Requirements Engineering - der Einsatz einer natürlichsprachlichen Methode bei der Ermittlung und Qualitätsprüfung von Anforderungen

Chris Rupp

chris.rupp@sophist.de

SOPHIST GROUP

Vordere Cramergasse 11-13

90478 Nürnberg

Die Autorin



Chris Rupp (chris.rupp@sophist.de) legte das Fundament des NLP-basierten Requirements Engineering. Darauf basierend erarbeitete und veröffentlichte sie pattern- orientierte Ansätze. Sie ist Autorin zahl- reicher international verlegter Publikationen und als Trainerin und Beraterin für Kunden im Einsatz. Chris ist Gründerin und geschäfts- führende Gesellschafterin der SOPHIST GmbH und SOPHIST Technologies GmbH. Seit 8 Jahren engagiert sie sich in der Analyse sicherheitskritischer technischer Großprojekte internationaler Kunden. Ihre Arbeitsmethodik umfasst unter anderem natürlichsprachliche und objektorientierte Methoden sowie Bereiche der Organisationspsychologie und des Neuro-Linguistischen Programmierens (NLP). Private Vorlieben (zu viele für zu wenig Zeit): Menschen, Philosophie, Rotwein, Reisen und die Suche nach dem Sinn des Lebens.

Abstract:
Um Fehler in der Analysephase der Software-Entwicklung zu vermeiden und Auftraggeber und Anwender stärker in die Analyse einzubinden, sieht die Vorgehensweise Object Engineering unter anderem den Einsatz **sprachlicher Methoden zur Ermittlung und Qualitätsprüfung von Anforderungen** vor. Hierbei handelt es sich nicht um einen weiteren Ansatz, Anforderungen formal zu beschreiben, sondern vielmehr um eine Methode, die direkt mit den in natürlicher Sprache verfassten Anforderungen arbeitet. Ziel der Methode ist die Entwicklung qualitativ hochwertiger, juristisch verbindlicher natürlichsprachlicher Anforderungen als Grundlage für die Systementwicklung.

Forschungsergebnisse anderer Wissenschaften (Linguistik und Psychologie) wurden von der SOPHIST Group auf die Informatik übertragen und ergeben dort ein handhabbares Werkzeug, um mehrdeutige, unvollständige und widersprüchliche Anforderungen aufzudecken oder gar nicht erst entstehen zu lassen. In diesem Papier sollen auszugsweise die sprachlichen Phänomene untersucht werden, die zu den genannten Defekten in Anforderungen führen.

Schlüsselworte: Anforderungen, Anforderungsanalyse, Requirements Engineering, natürlich sprachliche Methode, Linguistische Methoden, Analyse, Pflichtenheft, Neurolinguistisches Programmieren (NLP), Metamodell der Sprache.

Requirements Engineering - der Einsatz einer natürlichsprachlichen Methode bei der Ermittlung und Qualitätsprüfung von Anforderungen

Die Analyse als erste Phase der Softwareentwicklung entscheidet maßgeblich über den Erfolg oder Mißerfolg eines Projektes. Um Auftraggeber und Anwender stärker in die Analyse einzubinden, häufig natürlichsprachliche Methoden zur Ermittlung und Qualitätssicherung von Anforderungen eingesetzt.

Bei natürlichsprachlichen Methoden handelt es sich nicht um einen weiteren Ansatz, Anforderungen formal zu beschreiben, sondern vielmehr um Methoden, die direkt mit den in natürlicher Sprache verfassten Anforderungen arbeiten. Ziel der Methoden ist die Entwicklung qualitativ hochwertiger, juristisch verbindlicher natürlichsprachlicher Anforderungen. Der Artikel stellt die Erfahrungen mit natürlichsprachlichen Methoden dar.

Motivation für eine erfolgreiche Systemanalyse

Verschiedene Untersuchungen zeigen, dass die Mehrzahl aller Fehler in Softwareprodukten in der Analysephase entstehen. Fehler, die in einer frühen Phase der Softwareentwicklung gemacht und erst in späteren Phasen behoben werden (wenn es dafür eigentlich schon zu spät ist), sorgen für einen Summationseffekt, bei dem diese sich in spätere Phasen fortpflanzen und potenzieren. Tatsächlich lassen sich ca. 65% der schwerwiegenden Fehler in Programmen auf Unzulänglichkeiten in der Analysephase zurückführen.

Um das Problem der Explosion der Kosten sowie der Entwicklungszeit in den Griff zu bekommen, müssen die Ursachen beseitigt werden, die in der Analysephase liegen. Hier greifen Verbesserungen wesentlich effektiver als im Design oder gar während der Implementierung. Je früher man einen Fehler findet und behebt, desto weniger Folgefehler werden verursacht und desto eher können die relevanten Entwicklungsparameter *Kosten* und *Zeit* im Rahmen gehalten werden

Requirements Engineering – oder wie dokumentiere ich Anforderungen?

Requirements Engineering (im Folgenden kurz *RE*) befasst sich mit dem Auffinden, der Dokumentation und der Überprüfung von Anforderungen an ein Softwareprodukt.

In der Projektrealität finden sich klassisch etwa die im Folgenden beschriebenen Ausprägungen beim Vorgehen:

Ein nicht besonders ausführliches Pflichtenheft, das zu großen Teilen aus vagen Formulierungen von Anforderungen in natürlicher Sprache (Prosa) besteht, wird dem Analytiker vom Auftraggeber vorgelegt. Der Analytiker überführt dieses Pflichtenheft soweit möglich in eine (halb-)formale Darstellung, zum Beispiel ein OO-Modell. Anhand dieser Darstellung forscht der Analytiker weiter und ergänzt es um Aspekte, die er z.B. durch Befragung der Anwender ermittelt hat. Häufig fließen in das Modell aber auch die Interpretationen und Phantasie des Modellierungsteams mit ein, das von der Realität des Endanwenders weit entfernt ist. Das ursprüngliche Pflichtenheft verliert jegliche Relevanz, da alle Anforderungen im Modell stecken und das Pflichtenheft nicht „nachgeführt" wird.

Dieses Vorgehen (so wie es in der Projektrealität abläuft) liefert keine klare Darstellung. Das Ergebnis einer Analyse sollte für alle Beteiligten - also auch den Endanwender, den Vertriebsleiter, den Produktmanager, die Wartungs- und Serviceteams - verständlich sein, da alle diese Personengruppen Anforderungen an ein System stellen und diese formulieren sollten. Eine verständliche Darstellung ermöglicht dem Anwender zu verfolgen, was der Analytiker modelliert. Der Anwender kann sofort „gegensteuern", wenn er bemerkt, dass das Modell Defizite aufweist.

Aus den dargelegten Gründen kommt man nicht umhin, den in natürlicher Sprache formulierten Anforderungen (Prosaanforderungen) einen größeren Stellenwert als bisher einzuräumen. Das heißt nicht, dass (semi-)formale Darstellungen von Anforderungen, wie z. B. Objektmodelle, als nicht adäquat betrachtet werden. Von uns verwendete Vorgehensmodelle sehen meist die Übersetzung der Prosaanforderungen in ein Objektmodell und ein Simulationsmodell vor. Vielmehr soll die Darstellung in natürlicher Sprache und das Arbeiten mit den natürlichsprachlichen Anforderungen integraler Bestandteil des Analyseprozesses sein. Die Darstellung des gesamten Vorgehensmodells würde den Rahmen des Berichtes sprengen. Aus diesem Grund geht der Artikel nur auf die unbekannteste, aber erfolgskritischste Komponente ein, die sich mit der Erstellung der natürlichsprachlichen Spezifikation befasst.

Die Darstellung von Anforderungen

Die Anforderungen sind nicht nur das Kommunikationsmittel zwischen allen Projektbeteiligten, sondern auch die Basis für das Design und die Implementierung. Aus diesem Grund sollte der Satz von Anforderungen

- eindeutig
- vollständig und widerspruchsfrei
- umsetzbar und testbar

sein. Diese Punkte bilden die Forderung nach formaler Definiertheit und rechtlicher Verbindlichkeit. Zusätzlich sollen Anforderungen auch

- verständlich

sein. Dadurch erhöht sich die Akzeptanz beim Anwender wesentlich.

Sollen die Anforderungen an ein System nun nicht-formal, d.h. in natürlicher Sprache, dargestellt werden, so gibt es Probleme, die die natürliche Sprache mit sich bringt. Äußerungen in Prosa sind oft in weiten Grenzen interpretierbar. Die eben genannten Kriterien Eindeutigkeit, Vollständigkeit, Widerspruchsfreiheit, Umsetzbarkeit und Testbarkeit lassen sich (scheinbar!) nicht erreichen. Darüber hinaus eignet sich Prosa nicht unbedingt, um komplexe Zusammenhänge strukturiert, d.h. übersichtlich, darzustellen.

In diesem Kontext ist ein Regelwerk zum Verfassen und Überprüfen von Prosaanforderungen zweckdienlich, das bei der Behebung oder zumindest Einschränkung der Unzulänglichkeiten natürlicher Sprache behilflich ist. Ziel eines Analyseprozesses auf der Basis von Prosaanforderungen muss sein, dass die genannten formalen Kriterien aber auch eine Verständlichkeit für alle Beteiligten erreichen. Solch ein Regelwerk existiert und wurde in der Praxis beispielsweise bei der Deutschen Flugsicherung, der Deutschen Post, Swisscontrol, Eurocontrol und in vielen anderen Projekten erfolgreich angewandt.

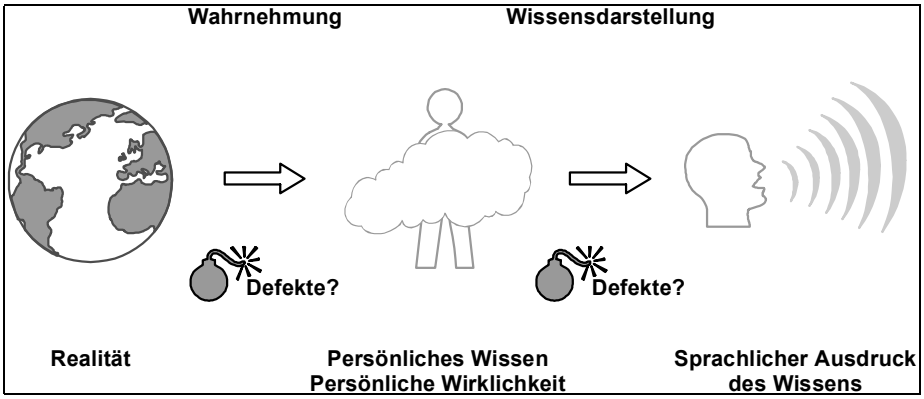


Abbildung 1: Potentielle Fehlerquellen liegen in der Wahrnehmung der Realität und ihrer Darstellung.

Methodisches Vorgehen bei der Analyse von Prosaanforderungen

Ziel der natürlichsprachlichen Methode ist es, Formulierungen aufzudecken, die durch subjektive Erfahrungen oder Formulierungen des Verfassers geprägt sind, und diese durch klare Formulierungen zu ersetzen. Ein methodisches Vorgehen ist an dieser Stelle nur deshalb möglich, weil die natürliche Sprache unabhängig von ihrem Inhalt ein System besitzt. Dieses System oder Modell der Sprache wurde durch Wissenschaftler aus dem Bereich der Sprachwissenschaft (Linguistik) untersucht und definiert.
Urvater der Theorie eines systematischen Aufbaus der Sprache ist Noam Chomsky, der Begründer der generativen Transformationslinguistik. Nach seiner Theorie sollte es möglich sein, eine beliebige sprachliche Äußerung (mündlich oder schriftlich) durch einen definierten Satz von Grammatikregeln zu bilden.

Aufbauend auf den Werken der Linguistik wurde im Bereich der Psychotherapie ein Modell menschlicher Kommunikation und Ausdrucksweise erstellt und verfeinert. Zusätzlich stellte die Psychologie ein systematisches Regelwerk auf, nach dem man durch genaue Betrachtung eines Satzes herausfinden kann, ob Transformationen bei der Bildung des Satzes im Spiel waren. Dabei zielt die Untersuchung eines Satzes immer darauf ab zu ergründen, was ein Mensch *meint*, wenn er etwas *sagt*. Die im Folgenden aufgeführten Sprachregeln entstammen größtenteils dem „Metamodell der Sprache“ von R. Bandler und J. Grinder (vgl. [Ban94]), den beiden Erfindern des Therapieansatzes „Neuro-Linguistisches Programmieren“ (NLP). Der genannte Ansatz sieht zwei potenzielle Stellen, an denen Fehler in Aussagen und damit auch in Anforderungen entstehen können, wie Abbildung 1 verdeutlicht:

- bei der Wahrnehmung eines Sachverhaltes (Wahrnehmung der Realität) und
- bei der sprachlichen Darstellung eines Sachverhalten (des Wissens)

Unterschiedliche Modelle der Sprache (z.B. aus der Linguistik und Therapie) wurden von der Sophist GROUP auf den Bereich der Informatik transformiert. Das entstandene Regelwerk stellt ein einfach zu handhabendes Vorgehen für die Erstellung und Qualitätskontrolle von Prosaanforderungen dar.

Mit Hilfe dieses Regelwerks lassen sich bei der Überprüfung von natürlichsprachlich formulierten Anforderungen auf definierte, systematische Art und Weise mehrdeutige, unvollständige und widersprüchliche Aussagen in Anforderungsdokumenten auffinden. Aber auch präventiv lässt sich die Kenntnis der Prozesse beim Verfassen von Anforderungen einsetzen, um die genannten Defekte zu vermeiden.

Die natürlichsprachliche Methode zur Ermittlung und Qualitätsprüfung von Anforderungen

Im Folgenden werden die wichtigsten Grundsätze einer natürlichsprachlichen Methode erläutert. In Abbildung 2 sind die am häufigsten zu beobachtenden sprachlichen Defekte in Anforderungsspezifikationen dargestellt. Im Rahmen des Artikels werden einige repräsentative Vertreter näher erläutert.

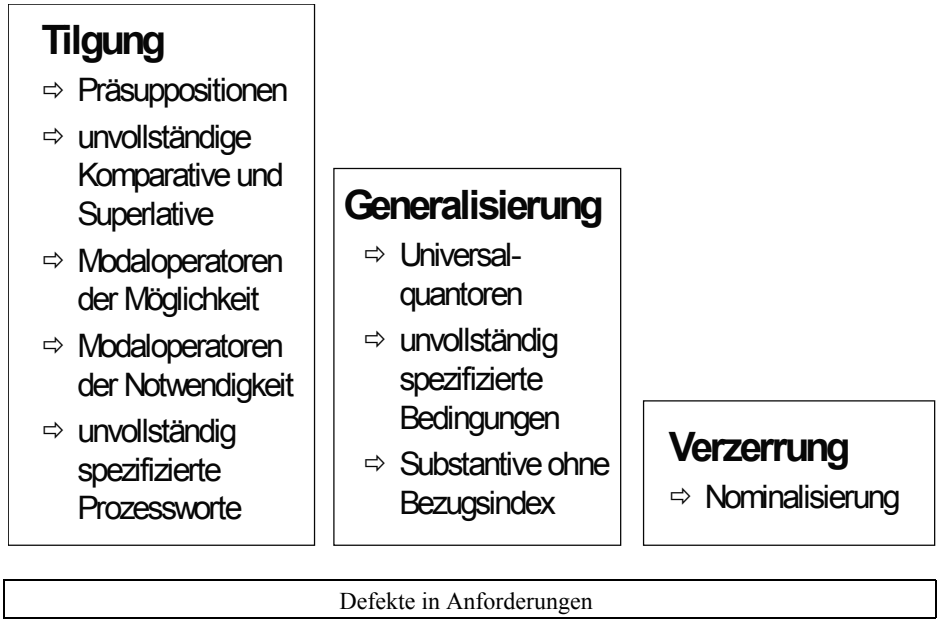


Abbildung 2: Die häufigsten sprachlichen Defekte in Anforderungsspezifikationen

Tilgungen

Der Prozess der Tilgung vernichtet Informationen bei der Wissensdarstellung und reduziert die Welt auf Ausmaße, mit denen wir umgehen können. Diese Reduktion kann in gewissen Kontext sinnvoll sein; im Bereich der Anforderungsdefinition können durch Tilgungen wichtige Informationen verloren gehen.

Die folgenden Beispiele zeigen verschiedene Arten von Tilgungen und geben damit einen Einblick in das Spektrum sprachlicher Vertreter von Tilgungstransformationen.

Unvollständig definierte Prozesswörter

Prozesswörter sind diejenigen Wörter in einem Satz, die einen Vorgang beschreiben. Dies müssen nicht notwendigerweise Verben sein, auch Substantive kommen für diese Rolle in Frage. Um vollständig zu sein, müssen zu einem Prozesswort normalerweise mehrere Argumente oder Substantivgruppen angegeben sein.

Beispiel 1:

*„Das System soll Datenverluste **melden.“***

Das Prozesswort „melden“ ist erst dann vollständig definiert, wenn die folgenden Fragen beantwortet sind.

Fragen:

„Wer meldet? Was wird gemeldet? Wie oder auf welche Weise wird gemeldet? Wo oder an wen wird gemeldet? Wann wird gemeldet? Wie lange wird gemeldet?“

Manche Prozesswörter (Verben, Adjektive oder Adverbien) erfordern mehr als ein Substantiv[1]. Das Verb „übertragen“ benötigt z. B. zu seiner vollständigen Erklärung zumindest die drei Ergänzungen, *was* übertragen wird, *von wo* und *wohin* übertragen wird. Das Sprachgefühl gibt darüber Auskunft, um welche Informationen ein Prozesswort ergänzt werden muss, um vollständig beschrieben zu sein.

Regel 1: Unvollständig definierte Prozesswörter aufdecken

Man bestimme die Prozesswörter und stelle fest, ob man sich einen Satz mit diesem Prozesswort vorstellen kann, der mehr Akteure[1] enthält. Ist das der Fall, so ist aus dem Originalsatz etwas getilgt worden. Hält man diese fehlende Information für wissenswert, kann man gezielt danach fragen.

Beispiel 2:

„Die **Auslastung** der internen Systemressourcen soll **überwachbar** sein.“

In dieser Aussage ist „auslasten“ in seiner substantivierten Form als Prozesswort enthalten. Vollständig definiert ist dieses Prozesswort erst, wenn folgende Fragen beantwortet werden:

Fragen:

„Wer oder was genau ist ausgelastet? Durch was ist er/sie/es ausgelastet?“

[1] Verb (Zeitwort), Adjektiv (Eigenschaftswort), Adverb (Umstandswort), Substantiv (Hauptwort)

Des weiteren ist das Prozesswort „überwachen“ referenziert. Hier ist die Aussage vollständig definiert, wenn die folgenden Fragen geklärt wurden:

Fragen:

„Wer überwacht? Was wird überwacht? Wie oder auf welche Weise wird überwacht?“ Wann wird überwacht?

Anhand der gestellten Fragen zu den Prozesswörtern lässt sich feststellen, ob die gegebenen Aussagen ausreichend spezifiziert sind oder die Anforderung um weitere Informationen ergänzt werden muss.

Unvollständige Komparative und Superlative

Eine weitere Art der Tilgung, die sich durch sprachliche Methoden auffinden lässt, sind unvollständige Komparative (Vergleiche) und Superlative (Steigerungen).

Sprachliche Vertreter für Komparative und Superlative sind Begriffe wie „besser“, „schneller“, „leichter“, „am schnellsten“. Sie besitzen in der deutschen Sprache folgenden Aufbau:

Adjektiv[2] + Endung „-er/-en“ oder „-ste“ oder Ergänzungen wie „weniger“, „mehr“.

In der englischen Sprache verhält es sich ähnlich:

Adjektiv + Endung „-er“, „-est“ oder Ergänzungen wie „more“, „less“, „least“.

Ein Komparativ oder ein Superlativ benötigt immer einen Bezugspunkt, um vollständig definiert zu sein. Zudem müssen das Messkriterium/die Maßeinheit (z. B. Meter, Sekunden) und die Messgenauigkeit (z. B. $\pm$ 0,1 Meter, $\pm$ 0,0001 Sekunde) angegeben werden. Wichtig ist natürlich auch, dass die Eigenschaft, die verglichen wird, objektiv messbar ist. Dies ist im folgenden Beispiel 2 nicht der Fall. Hier würde eine sprachliche Überprüfung zu einer vollständigen Umformulierung der Anforderung führen.

Regel 2: Unvollständige Komparative und Superlative erkennen

Man bestimme Vergleiche und Steigerungen in einer Anforderung. Enthalten sie die Bezugspunkte, auf den sich der Vergleich oder die Steigerung bezieht? Existiert eine Messmethode, mit der die jeweils geforderte Eigenschaft überprüft werden kann? Mit welcher Genauigkeit kann gemessen werden?

Beispiel 1:

[2] Eigenschaftswort

„Sonstige Konservierungen sollen von **schnelleren** Speichermedien aus gewonnen werden.“

Fragen:

Worauf bezieht sich der Begriff *sonstige*? Welches sind die sonstigen? *Schneller* im Vergleich wozu? *Schneller* in Bezug auf den Schreib- oder den Lesezugriff? Ist der Zugriff in Millisekunden messbar?

Beispiel 2:

„Die Berechtigungen sollen **leicht** änderbar sein.“

Frage:

Leicht im Vergleich wozu? Wodurch wird eine Änderung *leicht durchführbar*? Welches Messkriterium soll dafür bei der Abnahme angewendet werden?

Generalisierungen

Die menschliche Fähigkeit, Erfahrungen zu generalisieren, ist ein Prozess, der sinnvoll und überlebenswichtig ist. Der Mensch kann dadurch eine einmal gewonnene Erfahrung auf verwandte Zusammenhänge übertragen. Wichtig ist allerdings die passende Wahl der Gruppe von Sachverhalten, auf die eine Erfahrung transferiert wird.

Durch den Prozess der Generalisierung entstehen im Software-Engineering bei einer oftmals zu groben Verallgemeinerung globale Anforderungen an das System, die nur für einen Teilbereich des Systems richtig und sinnvoll sind. Für andere Teilbereiche des Spektrums können diese Anforderungen allerdings falsch sein. Typisch für den Prozess der Generalisierung ist das Unterdrücken bzw. Weglassen von Sonder- und Fehlerfällen. Im Folgenden werden zwei häufig auftretenden Varianten der Generalisierung dargestellt.

Universalquantoren

Universalquantoren sind Angaben über Häufigkeiten. Sprachliche Vertreter dieser Gruppe sind Begriffe wie „nie”, „immer”, „kein”, „jeder”, „alle”. Die Gefahr bei der Verwendung von Universalquantoren besteht darin, dass das spezifizierte Verhalten oftmals nicht für wirklich alle referenzierten Objekte der genannten Gruppe zutrifft.

Beispiel 1:

„**Jede** Meldung soll für die Aufzeichnung zusätzlich mit einem Zeitstempel etikettiert werden.“

Fragen:

Wirklich *jede* Meldung? Oder gibt es vielleicht Fälle, in denen der Zeitstempel nicht benötigt wird?

Auch Sätze, die keine explizite Aussage über die Menge von Objekten enthalten, für die das spezifizierte Verhalten eintreten soll, müssen überprüft werden. Häufig enthalten sie die implizite Annahme, dass das angegebene Verhalten für alle überhaupt relevanten Objekte gelten soll.

Beispiel 2:

„Das System soll eine durch den Benutzer eingeleitete Sicherung von aufgezeichneten Daten auf Band ermöglichen."

Fragen:

Darf jeder Benutzer die Sicherung einleiten? Dürfen auch mehrere Benutzer gleichzeitig Sicherungen einleiten? Darf sie immer, zu jedem Zeitpunkt eingeleitet werden (auch wenn bereits eine Sicherung läuft)? Sollen wirklich alle aufgezeichneten Daten seit Anbeginn des Systems gesichert werden können?

Der Realisierungsrahmen dieser Anforderung liegt zwischen einer einfachen Sicherungssoftware von 500 DM und einer Robotiksicherungsstation für einige Millionen DM.

Regel 3: Universalquantoren hinterfragen

Man bestimme die Universalquantoren einer Anforderung und frage sich jeweils, ob das geforderte Verhalten des Systems für wirklich *alle* Objekte aus der Menge gelten soll, die durch den Quantor gegeben wird. Vielleicht gibt es Ausnahmen, die dann zusätzlich spezifiziert werden müssen. Zusätzlich untersuche man jeden Satz dahingehend, ob eine Menge an Objekten ausdrücklich definiert ist, für die das spezifizierte Verhalten gelten soll.

Unvollständig spezifizierte Bedingungen

Ein weiterer Indikator für einen möglichen Informationsverlust durch eine Generalisierung sind unvollständig spezifizierte Bedingungen, deren Vertreter unter anderem Begriffe wie „wenn", „dann", „falls" und „abhängig von" sind. Bei der Verwendung unvollständiger Bedingungen fehlen häufig ganze Verarbeitungszweige in der Spezifikation.

Beispiel 1:

„Wenn das Datum X über die Schnittstelle Y angefordert wird, **dann** soll die Zeit für die Verarbeitung von X unter 0.5 s liegen."

Fragen:

Welche Zeitanforderungen existieren in den Fällen, in denen das Datum X *nicht* über die Schnittstelle zum Y angefordert wird? Was passiert wenn das Datum X über mehrere Schnittstellen gleichzeitig geliefert wird oder über keine der Schnittstellen?

Das Stellen der aufgeführten Fragen ermöglicht es, die Informationen über die restlichen Bedingungskonstellationen zu erheben.

Regel 4: Unvollständig spezifizierte Bedingungen erkennen

Man bestimme die Bedingung(en) in der Anforderung und überprüfe, ob sowohl für den Fall, dass die Bedingung eintritt ein Verhalten spezifiziert ist (then-Zweig), als auch dafür, dass sie nicht eintritt (else-Zweig). Zusätzliche Fragen: „Sind alle möglichen Entscheidungskriterien (Bedingungen) aufgezählt?", „Sind alle möglichen Varianten geschildert?"

Verzerrung

Das Phänomen der Verzerrungstransformation tritt bei der Erstellung von Anforderungsdokumenten häufig in Form von Nominalisierungen auf.

Nominalisierung

Bei einer Nominalisierung wird ein Prozess zu einem Ereignis umformuliert. Dabei kann es passieren, dass hierbei wichtige Informationen über den Prozess verlorengehen. Sprachlich erkennt man eine Nominalisierung an einem Prozesswort (Verb), das zu einem Ereigniswort (Substantiv oder Argument) umgeformt wurde.

Beispiel 1:

„Es sollen **Datenverluste** erkannt und gemeldet werden."

Hinter den Begriff Datenverlust verbirgt sich folgender Prozess:

„Jemand/etwas verliert zu einem Zeitpunkt auf eine gewisse Weise bestimmte Daten."

Fragen:

Welche Daten gehen verloren? Wodurch gehen die Daten verloren? Wie wird der Verlust der Daten erkannt/gemeldet?

Beispiel 2:

„Das bewirkt einen **Neustart**."

Fragen:

Was wird neu gestartet? Mit welchen Daten wird der Neustart durchgeführt? Wie wird der Neustart durchgeführt? Wer initiiert den Start (automatisch/manuell)? Wodurch wird der Neustart beendet? Was passiert während des Verlaufs des Neustarts (potentielle Abbrüche, Fehlerbedingungen)?

Grundsätzlich spricht nichts dagegen, nominalisierte Begriffe für einen komplexen Prozess zu verwenden, sofern der Prozess eindeutig definiert ist. Die Definition sollte dabei keinen Spielraum für eine Interpretation des Prozesses erlauben und sowohl den Verlauf des Prozesses als auch sämtliche Eingabe- und Ausgabeparameter klären. Bei Nominalisierungen ist es nicht das Ziel sie zu vermeiden, sondern sie nur dann zu verwenden, wenn der dahinter liegende Prozess klar ist.

Regel 5: Nominalisierungen hinterfragen

Man überprüfe jedes Substantiv im Satz und frage sich, ob man sich ein Verb denken kann, das einen Vorgang beschreibt und das ähnlich klingt/aussieht und ihm in der Bedeutung ähnelt. Dazu zwei Tests:

1. Passt das Substantiv sinnvoll in die Phrase „ein(e) andauernde(r) ..." (im Sinne von kontinuierlich), so handelt es sich um einen Vorgang.

2. Beschreibt ein Substantiv etwas, das man nicht anfassen kann?

Kann eine der beiden fragen mit „Ja" beantwortet werden, so muss das Substantiv dahingehend hinterfragt werden, ob nicht mit dem unterschlagenen Vorgang auch Informationen verloren gegangen sind.

Bemerkung zum ersten Test:

Dieser Test liefert fast immer gute Ergebnisse. Wörter wie *Tisch*, *Papier*, *Fenster* ergeben als *ein andauernder Tisch* etc. keinen Sinn. Hingegen passen die Wörter Projektion, Ausdruck (einer Datei auf einem Drucker) und Transformation ganz gut, etwa als *ein andauernder Ausdruck*. Vielleicht hängen an dem Vorgang *Ausdrucken* noch eine Reihe von fehlenden Informationen, etwa *was* gedruckt werden soll, *wie* und *wo*.

Ergebnisse und Schlussfolgerung

Die hier vorgestellte natürlichsprachliche Methoden des Requirements-Engineering wurden bereits in einer Vielzahl industrieller Großprojekte erfolgreich angewandt. Die zitierten Arbeitsergebnisse der Linguisten und Psychologen stellen - übertragen auf die Softwareentwicklung - ein leicht handhabbares und sehr effektives Werkzeug zum Erstellen und Prüfen von Prosaanforderungen dar. Die damit erstellten Anforderungen sind eine optimale Ausgangsbasis für die Systementwicklung, z. B. in Form einer Objektmodellierung.

Tatsächlich ist es möglich, das von der Informatik geforderte formale Modell, das die Gesamtheit der Anforderungen bilden soll, mit Hilfe der natürlichsprachlichen Methoden nicht-formal darzustellen. Somit wird Auftraggeber und Anwender die Möglichkeit eröffnet, die Beschreibung ihres Problems vollständig zu verstehen. Die Anwendung dieser Methode erzielte bei zahlreichen Projekten z. B. bei der Deutschen Flugsicherung oder der Deutschen Post AG große Erfolge.

Literatur

[Ban94] R. Bandler, J. Grinder, Metasprache und Psychtherapie, Junfermann, 1994

[Bec70] J. Bechert, Einführung in die generative Transformationsgrammatik, Hueber, 1970

[Boe90] D. Boedicker, Handbuch-Knigge, Spektrum, 1990, Erstellung von Software-Handbüchern

[Chom73] N. Chomsky, Sprache und Geist, Suhrkamp, 1973

[Gar73] T. Gardner, Hauptstränge der modernen Linguistik? Chomsky und die generative Grammatik, Vandenhoeck & Rupprecht, 1973

[Sophist01] C. Rupp, Requirements Engineering und Management, Professionelle, iterative Anforderungsanalyse für die Praxis, Hanser 2001; www.sophist.de