements Negotiation: Some Lessons Learned. In: *20th International Conference on Software Engineering*, 1998 11
- [BGB01] BOEHM, Barry ; GRÜNBACHER, Paul ; BRIGGS, Robert O.: Developing Groupware for Requirements Negotiation: Lessons Learned. In: *Software, IEEE* (2001), May/June, S. 46–55 3, 11
- [BI99] BOEHM, Barry ; IN, Hoh: Conflict Analysis and Negotiation Aids for Cost-Quality Requirements. (1999) 3
- [Cho73] CHOMSKY, Noam: *Strukturen der Syntax*. Mouton, 1973 8
- [Dam01] DAMIAN, Daniela E.: An Empirical Study of Requirements Engineering in Distributed Software Projects: Is Distance Negotiation More Effective? In: *APSEC*, IEEE Computer Society, 2001. – ISBN 0–7695–1408–1, S. 149– 11
- [DP98] DAVENPORT, Thomas H. ; PRUSAK, Laurence: *Working knowledge*. Harvard Business School, 1998. – ISBN 0–87584–655–6 10, 12
- [DS99] DAWSON, Linda ; SWATMAN, Paul: The use of object-oriented models in requirements engineering: a field study. In: *ICIS '99: Proceeding of the 20th international conference on Information Systems*. Atlanta, GA, USA : Association for Information Systems, 1999. – ISBN ICIS1999–X, S. 260–273 4, 15
- [Dut06] DUTOIT, Allen H.: *Rationale Management in Software Engineering*. 1. Auflage. Springer, 2006. – ISBN 3–540–30997–7, 978–3–540–30997–0 4, 5, 15
- [Gar73] GARDNER, Thomas: *Hauptströmungen der modernen Linguistik*. Vandenhoeck and Ruprecht, 1973. – ISBN 3–525–33347–1 8
- [GB89] GRINDER, John ; BANDLER, Richard: *Kommunikation und Veränderung*. 4. Aufl. Junfermann, 1989. – ISBN 3–87387–187–4 9
- [GR03] GOETZ, Rolf ; RUPP, Chris: Psychotherapy for system requirements. In: *Cognitive Informatics, 2003. Proceedings. The Second IEEE International Conference on* IEEE Computer Society, 2003, S. 75–80 12
- [Hof00] HOFMANN, Hubert F.: *Requirements Engineering*. 1. Auflage. Dt. Univ.-Verl., 2000. – ISBN 3–8244–7215–5 1

- [HW05] HOOD, Colin ; WIEBEL, Rupert: *Optimieren von Requirements Management & Engineering*. 1. Auflage. Springer, 2005. – ISBN 3-540-21178-0 [2](#)
- [LW00] LEFFINGWELL, Dean ; WIDRIG, Don: *Managing software requirements*. 1. Auflage. Addison-Wesley, 2000. – ISBN 0-201-61593-2 [5](#), [10](#)
- [NT97] NONAKA, Ikujiro ; TAKEUCHI, Hirotaka: *Die Organisation des Wissens*. 1. Auflage. Campus-Verl., 1997. – ISBN 3-593-35643-0 [4](#)
- [RR06] ROBERTSON, Suzanne ; ROBERTSON, James: *Mastering the requirements process*. 2. ed. Addison-Wesley, 2006. – ISBN 0-321-41949-9 [10](#), [15](#)
- [Rup02] RUPP, Chris: Requirements and psychology. In: *Software, IEEE* 19 (2002), May-June, Nr. 3, S. 16–18 [4](#), [6](#), [9](#)
- [Rup06] RUPP, Chris: *Requirements-Engineering und -Management. Professionelle, iterative Anforderungsanalyse für die Praxis*. 4., aktualisierte und erw. Aufl. Hanser Fachbuchverlag, Oktober 2006. – ISBN 3-446-40509-7, 978-3-446-40509-7 [7](#), [9](#), [10](#), [13](#)
- [SSTW80] SMITH, Gregory L. ; STEPHENS, Sharon A. ; TRIPP, Leonard L. ; WARREN, Wayne L.: Incorporating usability into requirements engineering tools. In: *ACM '80: Proceedings of the ACM 1980 annual conference*. New York, NY, USA : ACM Press, 1980. – ISBN 0-89791-028-1, S. 355–368 [11](#)
- [Sta92] STAHL, Thies: *Neurolinguistisches Programmieren (NLP)*. 5. Auflage. PAL, 1992. – ISBN 3-923614-43-8, 1992 [7](#)
- [Sut02] SUTCLIFFE, Alistair: *User centred requirements engineering*. 1. publ. Springer, 2002. – ISBN 1-85233-517-3 [1](#), [3](#)
- [ZJ97] ZAVE, Pamela ; JACKSON, Michael: Four dark corners of requirements engineering. In: *ACM Trans. Softw. Eng. Methodol.* 6 (1997), Nr. 1, S. 1–30. – ISSN 1049-331X [14](#)