



Leseprobe

Chris Rupp, SOPHIST GROUP

Requirements-Engineering und -Management

Professionelle, iterative Anforderungsanalyse für die Praxis

ISBN: 978-3-446-41841-7

Weitere Informationen oder Bestellungen unter

<http://www.hanser.de/978-3-446-41841-7>

sowie im Buchhandel.

## Das SOPHIST-REgelwerk – Psychotherapie für Anforderungen

---



- Wenn Anforderungen von Menschen formuliert werden, finden bewusst und unbewusst Transformationen statt, die zu unvollständigen, verzerrten oder sogar falschen Informationen führen.
- Mittels SOPHIST-REgelwerk, das auf dem Therapieansatz „Neurolinguistisches Programmieren (NLP)“ basiert, können Sie Lücken und Verfälschungen in Ihren Anforderungen systematisch aufdecken und gezielt beheben.
- Therapieren Sie Ihre Anforderungen Schritt für Schritt – diese Regeln helfen Ihnen dabei, wirklich das zu verstehen, was Ihr Gegenüber will.

### 6.1 Vom Phänomen der Transformation – Sprachliche Effekte

Anforderungen werden von vielen verschiedenen Stakeholdern formuliert – Menschen unterschiedlichen Wissens, sozialer Prägung und Erfahrung. Diese Vielfalt spiegelt sich in den Anforderungen wider und birgt die Gefahr von Informationsverlusten, Unvollständigkeits- oder Zweideutigkeiten. Vor diesem Hintergrund stellen sich bei der Ermittlung von Anforderungen folgende zentrale Fragen:

- Was meint der Stakeholder wirklich?
- Wie erfährt man das, was der Urheber der Anforderung meinte, als er sie formulierte?

Die Antwort ist leider nicht so trivial, wie man es sich wünschen würde. Um dieses Problem zu lösen, haben wir Methoden aus den Disziplinen Linguistik, Informatik, Psychologie und Psychotherapie im sogenannten „SOPHIST-REgelwerk“ vereinigt.

Bevor wir aber in die Tiefen des SOPHIST-REgelwerks einsteigen, stellen wir Ihnen zunächst sein theoretisches Fundament vor, das Neurolinguistische Programmieren (NLP). Wir zeigen Ihnen, wie Wissen von der Wahrnehmung der Realität bis hin zum sprachlichen Ausdruck Schritt für Schritt transformiert wird und zu den bekannten Unzulänglichkeiten in der natürlichen Sprache führt.

Sie kennen diese Grundlagen oder Sie finden sie zu theoretisch? Kein Problem. Steigen Sie direkt bei der praxistauglichen Beschreibung der Regeln des SOPHIST-REgelwerks (vgl. Abschnitt 6.3) ein, die wir und auch unsere Kunden in vielen Projekten bereits erfolgreich eingesetzt haben. Wir haben dabei großen Wert darauf gelegt, die Anwendung der Regeln anhand vieler Beispiele zu erläutern. Ferner finden Sie viele Tipps und Tricks aus unserer Projekterfahrung, die Ihnen den Einsatz des SOPHIST-REgelwerks in Ihrem Projektalltag erleichtern werden.

### 6.2 Die Wurzeln – Das Neurolinguistische Programmieren

Das SOPHIST-REgelwerk basiert im Wesentlichen auf dem Metamodell der Sprache des Therapieansatzes Neurolinguistisches Programmieren (NLP) [Bandler75], [Bandler94]. Die Regeln des NLP helfen Therapeuten, die bei jedem Menschen vorhandene Intuition, ob ein Satz „richtig“ (linguistisch: wohlgeformt) ist oder nicht, bewusst zu machen.

Bei der Systementwicklung kann der Requirements-Engineer mit ähnlichen Regeln Effekte in natürlichsprachlichen Anforderungen systematisch aufspüren und beheben. Wegweisend bei der Entwicklung natürlichsprachlicher Ansätze war der Linguist Noam Chomsky [Chomsky65], der wichtige linguistische Grundkonzepte erforschte und diese in der generativen Transformationsgrammatik beschrieb (siehe Artikel „Transformations-Linguistik“ unter [www.sophist.de](http://www.sophist.de)).



Das Metamodell der Sprache.



Als NLP-Kenner oder Theorie-Muffel können Sie direkt zu Abschnitt 6.3 springen.

Das ist das Modell des Modells „Sprache“, d. h. es beschreibt das Modell der Sprache.

Aufbauend auf einigen Thesen Chomskys wurde in der Psychotherapie ein Modell menschlicher Kommunikation und Ausdrucksweise erstellt und verfeinert. Maßgeblich daran beteiligt waren der Psychologe und Linguistikprofessor John Grinder und der Informatiker und Gestalttherapeut Richard Bandler. Mitte der 70er Jahre des 20. Jahrhunderts war es ihr Ziel, eine für jedermann erlernbare Therapieform zu entwickeln, die darauf beruht, dass der Therapeut die persönliche Wirklichkeit (Wahrnehmung) des Patienten versteht. Dazu muss der Therapeut herausfinden, was genau der Klient mit seinen Aussagen meint.

### 6.2.1 Transformationsprozesse

Jeder Requirements-Engineer sieht sich mit Anforderungen konfrontiert, die ein reales oder noch zu entwickelndes System beschreiben sollen. Wenn Menschen ein System beschreiben, also natürlichsprachliche Anforderungen formulieren, wird ein Transformationsprozess, wie in Abbildung 6.1 veranschaulicht, vollzogen.

Software-System,  
Produkt, Anwendung,  
Unternehmensprozess  
o. ä.

Die eingeschränkte persönliche Wahrnehmung führt zur Wahrnehmungstransformation.

Der sprachliche Ausdruck des persönlichen Wissens führt zur Darstellungstransformation.

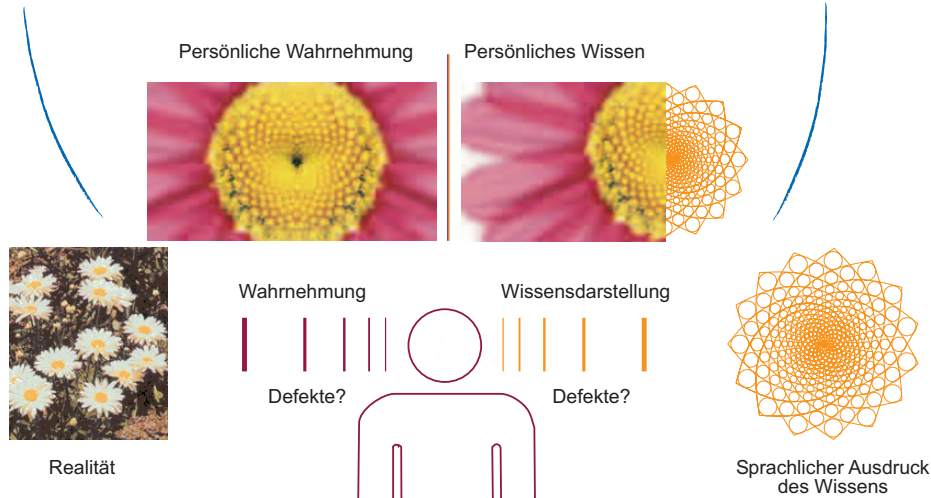


Abbildung 6.1: Transformationen als mögliche Informationsvernichter

Ausgangspunkt (in der Abbildung 6.1 links) ist die zu beschreibende Realität, das eigentlich von den Stakeholdern gewünschte System. Rechts befindet sich hingegen die Beschreibung des Systems, also die formulierten Anforderungen, die der Systemanalytiker im Anforderungsdokument vorliegen hat. Zwischen den beiden Enden kann eine (mehr oder weniger tiefe) Lücke klaffen, die aufgrund von komplexen, meist unbewussten Transformationsprozessen entsteht.

RE-Bauernregel:  
Wenn der Stakeholder sein Wissen transformiert, die Anforderung er mit Verlust notiert.

Um die Lücke zwischen Realität und sprachlichem Ausdruck schließen zu können, müssen Sie als Requirements-Engineer zunächst die möglichen Transformationsarten kennen. Nur wenn Sie wirklich verstehen, warum die Realität „verfälscht“ oder nicht mehr abbildungstreu in den Anforderungen formuliert ist, können Sie die Realität auch gezielt ergünden.

## 6 Das SOPHIST-REgelwerk

Jeder Mensch kann nur einen kleinen Bruchteil der gesamten Realität aufnehmen.

Transformationsprozesse resultieren aus unserer menschlichen Natur. Der Mensch wird durch seine soziale Prägung, sein Vorwissen und seine Erfahrungen in seiner Wahrnehmung beeinflusst. Psychologen sprechen hierbei von der so genannten „persönlichen Wirklichkeit“. Die persönliche Wirklichkeit wirkt sich auf die persönliche Wahrnehmung der Realität aus, so dass sich jeder Mensch seine eigenen Vorstellungen von der Realität macht.

So wird zum Beispiel Bibliothekar A und Bibliothekar B dasselbe Leihobjekt, ein Buch, gezeigt. Bibliothekar A erfasst sofort die ansprechende Gestaltung mit Grafiken, die harmonisch mit dem Textkörper zusammenspielen und das Gesamtbild auflockern. Bibliothekar B hingegen nimmt dasselbe Buch vollkommen anders wahr. Er bemerkt die klare Gliederung des Werkes mit Einleitung, Hauptteil und Schluss, welche dem Leser das Verständnis erleichtern soll. Die ansprechenden Grafiken bemerkte er überhaupt nicht. Beide Bibliothekare haben somit unterschiedliche Bilder (in der Sprache der Psychologen: Modelle) von dem gleichen Buch – von der gleichen Realität.

Woran liegt es aber, dass Bibliothekar A sich ausschließlich auf Grafiken und Layout konzentriert, während Bibliothekar B vor allem die thematische Gliederung wahrnimmt? Dieses Phänomen bezeichnen Psychologen als Wahrnehmungstransformation. Jeder Mensch wendet bei der Wahrnehmung der Realität unbewusst solche Transformationen (Umgestaltungsprozesse) an, die abhängig von seiner persönlichen Wirklichkeit sind.

Daraus ergibt sich für die Systemanalyse eine wichtige Erkenntnis:  
Die Gesamtinformation über die Realität ist immer „auf mehrere Köpfe“ verteilt, da jeder Mensch nur einen Bruchteil der Realität wahrnimmt.

Dies hat zur Folge, dass es zur vollständigen Abbildung der Realität nicht ausreicht, nur einen Bibliothekar zu befragen.

Jeder Mensch setzt ein gewisses Grundwissen bei anderen voraus.

Interessanterweise treten die Transformationen aber noch an einer weiteren Stelle im Abbildungsprozess von der Realität zur Anforderung auf. Und zwar dann, wenn das persönliche Wissen in Sprache ausgedrückt wird (in Abbildung 6.1 als Wissensdarstellung bezeichnet). Wird Bibliothekar B nach dem Aufbau befragt, könnte die Antwort „äußerst sinnvoll“ lauten, obwohl er eigentlich den genauen Aufbau (nämlich Einleitung-Hauptteil-Schluss) meint. Er wandelt (transformiert) somit sein vorhandenes Wissen um, während er es in Sprache ausdrückt.

Analog transformiert der Schreiber von Anforderungen oder ein Interviewpartner im Gespräch sein Wissen beim konkreten Formulieren der Anforderungen.

Es gibt also zweierlei Transformationsprozesse:

Fokussierung

**Wahrnehmungstransformationen** treten auf, weil jeder Mensch die Realität anders wahrnimmt und sich ein individuelles Bild davon macht.

Vereinfachung

**Darstellungstransformationen** treten auf, weil eine Wandlung erfolgt, sobald ein Mensch sein Wissen (dieses Bild) in Sprache ausdrückt.

Transformationen sind grundsätzlich nichts Problematisches und, wie wir noch sehen werden, sogar lebenswichtig. Für die Anforderungsanalyse stellen sie jedoch ein entscheidendes Problem dar, da mit den Transformationen möglicherweise essenzielle Informationen ver-

lorengehen. Es macht schon einen Unterschied, ob der Bibliothekar von einem Buch, das sich in Einleitung-Hauptteil-Schluss gliedert, spricht, oder ob er den Aufbau mit „äußerst sinnvoll“ beschreibt.

Bei der Entwicklung eines Systems können sich solche Nuancen in enormen Kostenunterschieden ausdrücken. Um dem entgegenzuwirken, müssen die Transformationen rückgängig gemacht und die Anforderungen mit den verlorengegangenen Informationen angereichert werden.

Transformationsprozesse führen zu Informationsverlusten und Informationsverfälschungen, die aufgedeckt werden müssen.

Nun stellt sich die Frage: Ist es überhaupt möglich, diese Transformationen rückgängig zu machen? Und wenn ja, wie geht der Requirements-Engineer dabei vor? Zunächst lässt sich sagen, dass nur bestimmte Transformationen rückgängig gemacht werden können.

Die Wahrnehmungstransformation (Realität → Persönliche Wahrnehmung) ist nicht mehr aufzulösen, da das individuelle Bild eines Menschen sich nicht problemlos beeinflussen lässt. Dies ändert sich unter Umständen, wenn sich die Erfahrungen und das soziale Umfeld des Menschen ändern. Unabhängig davon erfasst jeder Mensch aber immer nur einen Teil der Realität.

Um Wahrnehmungstransformationen zu kompensieren, muss der Requirements-Engineer stets mehrere Stakeholder zum gleichen Sachverhalt befragen.

Die aus einer Befragung von mehreren Personen resultierenden unterschiedlichen Erfahrungen und Meinungen wirken nicht nur der Wahrnehmungstransformation entgegen, sondern stellen zudem auch eine Bereicherung für jedes Projekt dar. Wenn hier Synergien entstehen, kann ein insgesamt besseres System konstruiert werden.

Darstellungstransformationen (Persönliches Wissen → Ausdruck in Sprache) lassen sich dagegen glücklicherweise sehr gut auflösen. Voraussetzung dafür ist allerdings, dass der Requirements-Engineer die möglichen Transformationsarten und deren Konsequenzen genau kennt.

Um Darstellungstransformationen aufzulösen, kann der Requirements-Engineer das SOPHIST-REgelwerk einsetzen.

Darstellungstransformationen werden auch auf dem Gebiet der Linguistik untersucht. Linguisten nehmen an, dass der Mensch im Geist (unbewusst oder unterbewusst) von seiner Wahrnehmung eine vollständige sprachliche Repräsentation, eine so genannte Tiefenstruktur, bildet. Beginnt er dann zu reden oder zu schreiben, so trifft er eine Reihe von Auswahlentscheidungen hinsichtlich der Gestalt der zu artikulierenden Informationen und bildet somit die zugehörige Oberflächenstruktur. Er wählt aus einer Menge von Transformationen eine bestimmte oder meist mehrere aus, durch deren Anwendung auf die Tiefenstruktur die Oberflächenstruktur entsteht.

Das ist die „transformierte“ Form des Originals, also das, was der Mensch aussagt, wenn er das Gedachte in Sprache ausdrückt.

RE-Bauernregel:  
Lässt man Transformationen schleifen, muss man später tief in den Beutel greifen.

Auch wenn Sie es manchmal wollen ...  
therapieren Sie bitte nicht Ihre Stakeholder.

D. h. sprachliche Effekte gezielt erkennen und hinterfragen.

Das ist das Original, also das, was dem Menschen gedanklich vorschwebt, bevor er es in Sprache ausdrückt.

## 6 Das SOPHIST-REgelwerk

Ein Satz (eine Oberflächenstruktur) ist also eine andere Form des „Originals“ (der Tiefenstruktur) im Kopf des Menschen. Der Oberflächenstruktur fehlen unter Umständen aber Teile oder es werden Teile falsch dargestellt. Aus dieser Annahme leitet sich auch das (linguistische) Ziel der Analyse von Anforderungen ab:

Um ein vollständiges und nicht verändertes Bild der persönlichen Wahrnehmung eines Stakeholders zu erlangen, muss der Requirements-Engineer aus der Oberflächenstruktur die zugehörige Tiefenstruktur ermitteln.

Die Tatsache, dass Menschen sich bei der Bildung von Oberflächenstrukturen an Regeln halten, führte Chomsky zu der Annahme, dass das Verhalten der Menschen bei der Bildung sprachlicher Äußerungen offenbar regelgeleitet ist. Einen möglichen (und plausiblen) Satz von Regeln legte Chomsky in der bereits erwähnten Generativen Transformationsgrammatik erstmalig dar.

Ähnlich dem Ansatz von Chomsky können wir auch in der Systemanalyse Regeln definieren, durch die ein systematisches Aufspüren der Transformationen bei Prosa-Anforderungen und Aussagen des Kunden möglich ist. Der Requirements-Engineer will einen Teil der persönlichen Wirklichkeit des Verfassers verstehen, kennt aber nur die sprachlichen Oberflächenstrukturen des Verfassers. Kann er aus ihnen die verwendeten Transformationen folgern, so ist es ihm durch gezieltes Hinterfragen möglich, an die Tiefenstruktur dessen zu gelangen, was der Verfasser auszudrücken suchte.

Reminder:  
Neurolinguistisches  
Programmieren

### 6.2.2 Kategorien der Darstellungstransformation

Aufbauend auf den Erkenntnissen der Linguistik unterscheiden Bandler und Grinder im NLP drei prinzipielle „Arten der Umgestaltung“: Tilgung, Generalisierung und Verzerrung. Diese Einteilung mag vielleicht nicht disjunkt erscheinen, hat sich in der Praxis aber als durchaus nützlich erwiesen.

Sprachlichen Effekte  
lassen sich nicht  
immer strikt nur einer  
Kategorie zuordnen.

In Abbildung 6.2 sind diese drei Kategorien als Symbole grafisch dargestellt. Diese Symbole verwenden wir in den nachfolgenden Abschnitten, um den Regeln des SOPHIST-REgelwerks die Art der Umgestaltung entsprechend zuzuordnen.

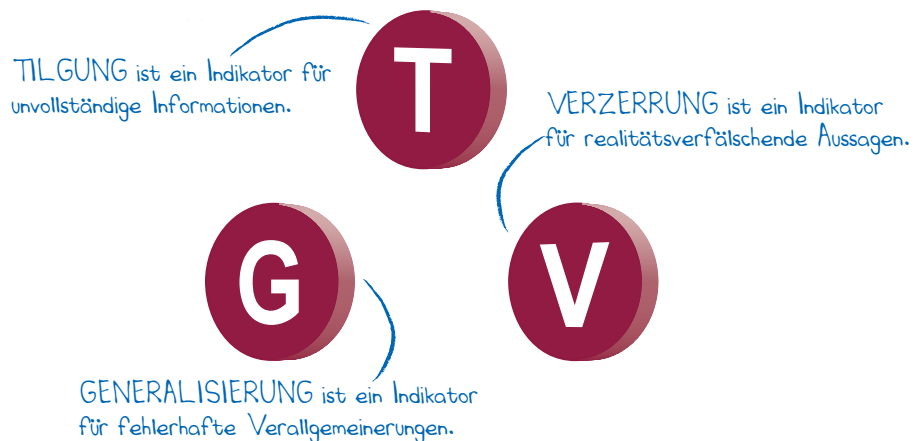


Abbildung 6.2: Klassifizierung sprachlicher Effekte

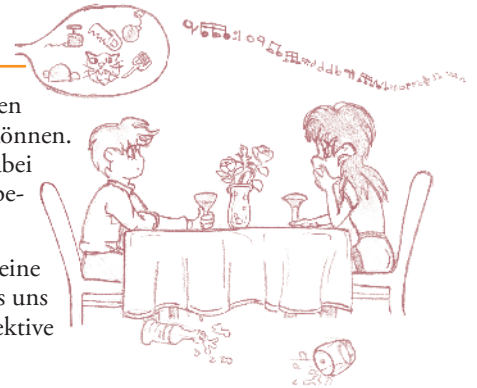


Nachfolgend beschreiben wir die Transformationskategorien nach Bandler und Grinder. Die Definitionen sind rein sprachwissenschaftlicher Natur, können aber durchaus auch auf die sprachlichen Effekte im Zusammenhang mit Anforderungen angewendet werden.

### Tilgung

Der Prozess der Tilgung reduziert die Flut an Informationen, mit denen wir konfrontiert werden, auf Ausmaße, mit denen wir umgehen können. Wenn wir wiederum Informationen weitergeben, dann tilgen wir dabei unbewusst jene Teile, die wir als „selbstverständlich“ ansehen oder unbewusst beim Informationsempfänger als bekannt voraussetzen.

Mit Hilfe der Tilgung ist es uns zum Beispiel möglich, das allgemeine Stimmengewirr in einem Raum mit vielen Menschen so zu filtern, dass uns nur noch die Stimme unseres Gesprächspartners bewusst erreicht (selektive Wahrnehmung).



**T**

**Tilgung** (engl. Deletion) ist ein Prozess, durch den wir unsere Aufmerksamkeit selektiv bestimmten Dimensionen unserer (im Moment möglichen) Erfahrungen zuwenden und andere ausschließen. Tilgung reduziert die Welt auf Ausmaße, mit denen wir umgehen können. [Bandler94]

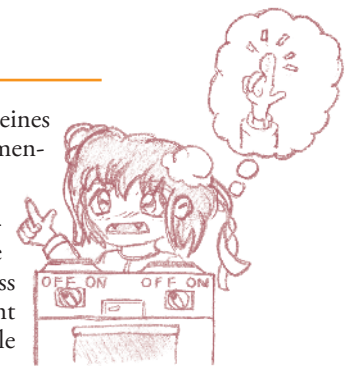
RE-Bauernregel:  
Wenn Informationen nicht vollständig kommen ans Licht, fehlendes Wissen herauskitzeln ist des RElers Pflicht.

Diese Reduktion kann in einem gewissen Kontext sinnvoll sein, in Anforderungen an ein Softwaresystem können durch Tilgungen jedoch wichtige Informationen verloren gehen. So mag ein Bibliothekar mitteilen: „Zum Ende eines Monats werden die Konten mit den anfallenden Gebühren belastet“. Ein anderer Bibliothekar, vertraut mit den Arbeitsprozessen der Bibliothek, wird aus diesem Satz die getilgten Inhalte wahrscheinlich rekonstruieren können: „Es fallen Mahngebühren für Kunden an, die Leihfristen überschreiten.“ Der unbedarfte Zuhörer, dem diese getilgten Informationen nicht bekannt sind, könnte die Aussage des Bibliothekars aber vollkommen anders interpretieren.

### Generalisierung

Die Generalisierung ist ein Prozess, durch den eine einmalige Erfahrung (Teile eines persönlichen Modells) auf andere, ähnliche Sachverhalte oder verwandte Zusammenhänge übertragen und somit als allgemeingültig angenommen wird.

Die menschliche Fähigkeit, Erfahrungen zu generalisieren, ist ein Prozess, der überlebenswichtig sein kann. Ein Beispiel für eine solche Generalisierung ist die heiße Herdplatte, bei deren schmerzvoller Berührung ein Kind richtig generalisiert, dass alle heißen Herdplatten gefährlich sind, nicht nur die eine, an der es sich verbrannt hat. Eine nicht ganz so richtige Generalisierung wäre die grundsätzliche Angst, alle Herdplatten anzufassen (also auch kalte), oder die Annahme, dass man sich nur an dieser einen speziellen Herdplatte die Finger verbrennen kann. Was bleibt, ist das Wissen, dass die Berührung irgendeiner heißen Herdplatte zu unerwünschten Verbrennungen führt, und hoffentlich auch die Vorsicht gegenüber allen Herdplatten (da sie ja heiß sein können).





## 6 Das SOPHIST-REgelwerk

RE-Bauernregel:  
Wenn Generalisierungen dem Stakeholder nicht richtig gelingen, der REler die Information auf den Punkt muss bringen.

G

**Generalisierung** (engl. Generalization) ist der Prozess, durch den Elemente oder Teile eines persönlichen Modells von der ursprünglichen Erfahrung abgelöst werden, um dann die gesamte Kategorie, von der diese Erfahrung ein Beispiel darstellt, zu verkörpern. [Bandler94]

Generalisierung ist auch im Kontext der Systemanalyse nützlich und sinnvoll. Wichtig ist allerdings die passende Gruppierung der Sachverhalte, auf die sich das zu beschreibende Systemverhalten bzw. die Systemeigenschaft generalisierend beziehen soll.

Durch eine zu starke Generalisierung entstehen globale Anforderungen an das System, die wahrscheinlich nur für einen Teilbereich des Systems richtig und sinnvoll sind. Sonder- und Fehlerfälle gehen hierbei häufig verloren. **So könnte der Bibliothekar berichten, dass die Kennungen aller Bücher, die zur Kunstabteilung gehören, mit dem Kürzel KST beginnen. In der Kunstabteilung gibt es aber auch Exfolianten, die aufgrund ihrer Übergröße nicht in die vorgesehenen Regale passen und daher an anderer Stelle untergebracht sind. Diese sind mit dem Kürzel EXF ausgezeichnet. Der Bibliothekar hat diesen Sonderfall versehentlich vergessen.**



### Verzerrung

Die Verzerrung ist der Vorgang, durch den die Realität umgestaltet oder sogar verfälscht wird. Von einer Verzerrung spricht man, wenn ein Aspekt der Realität so verändert wird, dass er beim Betrachter zu einem Zerrbild führen kann.

Verzerrung findet statt, wenn eine Situation mit Ausdrücken beschrieben wird, die nicht entsprechend dieser Situation sind. Somit wird die Situation ausgeschmückt oder getrübt und die eigene Wahrnehmung oder die des Empfängers (wenn auch meist unbewusst) beeinflusst. Das Phänomen der Verzerrung sorgt dafür, dass der Mensch neue Informationen leichter in seine Vorstellungen integrieren kann. So manches Detail wird nötigenfalls ein wenig verändert, um in das bereits erstellte Bild einer Situation zu passen. Somit verändern wir häufig eine Information oder unsere Wahrnehmung so, wie es für uns richtig erscheint.

Jede Verzerrung kann eine Menge an Informationen vernichten und ist damit implizit auch ein Tilgungsdefekt.

V

**Verzerrung** (engl. Distortion) ist der Prozess, der es uns ermöglicht, in unserer Erfahrung sensorischer Einzelheiten eine Umgestaltung vorzunehmen. [Bandler94]

RE-Bauernregel:  
Wenn der Stakeholder eine Umgestaltung hat vorgenommen, der REler der Realität auf die Spur muss kommen.

Verzerrungen stellen für jeden Requirements-Engineer eine harte Nuss dar, da er häufig nicht entscheiden kann, ob formulierte Sachverhalte korrekt sind, ob sie verzerrt wurden oder ob er sie durch seine eigene Wahrnehmung selbst verzerrt. **Ein Bibliothekar berichtet beispielsweise, dass bei jeder Reservierung eine Sperrung erfolgen soll. Der Bibliothekar verzerrt hier einen Sachverhalt, da er den Prozess der „Sperrung“ als selbstredend und somit vereinfacht beschreibt. Unter Umständen kann dieser Prozess aber aus einer Reihe komplexer Prozessschritte bestehen.**

### 6.3 Vom Umgang mit sprachlichen Effekten

Jede der erwähnten Transformationskategorien Tilgung, Generalisierung und Verzerrung zeigt sich in bestimmten so genannten sprachlichen Effekten. Jeder Effekt kann zu qualitativ minderwertigen Anforderungen führen, jedoch nicht in allen Fällen auch zu einem Defekt.

Inwiefern ein sprachlicher Effekt ein Problem darstellt und deshalb behoben werden sollte, hängt von vielen Randbedingungen ab. Ein wichtiger Faktor ist sicherlich das Detaillierungsniveau, auf dem Sie gerade Anforderungen schreiben.

Sprachliche Effekte auf der Ebene von Zielen oder von eher generischen Anforderungen (z. B. Spezifikationslevel 0 und 1, siehe Kapitel 3 „Von der Idee zur Spezifikation“) sind normal und auf dem angestrebten Detaillierungsniveau auch nicht immer zu vermeiden.

Da es allerdings Leser geben wird, die Ihre Spezifikation nur auf diesen Ebenen betrachten werden, ist es wichtig, dass die für diese Leserschaft wichtigen Informationen nicht durch Effekte vernichtet werden.



Gewisse grundlegende Regeln sollten Sie bereits auf Spezifikationslevel 0 und 1 einhalten:

- Schreiben Sie Ihre Anforderungen stets in vollständigen Sätzen.
- Formulieren Sie immer im Aktiv (siehe Regel 1 in Abschnitt 6.4.1).
- Verwenden Sie Begriffe konsistent und vermeiden Sie Synonyme oder Homonyme.
- Falls es bereits ein Glossar (siehe Kapitel 8 „Dokumentation von Anforderungen“ und Kapitel 7 „Schablonen“) gibt, verwenden Sie die darin definierten Begriffe.
- Formulieren Sie Prozesse durch Vollverben.

Wie z. B. berechnen,  
drucken, exportieren,  
überweisen, ....

Schreiben Sie sehr detaillierte Anforderungen (ab Spezifikationslevel 2), die auch noch Teil eines Vertrages für eine externe Beauftragung werden, so sind sprachliche Effekte bei weitem schädlicher und sollten sehr kritisch geprüft und nach Möglichkeit beseitigt werden. Auch wenn Ihre Anforderungen syntaktisch meist wohlgeformt sind, können sie sprachliche Effekte enthalten. Ihre Aufgabe ist es nun, diese Effekte zu erkennen und zu beheben, sofern die fehlenden oder verzerrten Informationen wesentlich sind.

Grammatikalisch richtig

Wir empfehlen daher, die Regeln des SOPHIST-REgelwerks im Wesentlichen ab Spezifikationslevel 2 anzuwenden. Ab diesem Level formulieren Sie detailliertere Anforderungen, durch die Anforderungen auf Level 0 und Level 1 verfeinert werden.

## 6 Das SOPHIST-REgelwerk

Unabhängig davon, auf welchem Spezifikationslevel Sie Anforderungen formulieren, sollen Sie sich immer an folgende grundlegende Regel halten:

Für jeden Spezifikationslevel gibt es zu der obigen Grundregel „Formulieren Sie Prozesse durch Vollverben“ eine Verfeinerung, die immer zu beachten ist:

- Verwenden Sie immer nur Vollverben, die für den jeweiligen Spezifikationslevel auch „gute“ Vollverben darstellen. Denn nicht jedes Vollverb ist ein für den Spezifikationslevel angemessenes Vollverb.

Auf Spezifikationslevel 1 werden beispielsweise für das Bibliothekssystem die Use-Cases definiert, von denen sich einer auf die Kundenverwaltung bezieht. Als Vollverb formuliert notieren wir den Use-Case als „Kunde verwalten“. Für Spezifikationslevel 1 ist das Vollverb „verwalten“ durchaus angemessen. Ab Spezifikationslevel 2 verwenden wir aber niemals das Vollverb „verwalten“, da es für diesen Level viel zu nichtssagend, zu „schwammig“ ist. Vielmehr verfeinern wir das Vollverb „verwalten“ durch detailliertere und für Level 2 geeignete Vollverben, wie beispielsweise „eingeben“, „speichern“, „anzeigen“, usw.

Setzen Sie hier auch die Anforderungsschablone aus Kapitel 7 „Schablonen“ ein.

Wenn Sie das Auftreten sprachlicher Effekte von Anfang an vermeiden wollen, dann setzen Sie das SOPHIST-REgelwerk direkt konstruktiv während der Ermittlung und bei der Formulierung Ihrer Anforderungen ein. Wollen Sie Effekte in bereits dokumentierten Anforderungen aufspüren? Dann empfehlen wir den analytischen Einsatz des SOPHIST-REgelwerks.

Kombinieren Sie das REgelwerk mit den analytischen Prüftechniken aus Kapitel 10 „Prüftechniken für Anforderungen.“

Und wenn Sie basierend auf einer Qualitätsaussage über Ihre Anforderungen eine Grundlage für die Entscheidungsfindung schaffen wollen, messen Sie die formale Qualität Ihrer Anforderungen mit Hilfe von Qualitätsmetriken (vgl. Kapitel 11 „Qualitätsmetriken“), die im Wesentlichen auf den Regeln des SOPHIST-REgelwerks aufbauen.

Grundsätzlich sollte es aber nicht Ihr Ziel sein, alle Anforderungen so weit zu perfektionieren, bis sie von allen Effekten befreit sind. Konzentrieren Sie sich auf die Informationen, die Sie für den weiteren Prozess benötigen. Hinterfragen Sie vor allem an den Stellen, an denen fehlende oder falsche Informationen das höchste Risiko für Ihren Projekterfolg darstellen.

Gehen Sie risikogetrieben vor.

Z. B. ab Spezifikationslevel 2

Gehen Sie auf Nummer sicher: Setzen Sie die Regeln des SOPHIST-REgelwerks umso intensiver ein,

- je detaillierter Sie Anforderungen beschreiben,
- je kritischer Ihr System und damit Ihr Projekt ist.

Eine Rangliste hierzu finden Sie in Abschnitt 6.8.

Nach unserer Erfahrung treten sprachliche Effekte in unterschiedlicher Häufigkeit auf. Daher ist es wichtig und ratsam, dass Sie sich zunächst auf die Gruppe der gefährlichen Wiederholungstäter konzentrieren.

## 6.4 Das Vorgehen beim SOPHIST-REgelwerk – Anforderungen auf die Couch gelegt



Basierend auf den Thesen des Linguisten Noam Chomsky und dem Metamodell der Sprache lässt sich auch ein Regelwerk für die Systemanalyse aufstellen: das SOPHIST-REgelwerk. Es basiert auf Regeln, nach denen der Mensch unbewusst vorgeht, wenn er sich natürlichsprachlich, d. h. in Wort und Schrift, äußert.

Mithilfe des SOPHIST-REgelwerks können auf definierte, systematische Art und Weise mehrdeutige, unvollständige und widersprüchliche Aussagen in Anforderungsdokumenten gefunden werden (analytische Qualitätssicherung). Auch präventiv, also im Zuge der konstruktiven Qualitätssicherung, lässt sich die Kenntnis der Transformationsprozesse beim Ermitteln und Dokumentieren von Anforderungen einsetzen (vgl. Kapitel 10 „Prüftechniken für Anforderungen“).

Das SOPHIST-REgelwerk ist eine Technik, die es Ihnen ermöglicht, Tilgungen, Generalisierungen und Verzerrungen in Anforderungen zu erkennen und somit fehlende und verzerrte Informationen in Anforderungsquellen aufzudecken.

RE-Bauernregel:  
Hast du einen wilden Anforderungshaufen, lass ihn gleich das REgelwerk durchlaufen.

Durch jede einzelne Regel des SOPHIST-REgelwerks können unterschiedliche sprachliche Effekte in einer Anforderung gezielt gefunden werden. Obwohl sich die Regeln in diesem Kapitel ausschließlich auf Anforderungen beziehen, lassen sie sich natürlich auf sämtliche natürlichsprachliche Ausdrucksformen (mündlich oder schriftlich) anwenden.

Z. B. auch auf semi-formale Notationen, wie beispielsweise Diagramme, Tabellen oder Bilder, in denen natürliche Sprache vorkommt.

Das Vorgehen beim Einsatz einer einzelnen Regel aus dem SOPHIST-REgelwerk ist denkbar einfach:

- **Schritt 1: Identifizieren** von sprachlichen Effekten in Anforderungen anhand von Signalwörtern.
- **Schritt 2: Analysieren** verlorener oder verfälschter Informationen durch gezielte Fragestellungen.
- **Schritt 3: Bereinigen** sprachlicher Mängel oder inhaltlicher Fehler durch Umformung der Anforderung mittels der gegebenen Antworten, so dass Effekte eliminiert werden.

Signalwort finden

Fragen zum Signalwort stellen

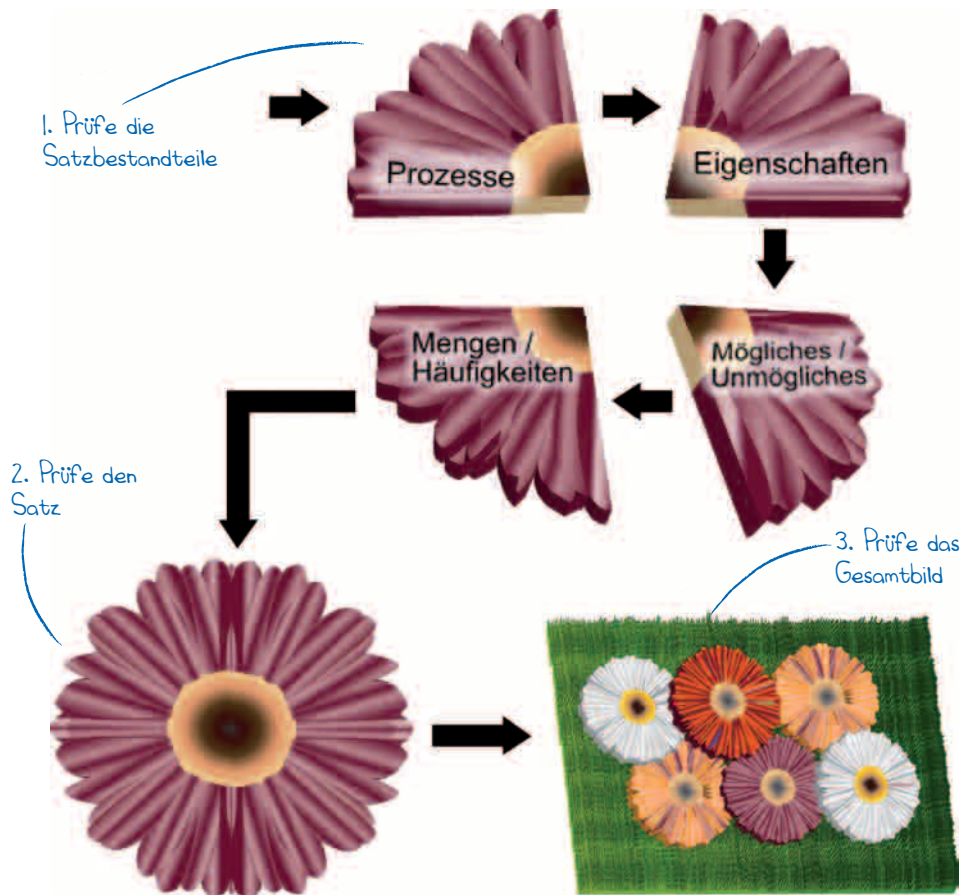
Antworten einarbeiten

Vom Grundsatz her ist das SOPHIST-REgelwerk eine Ansammlung von Regeln. In der Praxis fällt es uns jedoch häufig schwer, mit einem solch umfangreichen Pool an Regeln umzugehen, wenn darauf keine Ordnung definiert ist.

## 6 Das SOPHIST-REgelwerk

Vermutlich kennen Sie das auch: Sie erfahren von einem Kollegen, dass es da eine ganz tolle Ansammlung von Regeln und Empfehlungen gibt, durch die Sie die Qualität Ihrer täglichen Arbeit steigern können. Voller Euphorie stürzen Sie sich auf das Regelwerk und erkennen, dass alle Regeln einfach und verständlich sind. Mit Ihrem neu erlangten Wissen machen Sie sich voller Elan an Ihre nächste Aufgabe. Und dann sitzen Sie ganz verzweifelt da und wissen nicht, wie Sie diese eigentlich simplen Regeln am besten einsetzen sollen.

Sie stellen sich sicherlich die folgenden Fragen: Wie soll ich denn konkret vorgehen? Mit welcher Regel fange ich an und brauche ich wirklich alle Regeln? In welcher Reihenfolge soll ich welche Regeln mit welcher Priorität anwenden?



**Abbildung 6.3:** Das Vorgehen beim Einsatz des SOPHIST-REgelwerks

Den Fallstricken der natürlichen Sprache kann man wahrscheinlich nie endgültig entgehen. Es gibt nicht den einen, wahren Leitfaden, der immer gültig ist. Dennoch lässt sich ein Muster aufzeigen, das sich bei der Anwendung der Regeln in unseren Projekten bereits mehrfach bewährt hat. In Abbildung 6.3 ist dieses Vorgehen dargestellt.

Nach unserer Projekterfahrung sollten Sie beim Prüfen einer Anforderung immer vom Kleinen (den Satzbestandteilen) zum Großen (dem Gesamtbild) vorgehen. Grundsätzlich durchlaufen Sie das folgende Vorgehen Anforderungssatz für Anforderungssatz:

- **1. Prüfen Sie zunächst die einzelnen Wörter (Bestandteile) eines Anforderungssatzes**, um ihn schrittweise durch wesentliche, aber getilgte Informationen zu vervollständigen. Bestimmte Wörter in Ihrem Anforderungssatz stellen dabei Ihre Signalwörter dar, auf die Sie achten müssen. Jedes einzelne Signalwort weist auf einen spezifischen semantischen Effekt hin. An diesem Signalwort erkennen Sie dann auch die Regel, die Sie anwenden müssen.
- **2. Prüfen Sie danach den Anforderungssatz als Ganzes**, um ihn von unnötigem Ballast zu befreien und ihn somit aufs Essenzielle zu reduzieren. Für das System nicht relevante Informationen oder Floskeln blähen Ihren Satz nur unnötig auf und lassen ihn unübersichtlich werden. Solche Satzbestandteile können Sie dann ohne Informationsverlust entweder aus der Anforderung als Kommentar herauslösen oder komplett eliminieren.
- **3. Prüfen Sie zum Schluss, wie sich der Anforderungssatz in das zu beschreibende Gesamtbild einfügt.** Durch diese Kontrolle können Sie schrittweise das Gesamtbild („Big Picture“) vervollständigen. Nutzen Sie hierzu die Informationen, die Ihnen in dem Anforderungssatz bereits vorliegen. Fehlt Ihnen beispielsweise die Spezifikation einer bestimmten Systemfunktionalität oder -eigenschaft, damit der untersuchte Anforderungssatz sinnvoll in das Gesamtbild passt? Auch hier helfen Ihnen wieder bestimmte Signalwörter dabei, selbst die vom Gesamtbild fast vollständig getilgten Funktionalitäten oder Prozesse aufzufinden.

Wenn das Signalwort „speichern“ fällt, müssen Sie die typischen „W-Fragen“ zu „speichern“ stellen.

Wenn etwas „von grüner Farbe“ sein soll, dann ist es ausreichend, „grün“ zu schreiben.

Wenn nach einem „erfolgreichen Plausibilitätscheck“ Daten gespeichert werden sollen, muss auch der Negativfall spezifiziert sein.

Das beschriebene Vorgehen hört sich relativ einfach an. Die Analyse von Anforderungen ist aber leider alles andere als trivial. Insbesondere beim konstruktiven Einsatz des Vorgehens sollten Sie beachten, dass Sie bereits beim Analysieren der Satzbestandteile häufig nicht nur den untersuchten Anforderungssatz mit fehlenden Informationen anreichern, sondern auch auf zusätzliche Anforderungen stoßen oder den Anforderungssatz verfeinern müssen. Gerade in diesem Fall ist dann ein diszipliniertes Vorgehen nach dem Schema „Anforderungssatz nach Anforderungssatz“ äußerst wichtig. Ansonsten könnten Sie schnell Schiffbruch erleiden und in den (teilweise mächtig ausschlagenden) Wellen neuer Anforderungen oder gar neuer Prozesse untergehen.

Behalten Sie bei Interviews oder beim Schreiben von Anforderungen den roten Faden.

Wir empfehlen hier, sich die neuen bzw. verfeinerten Anforderungen zu notieren, diese aber noch nicht direkt detailliert zu analysieren. Sie sollten zunächst die Analyse der ursprünglich untersuchten Anforderung fortsetzen. Prüfen Sie dann anschließend die neu hinzugekommenen Anforderungen im Detail.

Nachfolgend werden alle Regeln des SOPHIST-REgelwerks im Detail vorgestellt und anhand vieler Beispiele erläutert. Generell gilt, dass die Reihenfolge, in der die Regeln beschrieben werden, das Vorgehen bei deren Anwendung widerspiegelt. Für NLP-Kenner ist jede Regel, die sich sinnvoll einer der Transformationskategorien gemäß Bandler und Grinder zuordnen lässt, mit dem entsprechenden Symbol aus Abbildung 6.2 gekennzeichnet.



RE-Bauernregel:  
Will der REler die  
Anforderung korrekt  
spezifizieren, muss er  
sich auf ihre Einzel-  
teile konzentrieren.

### 6.5 Prüfen der Satzbestandteile

Zunächst konzentrieren wir uns auf die einzelnen Bestandteile (Wörter) des Anforderungssatzes, um diesen schrittweise mit fehlenden Informationen zu vollständigen.

#### 6.5.1 Prüfen der Prozesse

Verb, aber auch Nominalisierung  
oder Funktionsverbgefüge



Wenn ein Prozess in einer Anforderung durch ein Prozesswort beschrieben wird, müssen für diesen Prozess weitere Informationen beschrieben werden.

Funktionalität, Vorgang oder Tätigkeit

Um den Interpretationsspielraum zu reduzieren, muss beispielsweise für den Prozess „drucken“ beschrieben werden, wer wann was worauf drucken möchte. Auch das Verb „übertragen“ benötigt zu seiner vollständigen Erklärung zumindest die vier Ergänzungen, wer etwas überträgt, was übertragen wird, von wo und wohin.

Ihr Sprachgefühl gibt Ihnen darüber Auskunft, um welche Informationen ein Prozesswort ergänzt werden muss, um vollständig spezifiziert zu sein.

Beim Prüfen der Prozesswörter liegt Ihr Hauptaugenmerk auf der in einer Anforderung beschriebenen Funktionalität. Das zu identifizierende Signalwort ist demnach das Prozesswort, also ein Verb (oder eine abgeleitete Form des Verbs), durch das ein Prozess beschrieben wird.

RE-Bauernregel:  
Wenn der REler im  
Passiv schreibt, der  
Akteur auf der  
Strecke bleibt.

### Schritt 1: Prüfe auf Aktivformulierung

Ein sprachlicher Effekt, nämlich die Tilgung des Akteurs, lässt sich dadurch vermeiden, dass Anforderungen im Aktiv formuliert werden. Liegt eine Anforderung dagegen im Passiv vor, so muss die Anforderung ins Aktiv umformuliert werden.

T

#### Regel 1: Formulieren Sie jede Anforderung im Aktiv.

Prüfen Sie, ob die Anforderung im Aktiv formuliert ist. Wenn nicht, wurde der Akteur, also die ausführende Person oder Einheit, getilgt.

Erfragen Sie den Akteur und ergänzen Sie diesen in der Anforderung, indem Sie die Anforderung im Aktiv formulieren.



Wer soll  
den Prozess ausführen  
bzw. über die Eigenschaft  
verfügen?

selbständige Systemaktivität Schnittstellenanforderung

Gerade bei der Formulierung von Anforderungen ist die Angabe des Akteurs entscheidend, da wichtig ist, ob die Aktivität vom System, vom Nachbarsystem oder vom Benutzer durchgeführt wird (vgl. Kapitel 7 „Schablonen“). Auf diese Weise erhält die Anforderung einen höheren Informationsgehalt und das Prozesswort wird näher spezifiziert, da angegeben wird, wer den Prozess durchführt bzw. über eine bestimmte Eigenschaft verfügen muss.

Benutzerinteraktion



Bei der im Passiv formulierten Anforderung „Das Benutzerkennwort muss an einem Terminal des Bibliothekssystems eingegeben werden.“ ist beispielsweise nicht klar, wer das Kennwort eingeben kann. Im Aktiv muss hingegen ein Akteur oder Verantwortlicher angegeben werden: „Der Bibliothekskunde muss das Benutzerkennwort an einem Terminal des Bibliothekssystems eingeben.“

Diese Aussage ist schon ein Stück klarer, doch noch lange keine Anforderung an ein System. Hier wird lediglich etwas von einem Benutzer gefordert (und Sie wollen bei der Abnahme wohl kaum Ihren Benutzer testen). Leiten Sie nun die Anforderung an das System ab, also die Funktionalität, die das System zur Verfügung stellen muss, damit der Benutzer die Eingabe machen kann.

Dies könnte beispielsweise das Bereitstellen einer Eingabemöglichkeit für das Benutzerkennwort sein: „Das Bibliothekssystem muss es dem Bibliothekskunden ermöglichen, ein Benutzerkennwort an einem Terminal des Bibliothekssystems einzugeben.“

### Schritt 2: Identifiziere die geforderte Funktionalität

Nachdem nun die Anforderung im Aktiv formuliert ist, lassen sich sprachliche Effekte, durch die eine konkret geforderte Funktionalität verschleiert wird, leichter identifizieren, wenn die in der Anforderung beschriebene Funktionalität durch ein Vollverb formuliert ist.

Verben, die alleine das Prädikat eines Satzes bilden, z. B. „speichern“, „anzeigen“ oder „eingeben“  
Hilfsverben sind dagegen Verben wie „haben“, „sein“, „werden“.

V

#### Regel 2: Drücken Sie Prozesse durch „gute“ Vollverben aus.

Schwammig formulierte Prozesswörter (Verben, Tätigkeitswörter) verschleiern die konkret geforderte Funktionalität.

Hinterfragen Sie die konkret geforderte Funktionalität und drücken Sie diese durch ein „gutes“ Vollverb aus.

Nicht alle Vollverben sind auf jedem Spezifikationslevel „gute“ Vollverben, wie z.B. „verwalten“ oder „prüfen.“

Hinterfragen Sie beispielsweise die schwammige Formulierung „umfassende Informationen angeben“ in der Anforderung „Das Bibliothekssystem muss dem Bibliothekar bei der Suche nach einem Leihobjekt umfassende Informationen über das gefundene Leihobjekt angeben.“, so wird sich wahrscheinlich die konkret vom System geforderte Funktionalität „anzeigen“ ergeben. Durch die Umformulierung in das Vollverb „anzeigen“ wird dann schnell deutlich, dass weitere Angaben zur vollständigen Beschreibung des Vollverbs erkundet werden müssen. Zumindest die Frage, was genau angezeigt werden soll, müssen Sie bei der Umformulierung der Anforderung beantworten.

Welche konkrete Funktionalität wird gefordert? Wie lautet das VOLLVERB?



Damit beschäftigt sich Regel 6.

Definieren Sie das Vollverb anschließend in Ihrer Prozesswortliste (vgl. Kapitel 7 „Schablonen“).

## 6 Das SOPHIST-Regelwerk

Bitte verwenden Sie ab Spezifikationslevel 2 keine Vollverben, wie z. B. „prüfen“, „verwalten“ oder „abbrechen“.

Eine verbesserte (aber noch lange nicht perfekte) Anforderung könnte, nachdem die Antworten auf die Fragen eingearbeitet wurden, folgendermaßen lauten: „Nachdem das Bibliothekssystem die Suche nach einem Leihobjekt durchgeführt hat, muss das Bibliothekssystem dem Bibliothekar alle verfügbaren Stammdaten des Leihobjekts anzeigen.“

Auch die Prozessbeschreibung „Plausibilität prüfen“ in der Anforderung „Nachdem der Bibliothekar das Speichern eines neuen Kunden initiiert hat, muss das Bibliothekssystem die Plausibilität der eingegebenen Kundendaten prüfen.“ ist streng genommen eine schwammige Prozessformulierung. Der Prozess „Prüfen“ erfordert vielmehr die Beschreibung des nachweisbaren Ergebnisses, wie beispielsweise „Daten speichern“ oder „Fehlernachricht anzeigen“. Hier muss die Anforderung also umformuliert werden: „Nachdem der Bibliothekar den Speichervorgang eines neuen Kunden initiiert hat und falls das Alter des neu eingegebenen Kunden größer als oder gleich 18 Jahre ist, muss das Bibliothekssystem die Daten des neu eingegebenen Kunden speichern.“

Damit beschäftigt sich Regel 17

Die Umformulierung der Anforderung hat dazu geführt, dass Sie eine präzise Fallunterscheidung machen müssen. Sie müssen genau beschreiben, welches Verhalten Ihr System bei Eintritt einer bestimmten Bedingung oder Bedingungskombination aufweisen muss. Jetzt müssen Sie natürlich noch die weiteren Fälle der obigen Anforderung hinterfragen und spezifizieren.

Das Offenlegen der konkret zu spezifizierenden Funktionalität ist aber nicht immer so einfach. Manche Anforderungen sind so effektbehaftet, dass der eigentlich zu spezifizierende Prozess vollkommen verschleiert wird. In diesem Fall ist es die Aufgabe des Requirements-Engineers, den konkreten Prozess zu identifizieren und zu präzisieren. Typischerweise treten Prozessformulierungen auch in Form von sogenannten „Nominalisierungen“ oder „Funktionsverbgefügen“ auf.

Ein (oft länger wählender) Prozess wird zu einem (einmaligen) Ereignis verzerrt.

### Nominalisierungen

RE-Bauernregel: Will der Stakeholder die Anforderung verderben, macht er Substantive aus den Verben.

Nominalisierungen sind Substantive (Nomen, Hauptwörter), durch die komplexe Prozesse in einem Begriff (einem „substantivierten Verb“) zusammengefasst werden, die im Detail aufwändig zu beschreiben wären. Durch diese Umwandlung gehen wichtige Informationen verloren, die für die Beschreibung des Prozesses wesentlich sind. Bei der sprachlichen Analyse werden alle Nominalisierungen identifiziert und analysiert.

Achten Sie auf Signalwörter wie „die Reservierung“, „die Registrierung“ (und auch „Requirements-Engineering“).

V

### Regel 3: Lösen Sie Nominalisierungen auf.

Nominalisierungen können einen kompletten Prozess verschleiern.

Analysieren Sie jede Nominalisierung und prüfen Sie, ob der Prozess an anderer Stelle im Anforderungsdokument ausreichend spezifiziert ist. Ist dies nicht der Fall, müssen Sie für die Nominalisierung:

- Eine oder mehrere neue Anforderungen mit jeweils einem „guten“ Vollverb spezifizieren, ODER
- Einen neuen Glossareintrag erstellen.

Analysiert man beispielsweise die Anforderung „Das System muss eine Archivierung ermöglichen.“, wird sich herausstellen, dass das Substantiv „Archivierung“ eine Nominalisierung ist. Nun müssen Sie die genaue Intention des Stakeholders hinterfragen. Der hinter der Anforderung steckende Prozess wird, als Vollverb ausgedrückt, wahrscheinlich „archivieren“ lauten. Eine verbesserte Anforderung, in der die Frage nach dem „Was muss archiviert werden?“ beantwortet wird, könnte dann die folgende sein: „Das Bibliothekssystem muss die Daten der aus dem Bestand aussortierten Leihobjekte archivieren.“

Ist das Substantiv eine Nominalisierung? Welcher konkrete Prozess steckt dahinter?



In der Beispielanforderung des Bibliothekssystems „Bei der Rückgabe eines reservierten Leihobjekts muss das Bibliothekssystem an den Bibliothekskunden, der das Leihobjekt reserviert hat, eine Benachrichtigung versenden.“ stellen die Begriffe „Rückgabe“ und „Benachrichtigung“ jeweils Nominalisierungen dar, die genauer analysiert werden müssen.

Was geschieht alles bei der „Rückgabe“?

Wann und durch wen oder was wird die „Rückgabe“ eingeleitet bzw. beendet?

Welche Fehler-/Ausnahmefälle können bei der „Rückgabe“ auftreten?

Werden alle Fragen aus dem Anforderungsdokument heraus beantwortet, so kann die obige Anforderung weitestgehend unverändert bestehen bleiben. Nur dann, wenn hinter den Nominalisierungen „Rückgabe“ und „Benachrichtigung“ Prozesse stecken, die nicht vollständig spezifiziert oder definiert sind, müssen die fehlenden Informationen dem Anforderungsdokument hinzugefügt werden.

Wird durch die Nominalisierung ein (unter Umständen komplexer) Prozess verzerrt, muss dieser in der Regel mit allen zugehörigen Schritten für Normal- und Ausnahmeverhalten durch entsprechende Anforderungen beschrieben werden. In der obigen Anforderung könnte die „Rückgabe“ eines Leihobjektes beispielsweise aus folgendem Verlauf bestehen: „Leihobjekt auswählen“, „Status des Leihobjektes ändern“, „Anzahl der ausgeliehenen Leihobjekte für Bibliothekskunde aktualisieren“...

Die Auflösung von Nominalisierungen resultiert meist in mehreren zusätzlichen Anforderungen.

Dieser Verlauf erfordert also eine Reihe von Anforderungen. Im nächsten Schritt muss geprüft werden, ob diese Anforderungen bereits im Anforderungsdokument spezifiziert sind. Ist dies nicht der Fall, müssen diese Anforderungen ergänzt werden. Nach Regel 2 müssen Sie also zunächst für jede in der Nominalisierung verschleierte Funktionalität das entsprechende Vollverb identifizieren, wie „auswählen“, „ändern“, „aktualisieren“, und diese Funktionalitäten anschließend detailliert analysieren (vgl. Regel 6). Wäre für das Bibliothekssystem die „Rückgabe eines Leihobjektes“ bisher noch nicht spezifiziert, würde dies gleichzeitig auch die Einführung eines neuen Use-Cases „Leihobjekt zurückgeben“ bedeuten.

Das Hinterfragen von Nominalisierungen deckt häufig auch einen komplett getilgten Use-Case auf.

Grundsätzlich ist es nicht unser Ziel, Nominalisierungen in Anforderungen ganz zu vermeiden oder zu verbieten.

Gehen Sie auf Nummer sicher: Versuchen Sie immer, Nominalisierungen aufzulösen und die hinter den Nominalisierungen verborgenen Prozesse durch Vollverben auszudrücken. Setzen Sie Nominalisierungen wirklich nur dann ein, wenn

- Ihre Entscheidung dafür fachlich motiviert ist,
- der Begriff an zentraler Stelle definiert ist und
- die Definition der Nominalisierung keinen Raum für Interpretation lässt.

## 6 Das SOPHIST-REgelwerk

Nominalisierungen werden u. a. auch dazu verwendet, um nicht den eigentlichen Prozess, sondern ein in den Prozess involviertes Ding oder Objekt zu beschreiben. Wenn Sie für den nominalisierten Begriff eine eindeutige Begriffsdefinition hinterlegt haben, spricht nichts dagegen, ihn auch in Ihren Anforderungen zu verwenden. Für das Bibliothekssystem wäre die Nominalisierung „Benachrichtigung“ ein solcher Kandidat, da die „Benachrichtigung“ im Kontext der obigen Beispielanforderungen ein Ding oder Objekt darstellt, das versendet werden soll.

Wenn Sie Nominalisierungen zur Benennung eines Prozesses in Ihren Anforderungen verwenden, empfehlen wir Ihnen, diese durch das Postfix „-vorgang“ oder „-prozess“ zu ergänzen. Das kennzeichnet die Nominalisierung eindeutig als einen Prozess. So können Sie auch vermeiden, dass Sie den gleichen nominalisierten Begriff für verschiedene, voneinander abzugrenzende Sachverhalte verwenden, wie beispielsweise das mit einem Prozess zusammenhängende Objekt und den Prozess selbst.

Z. B. die „Benachrichtigung“

Z. B. der „Benachrichtigungsvorgang“

### Funktionsverbgefüge

Achten Sie auf Signalwörter wie „zur Verfügung stellen“, „zu Ende bringen“ oder „zum Ausdruck bringen“

Nicht selten werden zur Beschreibung von Systemfunktionalitäten auch sogenannte Funktionsverbgefüge verwendet. Funktionsverbgefüge sind Kombinationen aus inhaltsarmen Verben (machen, können, haben, sein, ...) und sinngebenden Substantiven.



#### Regel 4: Lösen Sie Funktionsverbgefüge auf.

Funktionsverbgefüge können einen kompletten Prozess verschleiern.

Analysieren Sie das Funktionsverbgefüge und formulieren Sie für jede geforderte Funktionalität eine neue Anforderung, die das Systemverhalten durch ein „gutes“ Vollverb beschreibt.



Wird der Prozess durch ein Funktionsverbgefüge ausgedrückt? Welcher konkrete Prozess steckt dahinter?

Durch die enge Verbindung zwischen Verb und Substantiv in Funktionsverbgefügen tritt der eigentliche Prozess vollkommen in den Hintergrund. Er wird erst dann offengelegt, wenn das Funktionsverbgefüge hinterfragt und die für den Prozess wesentlichen Funktionalitäten durch Vollverben beschrieben werden.

Was genau bedeutet „eine Veränderung erfahren“?

In der Beispielanforderung „Nachdem ein Leihobjekt entliehen wurde, muss der Status des Leihobjektes eine Veränderung erfahren.“ verbirgt das Funktionsverbgefüge „eine Veränderung erfahren“ das unvollständig spezifizierte Vollverb „verändern“. Eine verbesserte Anforderung lautet z. B. „Nachdem ein Bibliothekskunde ein Leihobjekt entliehen hat, muss das Bibliothekssystem den Status des Leihobjektes in den Status „entliehen“ ändern.“

In unserem REgelwerk haben wir die Auflösung von Funktionsverbgefügen als eine separate Regel aufgeführt, weil sie, ähnlich den Nominalisierungen, nach unserer Erfahrung häufig mehrere Funktionalitäten gleichzeitig oder sogar einen kompletten Use-Case verschleiern. So kann in der Anforderung „Es sollten statistische Auswertungen zur Verfügung gestellt werden.“ das Funktionsverbgefüge „zur Verfügung stellen“ die Prozesse „Erstellen“, „Anzeigen“, „Drucken“, „Importieren“ etc. enthalten sein.

Was genau heißt „zur Verfügung stellen“?

Funktionsverbgefüge finden häufig dann Anwendung, wenn das Hauptproblem eines Anforderungsstellers darin besteht, dass er selbst noch nicht genau weiß, was er in Zukunft wirklich braucht und will. Unser Bibliothekar weiß, dass er statistische Auswertungen benötigt, ist aber noch unsicher bezüglich der konkreten Ausgestaltung dieser Funktionalität. Dann ist es Ihre Aufgabe als Requirements-Engineer, den Anfordernden bei der Präzisierung und damit der Auseinandersetzung mit dieser Funktionalität zu unterstützen, indem er beispielsweise die fünf W-Fragen beantwortet (siehe Regel 6).

... oder Stakeholder beschreiben einfach nur Prozesse, die sie nicht wirklich kennen.

Beispielsweise kann die Frage „In welcher Form müssen statistische Auswertungen zur Verfügung gestellt werden?“ zu der Aussage führen, dass statistische Auswertungen sowohl in elektronischer Form als auch als Papiausdruck vorliegen müssen. Ferner muss auch die Frage geklärt werden, aus welcher Quelle die statistischen Auswertungen stammen. Eine erste verbesserte Anforderung könnte folgendermaßen lauten: „Das Bibliothekssystem muss dem Bibliothekar die Möglichkeit bieten, statistische Auswertung erstellen, anzeigen und ausgeben zu können.“

Richtig! Das ist eine Nominalisierung, die nach Regel 3 noch tiefer analysiert werden muss.

### Schritt 3: Zerlege in vereinzelte Anforderungssätze

Bis jetzt haben wir die in einer Anforderung geforderte Funktionalität identifiziert und in mindestens ein Vollverb transformiert. Unter Umständen hat das Aufdecken der geforderten Funktionalität dazu geführt, dass die ursprüngliche Anforderung nun mehrere Vollverben beinhaltet. Unterschiedliche durch Vollverben formulierte Funktionalitäten verlangen in der Regel aber auch unterschiedliche Zusatzinformationen, so dass (zumindest zu deren Analyse) mehrere Anforderungssätze daraus generiert werden sollten.

Reminder: Eine Anforderung kann aus einem oder mehreren Anforderungssätzen bestehen.

T

#### Regel 5: Schreiben Sie für jedes Prozesswort genau einen Anforderungssatz.

Jedes Prozesswort (Vollverb) wird in der Regel durch unterschiedliche Informationen (Satzbestandteile, wie Subjekt, Objekt und Ergänzungen) vollständig beschrieben.

Identifizieren Sie die Prozesswörter (Vollverben) in der Anforderung und formulieren Sie für jedes identifizierte Prozesswort einen eigenen Anforderungssatz mit genau einem Vollverb.

Zerlegen wir beispielsweise die Anforderung aus dem Bibliothekssystem „Das Bibliothekssystem muss dem Bibliothekar die Möglichkeit bieten, die Daten eines neuen Kunden einzugeben und zu speichern.“, so resultieren hieraus folgende vereinzelte Anforderungssätze:

„Das Bibliothekssystem muss dem Bibliothekar die Möglichkeit bieten, die Daten eines neuen Kunden einzugeben.“

Das Bibliothekssystem muss dem Bibliothekar die Möglichkeit bieten, die eingegebenen Daten eines neuen Kunden zu speichern.“

Wie viele Prozesswörter (Verben) sind in der Anforderung enthalten?



## 6 Das SOPHIST-REgelwerk

Vorteile sind:  
Anforderungen können besser priorisiert, bewertet und nachvollzogen werden.

Nachteile sind:  
Zusammengehörige Informationen werden über mehrere Anforderungen verstreut und sind aufwändiger in der Verwaltung.

Die Vereinzelung hat nun dazu geführt, dass für jede geforderte Systemfunktionalität genau ein Satz gebildet wurde. Jeder Satz kann nun separat durch gezieltes Hinterfragen des enthaltenen Vollverbs systematisch analysiert werden (Regel 6).

Weitere Informationen zum Vorgehen bei der Vereinzelung von Anforderungen finden Sie auf unserer Homepage, [www.sophist.de](http://www.sophist.de).

### Schritt 4: Prüfe Unvollständigkeiten in Prozesswörtern

Um in einer Anforderung einen Prozess eindeutig zu beschreiben, ist es notwendig, dass alle zur vollständigen Erklärung des Prozesswortes (Vollverbs) erforderlichen Informationen in der Anforderung enthalten sind. Bleiben zum Prozess noch Fragen offen, so müssen die zugehörigen Informationen aufgedeckt und der Anforderung hinzugefügt werden.

**T**

#### Regel 6: Analysieren Sie fehlende Informationen zum Prozesswort.

Hinterfragen Sie das Prozesswort (Vollverb) der Anforderung mit den typischen W-Fragen. Wenn diese nicht alle durch die Informationen in der Anforderung beantwortet werden können, so wurde Information getilgt.

Ist die fehlende Information wissenswert, ergänzen Sie die Anforderung um die getilgte Information.

Um sprachliche Effekte in Anforderungen zu beheben, stellt der Requirements-Engineer gezielte Fragen rund um das Prozesswort. Die Antworten auf diese Fragen liefern Informationen, die zur Beseitigung eines Defekts notwendig sind. Nach unserer Erfahrung sollten zumindest folgende Fragen diskutiert und in der Anforderung beantwortet werden:

Objekt:  
An wem oder was wird die Funktionalität ausgeführt?

Häufigkeit:  
Wie oft wird die Funktionalität ausgeführt?

Methode:  
Wie wird die Funktionalität ausgeführt (nicht technologisch, sondern fachlich)?

Zeit/Logisch:  
Wann bzw. unter welchen Randbedingungen wird die Funktionalität ausgeführt?



Anhand der Fragen zum Prozesswort lässt sich feststellen, ob dieses ausreichend spezifiziert ist oder eine Anforderung um weitere Informationen ergänzt werden muss.



Für die Anforderung „Nicht zugelassene Kunden werden vom System angezeigt.“ ist beispielsweise das Vollverb „anzeigen“ zu hinterfragen.

Einige Fragen lassen sich aus der Anforderung heraus mehr oder weniger beantworten. Zum Beispiel lautet die Antwort auf die Frage „Was wird angezeigt?“ vermutlich: „eine Meldung über nicht zugelassene Kunden“. Hingegen bleiben die Fragen „Wem wird angezeigt?“, „Wie wird angezeigt?“, „Wann wird angezeigt?“ und „Wie oft wird angezeigt?“ in der Anforderung unbeantwortet. Hier handelt es sich um sprachliche Effekte, die genauer analysiert werden müssen. Sind die getilgten Informationen wesentlich, so muss die Anforderung um diese Informationen ergänzt werden.

Was?  
Wem?  
Wie?  
Wann?  
Wie oft?

Nachdem die Antworten auf die Fragen eingearbeitet wurden, könnte die verbesserte Anforderung folgendermaßen lauten: „Nachdem das Bibliothekssystem die Kundenkarte geprüft hat und falls der Kunde nicht zur Ausleihe des Leihobjektes zugelassen ist, muss das Bibliothekssystem dem Bibliothekar die Fehlermeldung „Kunde nicht zugelassen“ anzeigen.“

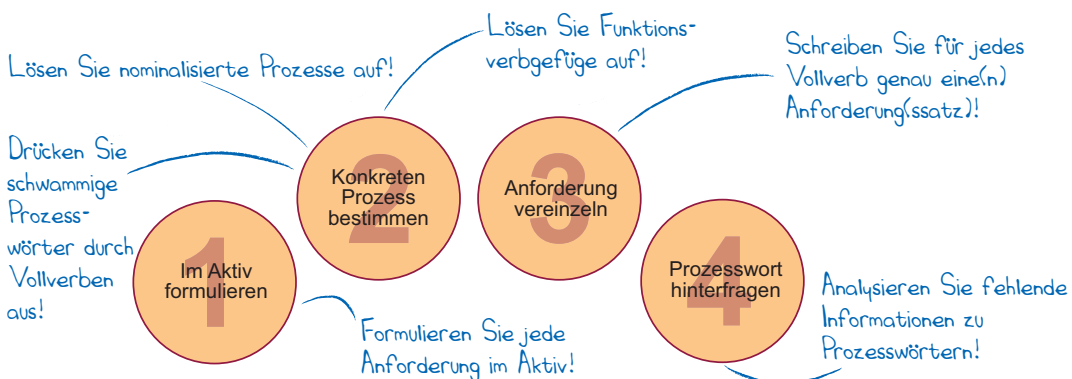
Ein weiteres Beispiel für eine effektbelastete Anforderung wäre: „Der Benutzer muss suchen können.“. Auch bei dieser Anforderung bleiben zu dem Vollverb „suchen“ eine Menge an Fragen offen, die durch die Anforderung beantwortet werden müssen. Nach was bzw. nach welchem Objekt wird gesucht? Wie oft kann gesucht werden? Nach welchen Kriterien wird gesucht? Wann bzw. unter welchen Randbedingungen wird gesucht?

Nach was?  
Wie oft?  
Nach welchen Kriterien?  
Wann?

Die Behebung der unvollständig spezifizierten Prozesswörter kann zu aufwändigen Umformulierungen der Anforderung führen. Häufig resultieren aus den Antworten zu den Fragen aber auch zusätzliche Anforderungen, oder man erkennt, dass die Originalanforderung durch mehrere detailliertere Anforderungen verfeinert und somit ersetzt werden muss.

### Die vier Schritte zum Prüfen des Prozesses

Fassen wir kurz zusammen. Entlang der in Abbildung 6.4 aufgezeigten Schritte haben Sie Ihre Anforderung(en) im Aktiv formuliert. Alle zu spezifizierenden Funktionalitäten sind durch Vollverben beschrieben, wobei alle offenen Fragen zum Prozesswort beantwortet und in die Anforderungen eingearbeitet sind.



**Abbildung 6.4:** Die vier Schritte zum Prüfen der Prozesse.

Dennoch befinden wir uns erst auf dem halben Weg zur perfekten Anforderung.



### 6.5.2 Prüfen von Eigenschaften



Ähnlich den Prozessbeschreibungen muss auch die Beschreibung von Eigenschaften eindeutig und vollständig sein.

Systemeigenschaften beziehen sich auf die Beschreibung nicht-funktionaler Aspekte des zu spezifizierenden Systems. In Anforderungen werden Eigenschaften in der Regel durch Adjektive und Adverbien ausgedrückt.

Eigenschaftswörter, wie z. B.  
groß, wichtig, verständlich, schnell

Umstandswörter, wie z. B.  
hier, gestern, anders

Für die Beschreibung von Systemeigenschaften reicht ein Adjektiv bzw. Adverb allerdings nicht aus. Eigenschaften müssen durch weitere Informationen ergänzt werden. Für das Adjektiv „sicher“ muss beispielsweise beschrieben werden, was vor wem oder was sicher sein muss, wer sichert und wie gesichert wird. Auch hier hilft Ihnen Ihr natürliches Sprachgefühl dabei, fehlende oder verzerrte Informationen aufzuspüren.

Bei der Prüfung der Adjektive liegt Ihr Hauptaugenmerk auf der in einer Anforderung beschriebenen Systemeigenschaft. Das zu identifizierende Signalwort ist ein Adjektiv oder Adverb, also ein Wort, durch das eine Eigenschaft oder ein Umstand beschrieben wird.

Achten Sie auf  
Signalwörter wie  
„schnell“,  
„performant“,  
„groß“, „gering“

#### Schritt 1: Analysiere geforderte Eigenschaften

Um die Eigenschaftsbeschreibung in einer Anforderung zu vervollständigen, müssen die für die Eigenschaft wesentlichen Informationen aufgedeckt und der Anforderung hinzugefügt werden (ähnlich Regel 6).

T

#### Regel 7: Analysieren Sie fehlende Informationen zum Eigenschaftswort.

Hinterfragen Sie das Eigenschaftswort (Adjektiv, Adverb) der Anforderung mit den typischen W-Fragen. Wenn diese nicht alle durch die Informationen in der Anforderung beantwortet werden können, so wurde Information getilgt.

Ist die fehlende Information wissenswert, ergänzen Sie die Anforderung um die getilgte Information.



Für wen  
oder was wird die Eigenschaft  
gefordert? Wann bzw. unter welchen  
Rahmenbedingungen? ...

Für wen soll das Bibliothekssystem intuitiv bedienbar sein? Was genau muss intuitiv bedienbar sein?

Für die Anforderung „Das Bibliothekssystem muss intuitiv bedienbar sein“ muss beispielsweise die Eigenschaft „intuitiv bedienbar“ hinterfragt werden. Die sprachlichen Effekte müssen genauer analysiert und wesentliche Informationen der Anforderung beigefügt werden. Eine verbesserte (aber lange noch nicht effektfreie) Anforderung wäre beispielsweise: „Die Benutzeroberfläche des Bibliothekssystems muss für den Bibliothekar intuitiv bedienbar sein.“

Richtig! „Intuitiv bedienbar“ hat vollen Interpretationsspielraum und muss quantifiziert werden (vgl. Regel 8).

Häufig finden wir nicht-funktionale Aspekte in funktionalen Anforderungen. Diese werden durch Adjektive beschrieben, die entweder als Attribut zu einem Nomen auftreten oder als Ergänzung zum Prozesswort (Adverb).

Z. B. etwas „schnell“ übertragen

Z. B. die „schnelle“ Übertragung

Um nicht-funktionale Aspekte in Anforderungen offenzulegen, extrahieren Sie diese zunächst aus der funktionalen Anforderung, um sie auch separat analysieren zu können.

Die Anforderung „Ein schnelles Ausdrucken des Benutzerausweises ist erforderlich.“ wurde durch Anwendung der Regeln 1 bis 6 beispielsweise folgendermaßen umformuliert: „Nachdem das Bibliothekssystem die Registrierungsdaten eines neuen Bibliothekskunden gespeichert hat, muss das Bibliothekssystem den Benutzerausweis des neu registrierten Bibliothekskunden schnell ausdrucken.“

Die Forderung „schnelles Ausdrucken“ weist darauf hin, dass der Ausdruckprozess ein bestimmtes Zeitverhalten aufweisen muss. Daraus könnte zunächst die nicht-funktionale Anforderung „Das Bibliothekssystem muss das Ausdrucken des Benutzerausweises schnell durchführen.“ generiert werden. Klarerweise bleibt an dieser Stelle die Frage nach dem konkreten Zeitverhalten noch unbeantwortet: „Was genau bedeutet schnell?“. Zur Analyse solcher sprachlicher Effekte haben wir eine eigene Regel formuliert.

Achten Sie auf Signalwörter wie „besser“, „schneller“, „leichter“.

Achten Sie auf Signalwörter wie „am schnellsten“, „größte“, „geringste“.

Diese Regel besagt, dass Adjektive oder Adverbien in Anforderungen häufig Vergleiche (Komparative) oder Steigerungen (Superlative) ausdrücken. Für einen Vergleich oder eine Steigerung muss als Zusatzinformation aber immer der Bezugspunkt konkret spezifiziert sein. Für den Vergleich „schneller“ muss beispielsweise angegeben werden, um wie viel etwas schneller als was sein muss oder wie schnell genau etwas sein muss.

RE-Bavarnregel:  
„Schneller, besser, schöner, breiter...“, frag stets „... als was?“, das wirkt gescheiter.

T

### Regel 8: Formulieren Sie Vergleiche und Steigerungen mess- bzw. testbar.

Vergleiche und Steigerungen benötigen zur vollständigen Beschreibung immer einen Bezugspunkt.

Hinterfragen Sie den Bezugspunkt des Vergleichs bzw. der Steigerung und ergänzen Sie die Anforderung um die getilgte Information.

Eine Abweichung von der Forderung muss natürlich auch messbar und damit überprüfbar (testbar) sein. Dies bedeutet, dass die Messmethode bekannt sein sollte, damit schon beim Erstellen der Anforderungen klar ist, ob die Anforderung anschließend wirklich getestet werden kann.

In Bezug  
bzw. Vergleich zu wem bzw. was?  
Wie kann die Erfüllung bzw. Abweichung gemessen werden?

In vielen Fällen ist es sinnvoll, sich bereits mit der Messgenauigkeit auseinanderzusetzen.



## 6 Das SOPHIST-Regelwerk

Für die Anforderung „Das Bibliothekssystem muss die Benutzerdaten sicher verwalten.“ müssen Sie beispielsweise den Bezugspunkt für die Eigenschaft „sicher“ hinterfragen und das gewonnene Wissen in die Überarbeitung der Anforderung einfließen lassen.

- Sicher in Bezug zu was bzw. vor wem?
- Was macht Benutzerdaten sicher?
- Nach welchen Kriterien kann sicher bei der Abnahme getestet werden?

Basierend auf den Antworten zu den Fragen könnte eine überarbeitete Anforderung dann folgendermaßen aussehen: „Das Bibliothekssystem muss so gestaltet sein, dass die in Standard X definierten Datenschutzrichtlinien eingehalten werden.“

Die Behebung der unvollständigen Vergleiche und Steigerungen gestaltet sich unterschiedlich schwierig. Im obigen Beispiel kann durch leichtes Modifizieren der Anforderung der Bezugspunkt hinzugefügt werden. Dagegen lässt sich im nächsten Beispiel die verwendete Phrase „intuitiv bedienbar“ nur schwer oder gar nicht messen: „Die Benutzeroberfläche des Bibliothekssystems muss für den Bibliothekar intuitiv bedienbar sein.“ Für diese Anforderung können aufwändige Umformulierungen erforderlich sein, die mit der Einführung neuer, messbarer Kriterien einhergehen. Zuerst muss allerdings einmal der Wunsch hinterfragt werden, der sich hinter der Formulierung „intuitiv bedienbar“ eines Stakeholders verbirgt.

- Intuitiv bedienbar in Bezug zu was?
- Was macht die Benutzeroberfläche intuitiv bedienbar?
- Welches Messkriterium kann bei der Abnahme angewendet werden?

Hier kann es hilfreich sein, Eigenschaften in paradoxer Weise zu hinterfragen, z. B.: Wie muss das System gestaltet sein, damit es NICHT „intuitiv bedienbar“ ist?

Sicherlich hatte der Schreiber dieser Anforderung einige sehr konkrete Sachverhalte im Sinn. Beispielsweise kann hinter einer derartigen Formulierung der Wunsch nach einem Hilfesystem stecken, welches bei jeder Benutzerinteraktion konsultiert werden kann und in mehreren Detaillierungsebenen Auskunft über mögliche Eingabewerte, Folgen und Ausnahmbedingungen der Aktion gibt. Dahinter kann sich aber auch die Forderung nach einer Menüsteuerung, einem interaktiven Lernprogramm, nach Spracherkennung, nach Mehrsprachigkeit oder nach der Konformität mit gängigen Oberflächenstandards verbergen. Vielleicht liegt hinter der Anforderung aber auch lediglich folgender Wunsch: „Das Bibliothekssystem muss so gestaltet sein, dass der Bibliothekar jeden Ausleihvorgang innerhalb von „5 Klicks“ ausführen kann.“

Es lohnt sich auf jeden Fall, dies zu erkunden, und derartig defektbelastete Anforderungen durch detailliertere zu ersetzen.

### Schritt 2: Prüfe Separierbarkeit nicht-funktionaler Anforderungen

Nicht unbedeutend, aber häufig kaum beachtet ist die Trennung zwischen funktionalen und nicht-funktionalen Anforderungen. In der Projektrealität erleben wir häufig unterschiedliche Varianten. In manchen Projekten werden beide Anforderungsarten ganz strikt getrennt, was die Frage aufwirft, ob bei der Abnahme dann auch wirklich alle Anforderungen eigenständig getestet werden können. In anderen Projekten dagegen finden wir Spezifikationen vor, in denen in derselben Anforderung sowohl funktionale als auch nicht-funktionale Aspekte beschrieben werden. Wie so häufig im Leben muss man auch hier den goldenen Mittelweg finden.

### Regel 9: Formulieren Sie eigene Anforderungen für nicht-funktionale Aspekte.

Trennen Sie nicht-funktionale Aspekte aus funktionalen Anforderungen, wenn mindestens eine der folgenden Bedingungen erfüllt ist:

- Der nicht-funktionale Aspekt ist eigenständig testbar, ODER
- Der nicht-funktionale Aspekt wird übergreifend als ein nicht-funktionales Constraint für mehrere Funktionalitäten gefordert.

Die Entscheidung, ob funktionale und nicht-funktionale Aspekte getrennt werden sollten, gestaltet sich unterschiedlich schwierig. Beispielsweise würde man die Anforderung „Das Bibliothekssystem muss die eingegebenen neuen Benutzerdaten innerhalb von maximal 1 Sekunde speichern.“ genau dann unverändert bestehen lassen, wenn die geforderte Zeitvorgabe nur für die spezifizierte Funktionalität, nämlich das „Speichern neuer Kundendaten“, gelten soll. Das Separieren der Zeitvorgabe in eine eigene Anforderung würde keinen Sinn ergeben, da Sie die Zeitvorgabe bei der Abnahme niemals testen könnten, ohne dabei auch die Funktionalität auszuführen.

Ist die Eigenschaft eigenständig testbar? Wird sie global gefordert?



Andererseits könnte es sein, dass sich während der Analyse des nicht-funktionalen Aspekts „innerhalb von maximal 1 Sekunde speichern“ herausstellen würde, dass diese Zeitvorgabe ein globales Constraint darstellt. Beispielsweise müssen auch Stammdaten neuer Leihobjekte oder Änderungen von Benutzerdaten dieser Forderung entsprechen. Formulieren Sie in diesem Fall eine separate Anforderung: „Das Bibliothekssystem muss jeden Speichervorgang innerhalb von maximal 1 Sekunde durchführen.“

### Die zwei Schritte zum Prüfen von Eigenschaften

Fassen wir weiter zusammen. Sie haben durch Einsatz der in Abbildung 6.5 aufgezeigten Schritte fehlende Informationen zu Eigenschaftsbeschreibungen aufgedeckt. Dabei haben Sie nicht-funktionale Aspekte (Eigenschaften) analysiert und qualifizierte Aussagen in messbare Anforderungen transformiert.

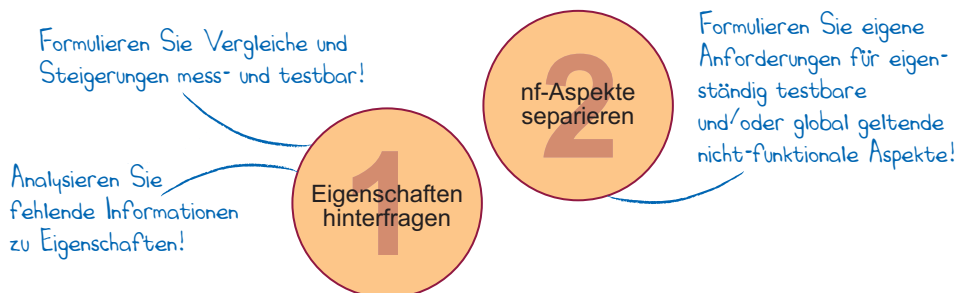


Abbildung 6.5: Die zwei Schritte zum Prüfen von Eigenschaften

### 6.5.3 Prüfen von Mengen und Häufigkeiten



Zur Vermeidung von Redundanzen formuliert man nicht für jede kleinste Einheit die eigentlich gleiche Anforderung mehrfach, sondern beschreibt ein Verhalten oder eine Eigenschaft nur einmal und weist dieser Beschreibung eine Gruppe von Sachverhalten zu, für die das Spezifizierte gültig sein soll. Wichtig ist allerdings, dass die Sachverhalte richtig zusammengefasst sind (siehe Transformationseffekt „Generalisierung“ in Abschnitt 6.2.2).

Z. B. „Kundenname speichern“, „Kundenadresse speichern“, ... lässt sich durch „Registrierungsdaten eines Kunden speichern“ zusammenfassen.

Der Linguist sagt hierzu „Universalquantoren“ oder einfach nur „Quantoren“.

Signalwörter, die auf eine Generalisierung von Wissen hinweisen, sind Zahl- und Mengenwörter sowie Substantive, durch die Akteure oder Objekte zusammengefasst werden. Die Prüfung von Mengen- und Häufigkeitsangaben zielt nicht darauf, Generalisierungen zu vermeiden. Eher geht es darum zu hinterfragen, ob ein Stakeholder eine Gruppe von Sachverhalten richtig zusammengefasst hat.

#### Schritt 1: Prüfe Zahl- und Mengenwörter

RE-Bauernregel:  
Sei allzeit bereit für  
Quantoren wie „alle,  
keiner, jeder, nie ...“,  
doch Obacht: Prüf  
sie erst, dann  
verwende sie.

Durch Anforderungen werden Funktionalitäten oder Eigenschaften spezifiziert, die für eine bestimmte Menge an Objekten gültig sein sollen. Hierzu verwendet man in der natürlichen Sprache Zahl- oder Mengenwörter. In einer Anforderung wird dann eine Aussage getroffen über das Verhalten oder die Eigenschaft dieser Menge von Objekten.

Achten Sie auf Signalwörter wie „alle“, „jeder/jeden“, „immer“, „kein“, ...

Die Anforderung „Das Bibliothekssystem muss es jedem Bibliothekskunden ermöglichen, jedes Leihobjekt auszuleihen.“ würde faktisch zu der Annahme führen, dass auch die jüngeren Bibliothekskunden solche Leihobjekte ausleihen dürfen, die laut Jugendschutzgesetz nicht an diese ausgegeben werden dürfen. Weiterhin können sich im Portfolio der Bibliothek auch Leihobjekte befinden, die nur zur Ansicht, nicht aber zur Ausleihe gedacht sind. Dass solche Leihobjekte dann tatsächlich an alle Bibliothekskunden ausgeliehen werden können, ist sicherlich nicht im Sinne der Bibliotheksverwaltung. An dieser Stelle hat der Bibliothekar falsch generalisiert.

Die Gefahr bei der Verwendung von Zahl- oder Mengenwörtern besteht also darin, dass das spezifizierte Verhalten bzw. die Eigenschaft nicht für alle Objekte der bezeichneten Menge zutrifft. Häufig sind Elemente in der Menge enthalten, die einen Sonder- oder Ausnahmefall darstellen und für die das spezifizierte Verhalten falsch ist. Hier ist es Ihre Aufgabe als Requirements-Engineer, solche Ausnahmen durch gezieltes Hinterfragen der verwendeten Mengen- und Häufigkeitsangaben aufzudecken.

Hinterfragen Sie Zahl- oder Mengenwörter sehr gezielt, denn wenn Sie bei jeder Anforderung „Wirklich alle“, „wirklich jeder“ ... hinterfragen, laufen Sie Gefahr, dass Ihre Stakeholder schnell die Geduld verlieren.

G

**Regel 10: Hinterfragen Sie verwendete Zahl- und Mengenwörter.**

Jedes geforderte Verhalten bzw. jede geforderte Eigenschaft muss für wirklich alle Objekte der angegebenen Menge und nur für die Objekte der angegebenen Menge gelten.

Prüfen Sie die verwendeten Zahl- oder Mengenwörter der Anforderung. Wenn falsch zusammengefasst wurde, müssen Sie:

- Die Menge der Objekte einschränken, wenn nur ein Teil der Menge betroffen ist, ODER
- Die Menge der Objekte erweitern, wenn zusätzliche Objekte betroffen sind.

Häufig gibt es auch Ausnahmen, die Sie zusätzlich spezifizieren müssen.

In der Anforderung „Das Bibliothekssystem muss es jedem Benutzer ermöglichen, alle Benutzerdaten zu ändern.“ sollten die Mengenangaben „jedem Benutzer“ und „alle Benutzerdaten“ hinterfragt werden.

Gilt das Verhalten/die Eigenschaft für wirklich alle Objekte der Menge? Oder gibt es auch Ausnahmen?



Muss wirklich jeder Benutzer die Benutzerdaten ändern können?

Muss jeder Benutzer wirklich alle jemals im System abgespeicherten Benutzerdaten ändern können?

Falls sich bei der Beantwortung der Fragen herausstellen sollte, dass es Ausnahmefälle gibt, so müssen diese spezifiziert und die Ausgangsanforderung geändert werden: „Das Bibliothekssystem muss es jedem Bibliothekskunden ermöglichen, die über ihn gespeicherten Registrierungsdaten zu ändern.“

Auch Sätze, die keine explizite Mengen- oder Häufigkeitsaussage für das spezifizierte Verhalten oder die Eigenschaft angeben, sollten überprüft werden. Fehlende Angaben hierzu enthalten die implizite Annahme, dass das angegebene Verhalten für alle überhaupt relevanten Objekte immer gelten soll. Diese Annahme kann durchaus korrekt sein. Als Requirements-Engineer laufen Sie aber Gefahr, dass Ausnahmen übersehen werden.

G

**Regel 11: Klären Sie fehlende Zahl- und Mengenwörter.**

Für jedes geforderte Verhalten bzw. jede geforderte Eigenschaft muss die Menge der Objekte, für die das spezifizierte Verhalten bzw. die spezifizierte Eigenschaft gelten soll, explizit beschrieben sein.

Prüfen Sie, ob der Anforderung Zahl- oder Mengenwörter hinzugefügt werden können. Ist dies der Fall, müssen Sie prüfen, für welche Objekte genau die Anforderung gültig sein soll. Ergänzen Sie in der Anforderung das zugehörige Zahl- oder Mengenwort, falls erforderlich.



## 6 Das SOPHIST-Regelwerk



Für welche  
Objekte genau soll das  
geforderte Verhalten bzw. die  
Eigenschaft gelten?

Die Beispielanforderung „Das Bibliothekssystem muss es dem Benutzer ermöglichen, die gespeicherten Daten auf Band zu sichern.“ enthält beispielsweise keine expliziten Mengen- und Häufigkeitsangaben. Die Interpretation der Anforderung wäre daher, dass „jeder Benutzer jederzeit alle gespeicherten Daten auf jedes Band sichern kann“.

Um aber sicherzustellen, dass die Angaben in dieser Anforderung inhaltlich korrekt sind, müssen die Mengen- und Häufigkeiten konkret hinterfragt und ggf. hinzugefügt werden.

Darf jeder Benutzer die Sicherung einleiten? Über alle jemals gespeicherten Daten?  
Kann die Sicherung immer geschehen? Also auch mehrfach parallel?

Der Realisierungsrahmen der oben definierten Anforderung liegt zwischen einer trivialen Sicherungssoftware für 1.000 €, die den Bibliothekar seine Daten auf Band kopieren lässt, und einer vollautomatischen Robotik-Sicherungsstation für mehrere Millionen Euro. Einen derartigen Sachverhalt sollte man dann vermutlich doch klarstellen, bevor das System beauftragt wird, sofern man bei seinem Auftragnehmer oder seiner Entwicklungsabteilung keine hellseherischen Fähigkeiten voraussetzt.

Um die Eindeutigkeit zu gewährleisten und das Schreiben von Anforderungen zu erleichtern, empfiehlt es sich, die Menge der verwendeten Mengen- und Zahlwörter auf eine Anzahl eindeutig definierter Quantoren zu reduzieren.

Für die obige Beispielanforderung könnte die Analyse zu folgender Verbesserung führen: „Das System muss jedem Bibliothekar jederzeit die Möglichkeit bieten, alle im Bibliothekssystem aufgezeichneten Benutzer- und Leihobjektdaten als Sicherung auf Band zu speichern.“

Verwenden Sie  
beispielsweise nur:  
„alle“, „jeder/jeden“,  
„immer“, „kein“ ...  
oder „all“, „every“,  
„each“, „none“ ...

Linguisten  
sprechen hier von  
„Substantiven ohne  
oder mit zu wenig  
Bezugsindex“.

Achten Sie auf  
Signalwörter wie  
„der Anwender“,  
„die Meldung“,  
„die Daten“,  
„die Funktion“...

### Schritt 2: Prüfe Unvollständigkeiten in Substantiven

In einer Anforderung werden durch Substantive (Hauptwörter) sowohl Akteur(e) als auch Objekte repräsentiert, für die ein Verhalten bzw. die Eigenschaft gefordert wird. Häufig finden wir in Anforderungen aber Substantive, die Akteure oder Objekte so schwammig beschreiben, dass zu ihrer eindeutigen Spezifikation weitere Informationen erforderlich sind.

Durch das Substantiv „Werte“ könnten beispielsweise alle Datenwerte zusammengefasst werden, die in irgendeiner Weise im System vorhanden sind und eingegeben werden können. In der Regel beschränkt sich eine Anforderung aber nur auf einen Teil dieser „Werte“. Zur vollständigen Erklärung muss also genau angegeben werden, auf welche „Werte“ genau sich die Anforderung bezieht.



#### Regel 12: Hinterfragen Sie schwammige Substantive.

Beschreibt ein Substantiv eine nicht genau einzugrenzende Menge von Objekten, so wurde aus dem Originalsatz eine Information getilgt.

Hinterfragen Sie schwammig formulierte Substantive und stellen Sie fest, für welche Objekte oder Akteure genau diese Anforderung gelten soll. Ergänzen Sie dann die Anforderung um die getilgte Information.

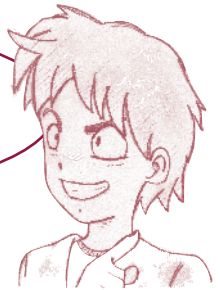


Der Bibliothekar berichtet beispielsweise: „Die Daten müssen dem Anwender grafisch dargestellt werden.“ In dieser Anforderung müssen zumindest die Substantive „Daten“ und „Anwender“ detaillierter analysiert werden.

Wer/Was ... genau? Welcher Teil der genannten Menge?

Welchen Anwendern genau?

Welche Daten genau?



Die Analyse der Substantive könnte beispielsweise ergeben, dass sich „Daten“ auf „statistisch berechnete Daten der Leihobjekte“ beziehen und der Akteur „Anwender“ auf den „Bibliothekar“ eingeschränkt wird. Eine verbesserte Anforderung zu dem obigen Beispiel wäre dann: „Nachdem der Bibliothekar die Berechnung der Leihobjektstatistik initiiert hat, muss das Bibliothekssystem dem Bibliothekar alle statistisch berechneten Daten der Leihobjekte grafisch anzeigen.“

Diese Umformulierung setzt natürlich voraus, dass es weitere Anforderungen gibt, die den Prozess „Berechnung der Leihobjektstatistik“ detailliert beschreiben und dass auch wirklich alle statistisch berechneten Daten der Leihobjekte grafisch dargestellt werden können.

Was genau bedeutet „grafisch“?  
Welche Daten der Leihobjekte genau?

Des Weiteren muss natürlich auch das Adjektiv „grafisch“ präziser beschrieben werden. Häufig resultiert aus diesen Fragen eine Verfeinerung der Anforderung, beispielsweise dann, wenn aus den Antworten hervorgeht, dass für die unterschiedlichen statistischen Leihobjekt-daten auch unterschiedliche grafische Darstellungen gefordert werden.

## Die zwei Schritte zum Prüfen von Mengen und Häufigkeiten

Nun haben Sie es bald geschafft. Nachdem Sie die in Abbildung 6.6 aufgezeigten Schritte absolviert haben, haben Sie sprachliche Effekte in Bezug auf Mengen- und Häufigkeitsangaben identifiziert und behoben. Hierzu haben Sie Zahl- und Mengewörter sowie schwammige Substantive dahingehend geprüft, ob Objekte oder Akteure durch diese richtig zusammengefasst wurden.

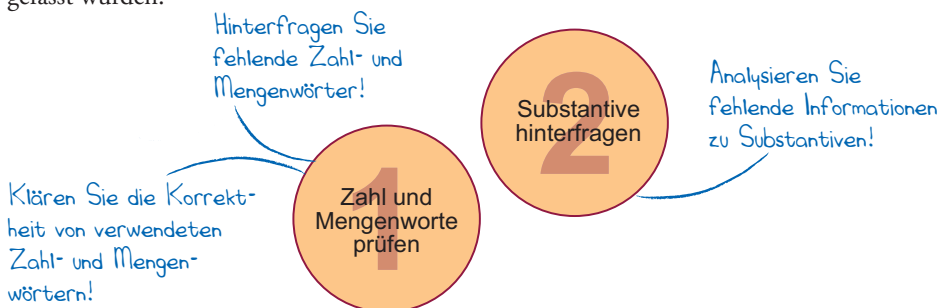


Abbildung 6.6: Die zwei Schritte zum Prüfen von Mengen und Häufigkeiten

Nun sind Sie nur noch wenige Schritte von der nahezu „perfekten“ Anforderung entfernt.

### 6.5.4 Prüfen von Begriffen, die Möglichkeiten beschreiben



Oft ist es nicht nur notwendig, eine Funktion des Systems durch eine Anforderung einfach zu fordern, sondern auch, den Weg zu beschreiben, wie das System die Forderung erfüllen soll und welche Mittel dazu verwendet werden sollen. Dies gilt insbesondere für fachlich komplexe Abläufe, die so genannte Geschäftslogik.

Im Englischen: Business Logic

Ups, wie soll ein Implementierer, der nicht nebenbei noch Flugsicherungs-  
experte ist, wissen, wie diese zu berechnen ist?

#### Schritt 1: Prüfe Mögliches und Unmögliches

Wir erleben häufiger, dass Anforderungen nur die Ergebnisse komplexer Berechnungen spezifizieren, zum Beispiel: „... Bei jedem Radar-Update muss die Flugplanroute neu berechnet werden.“ Oftmals reicht es einfach nicht aus, nur zu fordern, dass gewisse Ergebnisse ermittelt werden, sondern die Anforderungsspezifikation muss auch die Details enthalten, welche fachliche Geschäftslogik hinter der Ermittlung bestimmter Ergebnisse steht. Signalwörter, die auf eine getilgte Fachlogik hinweisen, sind Begriffe, die ausdrücken, dass etwas möglich oder unmöglich sein soll.

Der Linguist sagt hierzu  
„Modaloperatoren der Möglichkeit“

Achten Sie auf Signalwörter wie  
„kann“, „darf“, „es ist (nicht) möglich“,  
„er/sie/es ist außerstande“.

Hinterfragen Sie die fachliche Logik, nicht die technische Lösung!

Achten Sie immer darauf, dass die Beschreibung der fachlichen Logik (die unweigerlich zu der Fragestellung „Wie“ führt) nicht gleichzusetzen ist mit der Beschreibung von Implementierungsdetails. Insbesondere bei der Formulierung fachlicher Anforderungen (Spezifikationslevel 2 und 3) sollten Implementierungsdetails (wie beispielsweise der Einsatz einer bestimmten Datenbank, ...) nicht beschrieben werden, außer wenn diese auch ausdrücklich gefordert werden (zum Beispiel aus Wartungsgründen).

T

#### Regel 13: Klären Sie Mögliches und Unmögliches.

Aussagen, die nur beschreiben, dass ein Verhalten möglich oder unmöglich sein soll, tilgen die mit der Forderung verbundene fachliche Logik.

Hinterfragen Sie, was das geforderte Verhalten oder die Eigenschaft möglich oder unmöglich macht und ergänzen Sie die Anforderung um die identifizierte fachliche Logik.

Durch Hinterfragen von Möglichem und Unmöglichem finden Sie vergangene Vorbedingungen.



Was macht das Verhalten möglich/unmöglich?  
Welche fachliche Logik steckt dahinter?  
Welche Vorbedingung muss gelten?

Wenn in einer Anforderung Begriffe verwendet werden, die ausdrücken, dass etwas möglich oder unmöglich sein soll, können Anforderungen fast immer in das sprachliche Konstrukt „Etwas verhindert bzw. macht es möglich, dass etwas anders möglich oder unmöglich ist.“ umformen. In der Anforderung eines Stakeholders wird häufig aber nur der letztere Satzteil expliziert. Der vordere Satzteil (der Begründungssatz) bleibt dann unbeantwortet. Dann wurde die Voraussetzung, die das geforderte Verhalten überhaupt erst möglich oder unmöglich macht, getilgt. Prüfen Sie deshalb immer, ob alle relevanten Voraussetzungen bzw. Vorbedingungen notiert sind.

Hierzu eine Beispielanforderung aus dem Bibliothekssystem: „An einen Bibliothekskunden, für den noch mindestens zwei offene Mahnungen vorliegen, darf kein weiteres Buch ausgeliehen werden.“ Zur Umsetzung dieser Anforderung muss das System auf eine Information zugreifen können, die über die offenen Mahnungen eines Bibliothekskunden Auskunft gibt. Aber woher kommt diese Information?

Wodurch wird dies unmöglich?

Woher erhält das System das erforderliche Wissen?

Die Antworten auf diese Fragen können, falls noch nicht notiert, zusätzliche Anforderungen erfordern. Durch diese wird überhaupt erst die Voraussetzung für die obige Anforderung expliziert, nämlich die Verwaltung einer Mahnhistorie für jeden Bibliothekskunden: „Sobald der Rückgabetermin für ein Leihobjekt erreicht ist, muss das Bibliothekssystem für den Bibliothekskunden, der das Leihobjekt ausgeliehen hat, die Anzahl der offenen Mahnungen in der Mahnhistorie des Bibliothekskunden um 1 erhöhen“. Basierend auf dieser Funktionsbeschreibung wird nun klarer, wie das System beim Entleihvorgang die Anzahl noch offener Mahnungen ermitteln kann: „Nachdem der Bibliothekar das Ausleihen eines Leihobjektes für einen Bibliothekskunden initiiert hat und falls für den Bibliothekskunden noch mindestens zwei offene Mahnungen in seiner Mahnhistorie existieren, muss das System dem Bibliothekar die Fehlermeldung „noch offene Mahnungen“ anzeigen.“

Auch die Bibliothekssystem-Anforderung „Es darf nicht möglich sein, dass ein neuer Bibliothekskunde registriert wird, der jünger als 18 Jahre ist und keine Einverständniserklärung der Eltern vorlegen kann.“ sollte durch die Zusatzinformation ergänzt werden, wie das System diese Forderung erfüllen soll. Sicherlich hat der Stakeholder eine sehr konkrete Vorstellung davon, wie die Neuregistrierung eines Kunden geregelt werden soll. Eine verbesserte Anforderung wäre daher: „Nachdem der Bibliothekar das Speichern neuer Kundendaten initiiert hat und falls der neu eingegebene Kunde ein Alter von unter 18 Jahren hat und falls keine Einverständniserklärung vorliegt, muss das Bibliothekssystem dem Bibliothekar die Fehlermeldung „Kunde unter 18“ anzeigen.“

Durch wen oder was wird das verhindert?

Welche fachliche Logik steckt dahinter?

Diese Aussage ist schon ein Stück klarer, doch nach wie vor mit Effekten behaftet. Die Zusatzinformation, wie das Bibliothekssystem wissen kann, ob eine Einverständniserklärung vorliegt oder nicht, führt uns natürlich zur wiederholten Anwendung dieser Regel und wahrscheinlich zu folgender Anforderung: „Falls der neu eingegebene Kunde ein Alter von unter 18 Jahren hat, muss das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, das Vorhandensein einer Einverständniserklärung zu bestätigen.“

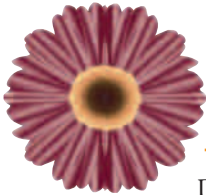
## Der Schritt zum Prüfen von Möglichkeiten und Unmöglichkeiten

Nachdem Sie nun den in Abbildung 6.7 aufgezeigten Schritt durchgeführt haben, haben Sie der Anforderung noch die fehlende fachliche Logik hinzugefügt (falls dies erforderlich war) und sie von wesentlichen sprachlichen Defekten bereinigt.



Abbildung 6.7: Ein Schritt zum Prüfen von Möglichkeiten und Unmöglichkeiten

### 6.6 Prüfen des Satzes



Nach der Prüfung anhand einzelner Satzbestandteile wenden wir uns jetzt ganzen Anforderungssätzen zu.

#### Schritt 1: Extrahiere unwesentliche Nebensätze

Die einzigen Nebensätze, die für eine Anforderung notwendig sind, sind logische oder zeitliche Bedingungssätze (siehe dazu auch Kapitel 7 „Schablonen“).

Achten Sie auf Signalwörter wie „weil“, „damit“, „um“, „deshalb“, „so dass“...

Nebensätze, die eine Begründung, Absicht oder Folge enthalten, sollten aus der eigentlichen Anforderung eliminiert und als Kommentar notiert werden. Sie sind für die eigentliche Systembeschreibung überflüssig und führen nur zu Verwirrungen und im schlimmsten Fall zu Missverständnissen.

#### Regel 14: Extrahieren Sie unbedeutende Informationen.

Nebensätze, die eine für die Systembeschreibung unwesentliche Information beschreiben, wirken sich negativ auf die Lesbarkeit aus und lassen die konkret geforderte Systemanforderung in den Hintergrund treten.

Prüfen Sie den Informationsgehalt eines Nebensatzes. Ist die Information für die Systembeschreibung unwesentlich, dann:

- Lösen Sie den Nebensatz aus der Anforderung heraus UND
- Notieren sie ihn als „Kommentar“ zur Anforderung.



Welche Relevanz hat die Information im Nebensatz?

Die folgende Beispielanforderung aus unserem Bibliothekssystem zeigt eine Anforderung, wie wir sie nur allzu häufig in Spezifikationen vorfinden: „Um dem Bibliothekar das Erstellen einer Leihobjekt-Statistik zu erleichtern, muss das Bibliothekssystem die Verwendung eines virtuellen Assistenten anbieten.“

Das ist die fachliche Begründung für die Anforderung – aber eine unwesentliche Information für die Systembeschreibung.

Obwohl der Nebensatz „Um ... zu erleichtern“ einen Sachverhalt beschreibt, der beispielsweise zur Entscheidungsfindung für die Umsetzung der Anforderung wesentlich sein kann, stellt er keine geforderte Funktionalität des Systems dar. Dieser Nebensatz sollte daher als Kommentar aus der Anforderung herausgezogen werden.

Die obige Anforderung lautet dann umformuliert: „Das Bibliothekssystem muss dem Bibliothekar die Möglichkeit bieten, eine Leihobjekt-Statistik mit Hilfe eines virtuellen Assistenten zu erstellen.“ Der verbleibende Nebensatz, durch den lediglich die Begründung für die Anforderung beschrieben wird, wird als Kommentar zur entsprechenden Anforderung formuliert: „Kommentar: Dadurch wird dem Bibliothekar das Erstellen der Leihobjekt-Statistik erleichtert.“

Falls sich allerdings etwas für den Systemkontext Wesentliches hinter diesem Nebensatz verbergen sollte, muss dieser Sachverhalt als eine separate Anforderung notiert werden.

## Schritt 2: Beseitige redundante Informationen

Oft ist uns gar nicht bewusst, dass wir Sätze bilden, in denen redundante Informationen stecken. Für die verbale Kommunikation zwischen Menschen ist dies oft erheiternd oder sinnvoll, Anforderungen jedoch werden dadurch unnötig kompliziert.

### Regel 15: Vermeiden Sie redundante Informationen.

Entfernen oder kürzen Sie alle Teile des Satzes, die Sie ohne Bedeutungsverlust straffen können:

- Redundante Informationen
- Floskelhafte Wörter und Wendungen

klein in der Größe => klein  
 gelb in der Farbe => gelb  
 eine wahre Tatsache => eine Tatsache

Wird etwas doppelt oder mit Floskeln ausgedrückt?



Viele Anforderungen können Sie ohne Informationsverlust straffen. Die Anforderung aus dem Bibliothekssystem: „Für den Fall, dass das Bibliothekssystem zusammen mit dem Bestellservers interoperiert, muss die Funktionsfähigkeit des Bibliothekssystems für die Benutzer mit der gleichen Funktionalität erhalten bleiben.“ lässt sich beispielsweise durch folgende Formulierung straffen: „Solange das Bibliothekssystem mit dem Bestellservers interoperiert, muss das Bibliothekssystem die uneingeschränkte Funktionsfähigkeit für den Benutzer sicherstellen.“

## Die beiden Schritte zum Prüfen des Satzes

Herzlichen Glückwunsch. Nachdem Sie die in Abbildung 6.8 aufgezeigten Schritte durchgeführt haben, haben Sie nicht nur sämtliche semantischen Effekte in Ihrer Anforderung erkannt und behoben, sondern die Anforderung auch von unnötigem Ballast befreit.

Klären Sie die Relevanz von Nebensätzen!

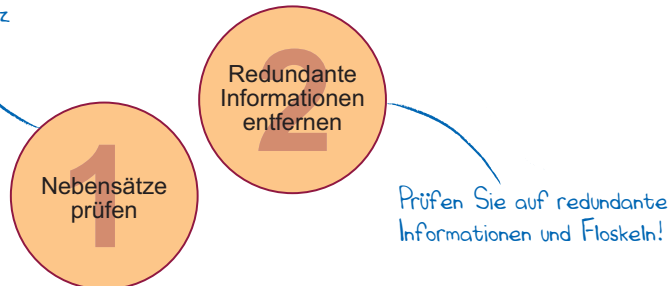


Abbildung 6.8: Die zwei Schritte zum Prüfen des Satzes

### 6.7 Prüfen des Gesamtbildes



So, nun begeben wir uns noch eine Stufe höher und betrachten nicht nur den einzelnen Satz, sondern dessen Integration in die Gesamtspezifikation.

#### Schritt 1: Prüfe Ausnahmen vom Normalverhalten

Um die Funktionalität eines Systems vollständig zu beschreiben, müssen Anforderungen sowohl das gewünschte Normalverhalten als auch das Verhalten im Ausnahmefall – sofern dieser auftreten kann – beschreiben. Beim Schreiben einer Anforderung sollten Sie sich grundsätzlich fragen, ob das System das geforderte Verhalten immer sicherstellen kann oder ob es Randbedingungen gibt, unter denen dies nicht möglich ist.

Der Linguist sagt hierzu „Modaloperatoren der Notwendigkeit“.

Leider sind Systeme in der Realität von äußeren Einflüssen abhängig, so werden zum Beispiel Daten und Ereignisse für die Weiterverarbeitung erwartet, die eventuell ausbleiben können. Beschreiben Sie das Verhalten des Systems in derartigen Ausnahmefällen, die es nicht selbst beheben kann. Solche Anforderungen erkennen Sie häufig an Begriffen, die ausdrücken, dass etwas notwendig sein muss.

Achten Sie auf Signalwörter wie „müssen“, „sollen“, „sollte“, „es ist notwendig“, ...

Wenden Sie diese Suchmethode verstärkt auf die frei formulierten Anforderungen an.

Für Anforderungen, die mit Hilfe unserer Anforderungsschablonen formuliert wurden (vgl. Kapitel 7 „Schablonen“), die also bewusst solche Modaloperatoren als Schlüsselwörter zur Festlegung der rechtlichen Verbindlichkeit enthalten, wird die Suche nach Modaloperatoren natürlich erfolgreich ausfallen.

T

#### Regel 16: Klären Sie Ausnahmen vom Normalverhalten.

Jedes Normalverhalten benötigt häufig auch die Beschreibung dessen, was passieren soll, wenn das System das Verhalten nicht sicherstellen kann.

Klären Sie, ob es Situationen gibt, in denen das System das Normalverhalten nicht gewährleisten kann. Ist dies der Fall, dann:

- Beschreiben Sie entweder das Ausnahmeverhalten durch eine zusätzliche Anforderung, ODER
- Erweitern Sie die bestehende Anforderung um das Ausnahmeverhalten.



Kann das System das geforderte Verhalten immer sicherstellen? Was passiert, wenn nicht? Welche Ausnahmen sind möglich?

Die Anforderung „Das Bibliothekssystem muss Bestellungen für neue Leihobjekte an das Versandsystem versenden.“ fordert ein Verhalten, welches den Normalfall betrachtet, dass Daten an ein Nachbarsystem geliefert werden müssen. Einige Fragen bezüglich möglicher Ausnahmefälle bleiben jedoch offen. Was passiert, wenn die Verbindung zum Versandsystem nicht hergestellt werden kann und



somit keine Daten versendet werden können? Was passiert, wenn während der Datenübertragung Fehler aufgetreten sind?

Welches Verhalten wird also vom System erwartet, wenn die geforderte Funktionalität systemisch nicht geleistet werden kann? Soll es einfach kommentarlos abstürzen? Wahrscheinlich erwarten Sie eher, dass es ein definiertes Verhalten aufweist. Zum Beispiel sollen mögliche Probleme während der Datenübertragung dem Benutzer über die Benutzungsschnittstelle angezeigt werden, so dass dieser auch eine Neuübertragung veranlassen kann. Nach Beantwortung der Fragen kann für die Anforderung das Verhalten im Ausnahmefall spezifiziert werden: „**Falls das Bibliothekssystem aus technischen Gründen keine Verbindung zum Versandsystem herstellen kann, muss das System dem Bibliothekar die Fehlermeldung „Verbindung kann nicht hergestellt werden“ anzeigen.**“

Die Beschreibung von Anforderungen in Ausnahmesituationen wird häufig vernachlässigt. Eine vollständige Definition des Systemverhaltens in Ausnahmefällen kann den identischen Aufwand bedeuten wie die Definition des Systemverhaltens für die Normalfälle. Das wird gerne unterschätzt.

Kalkulieren Sie diesen Aufwand daher frühzeitig mit ein. Wir erleben es in unserer Projektpraxis nur allzu häufig, dass die Definition des Verhaltens in Ausnahmesituationen aus Zeitgründen unterlassen oder nur angerissen wird. Im schlimmsten Falle resultiert dann ein System, welches bei minimalen Störungen der Umgebung ein unkalkulierbares Verhalten aufweist.

Spezifizieren Sie mögliche Ausnahmefälle sofort bei der Ermittlung der Benutzeranforderungen (ab Spezifikationslevel 2).

### Schritt 2: Prüfe unvollständig spezifizierte Bedingungen

Ähnlich der Spezifikation von Ausnahmefällen (siehe vorhergehende Regel), die das Systemverhalten in Situationen beschreiben, in denen das System das geforderte Verhalten nicht sicherstellen kann, müssen auch alle (häufig komplexen) Bedingungsstrukturen des „Normalverhaltens“ spezifiziert werden.

Anforderungen, die Bedingungen enthalten, geben das Verhalten bei Gültigkeit bzw. Eintritt der Bedingung an. Wichtig ist, dass dann auch beschrieben ist, was passieren soll, wenn die Bedingung nicht eintritt. Nach unserer Erfahrung wird aber letztere Information bei der Systemspezifikation häufig vernachlässigt.

Achten Sie auf Signalwörter wie „wenn ... dann“, „falls ... dann“, „im Falle von“ und „abhängig von“.



#### Regel 17: Analysieren Sie unvollständige Bedingungsstrukturen.

Jedes bedingte Verhalten benötigt zumindest auch die Beschreibung dessen, was passieren soll, wenn die Bedingung nicht eintritt.

Klären Sie das Systemverhalten für den Fall (oder die Fälle), dass die Bedingung nicht erfüllt ist:

- Spezifizieren Sie für jede noch nicht beschriebene Bedingung eine zusätzliche Anforderung, ODER
- Erweitern Sie die bestehende Anforderung um die fehlenden Fälle.

## 6 Das SOPHIST-REgelwerk



Wie soll sich das System verhalten, wenn die Bedingung nicht eintritt? Gibt es noch weitere Fälle?

Welches Systemverhalten wird für den Fall gefordert, dass das Leihobjekt reserviert ist?

Bei der Anforderung aus dem Bibliothekssystem: „Falls ein Leihobjekt nicht reserviert ist, muss das Bibliothekssystem dem Bibliothekar die Fortsetzung des Ausleihprozesses ermöglichen.“ bleibt mindestens die Frage nach dem Systemverhalten im Falle eines reservierten Leihobjekts unbeantwortet.

Für unser Bibliothekssystem würde die Klärung der Frage nach dem Systemverhalten im Falle eines „reservierten Leihobjektes“ nur dann zu einer Erweiterung führen, wenn das entsprechende Systemverhalten bisher noch nicht beschrieben wurde. In diesem Fall muss das Systemverhalten für den Nichteintritt der Bedingung, beispielsweise durch eine zusätzliche Anforderung, beschrieben werden: „Falls das Leihobjekt reserviert ist, muss das Bibliothekssystem dem Bibliothekar eine Fehlermeldung anzeigen.“

Schreiben Sie separate Anforderungen, wenn es viele zu unterscheidende Fälle gibt. Ansonsten erweitern Sie einfach die bestehende Anforderung.

Alternativ können Sie natürlich auch die bestehende Anforderung ergänzen:

- „Nachdem das Bibliothekssystem die Entleihbarkeit des Leihobjektes geprüft hat und falls das Leihobjekt nicht reserviert ist, muss das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, den Ausleihprozess fortzusetzen.“
- „Falls das Leihobjekt reserviert ist, muss das Bibliothekssystem dem Bibliothekar die Fehlermeldung „Leihobjekt reserviert“ anzeigen.“

Wichtig ist vor allem, dass Sie alle Fälle unterscheiden und das geforderte Systemverhalten auch für alle Fälle beschreiben. Leider lassen sich die Fallunterscheidungen nicht immer einfach nur in zwei Kategorien (erfüllt/nicht erfüllt bzw. reserviert/nicht reserviert) einteilen. Häufig kommt es vor, dass eine genauere Fallunterscheidung getroffen werden muss.

Als Beispiel prüfen wir eine Anforderung aus unserem Bibliothekssystem: „Nachdem das Bibliothekssystem den Bibliothekskunden identifiziert hat und falls für den Bibliothekskunden höchstens eine offene Mahnung existiert oder falls für den Bibliothekskunden zum Zeitpunkt der Ausleihe weniger als 24 noch ausgeliehene Leihobjekte registriert sind, muss das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, das zu entleihende Leihobjekt anzugeben.“

Welches Systemverhalten wird für den Fall gefordert, dass für den Bibliothekskunden mindestens zwei Mahnungen existieren?

Welches Systemverhalten wird für den Fall gefordert, dass der Bibliothekskunde mindestens 25 Leihobjekte ausgeliehen und noch keines zurückgegeben hat?

Für diese Anforderung müssen gleich mehrere Fragen geklärt und im Anforderungsdokument beschrieben werden. Die Analyse dieser Fallunterscheidung kann zu sehr komplexen Anforderungen führen. Wie komplex Anforderungen sein können, zeigt sich, wenn beim letzten Beispiel noch die Bedingungen hinzukommen „Das Bibliothekssystem muss den Bibliotheksausweis eines Bibliothekskunden zerstören, falls für diesen mehr als fünf offene Mahnungen existieren.“ und „Das Bibliothekssystem muss dem Bibliothekar eine

Warnmeldung anzeigen, falls für den Bibliothekskunden eine „beschädigte Rücklieferung“ registriert ist.“

Bedingte Anforderungen können extrem komplex werden, so dass sie mittels der natürlichen Sprache nur sehr unübersichtlich beschrieben werden können. Hier sollten Sie auf geeignetere Darstellungsmittel, wie beispielsweise eine Entscheidungstabelle (siehe Kapitel 10 „Prüftechniken für Anforderungen“), zurückgreifen.

Diese können dabei helfen, komplexe Bedingungsstrukturen übersichtlich darzustellen. Im Rahmen der Vollständigkeitsprüfung sind sie in der Regel das Mittel der Wahl, um nicht beschriebene Varianten von Aktionen oder Bedingungen ausfindig zu machen.

Oder verwenden Sie Aufzählungszeichen für die einzelnen Fälle (wie im obigen Beispiel).

### Schritt 3: Prüfe implizite Annahmen

Viele Sachverhalte des zu beschreibenden Systems sind gerade dem fachlich erfahrenen Stakeholder so selbstverständlich, dass er sie bei der Ermittlung von Anforderungen erst gar nicht mehr kommuniziert. Auch in anderen Anforderungsquellen, wie z. B. Dokumenten, sind diese Sachverhalte häufig nicht zu finden. Sie werden als bekannt vorausgesetzt oder man scheut sich davor, solche vermeintlich banalen Sachverhalte niederzuschreiben. Wir bezeichnen das als Muskelgedächtnis.

D. h. dem Stakeholder ist nicht mehr aktiv bewusst, über welches umfassende Wissen er verfügt.

Spätestens bei der Umsetzung der Anforderungen in ein Produkt müssen diese implizit enthaltenen Annahmen aber explizit formuliert werden (Qualitätskriterium der Vollständigkeit, vgl. Kapitel 1 „In medias RE“), da ansonsten die angemessene Funktionsfähigkeit des Systems gefährdet ist.

Implizite Annahmen (Präsupposition) sind getilgte Aussagen, die wahr sein müssen, damit andere Aussagen (Anforderungen) überhaupt einen Sinn ergeben.



Im Prinzip decken Sie durch alle beschriebenen Regeln gewisse implizite Annahmen auf, beispielsweise durch die Analyse von Nominalisierungen oder Mengen- und Häufigkeitsangaben. Werden allerdings ganze Sachverhalte oder Teile eines Systems verschwiegen, sind implizite Annahmen nur sehr schwer zu entdecken. In unserer Projektpraxis stoßen wir daher häufig auch an gewisse Grenzen.

Wenn aber auch nur die kleinste Information in den Anforderungen zu finden ist, die auf eine implizite Annahme hinweist, so können Sie diese durch das SOPHIST-REgelwerk aufdecken. Um Sie dabei zu unterstützen, haben wir unsere Erfahrung zum Auffinden impliziter Annahmen in eine weitere, relativ einfache Regel gegossen:

Hier hilft nur noch:

- Sichtenbildung bei der Use-Case-Methodik
- Geeignete Ermittlungstechniken
- Logischer Menschenverstand

T

### Regel 18: Analysieren Sie implizite Annahmen.

Achten Sie auf Signalwörter wie „Falls“, „Nachdem“, „Sobald“, „Wenn“ ...

Achten Sie auf Signalwörter, wie „eingebene Benutzerdaten“, „angezeigte View“, „berechneter Kapitalwert“ ...

Anforderungen beinhalten häufig eine (meist nur nebenbei erwähnte) Aussage, die wahr sein muss, so dass die eigentliche Anforderung überhaupt erst erfüllt werden kann. Solche Aussagen bilden dann Ihre Signalwörter, die auf implizite Annahmen hinweisen, wie z.B.

- Zeitliche/logische Ablaufbeschreibungen
- Substantive, die durch ein Bezugswort genauer beschrieben werden.

Prüfen Sie implizite Annahmen durch Anwendung der folgenden Schritte:

- Identifizieren Sie in Ihrer Anforderung das Signalwort, das auf eine implizite Annahme hinweist.
- Prüfen Sie, welche Funktionalität sich hinter dem Signalwort verbirgt.
- Prüfen Sie, ob diese Funktionalität durch andere, bereits vorhandene Anforderungen beschrieben ist.

Formulieren Sie für jede noch nicht beschriebene Funktionalität eine oder mehrere zusätzliche Anforderungen.



Steckt in der Anforderung eine implizite Annahme? Ist diese Aussage bereits irgendwo in der Spezifikation beschrieben?

Als Beispiel zur Anwendung dieser Regel schauen wir uns die folgende Anforderung des Bibliothekssystems an: „Nachdem das Bibliothekssystem die eingegebenen Registrierungsdaten eines neuen Bibliothekskunden gespeichert hat, soll das Bibliothekssystem einen Benutzer ausdrucken.“

Gibt es eine Anforderung, die beschreibt, dass Registrierungsdaten gespeichert werden müssen?

Damit die eigentliche Funktionalität der Anforderung „Benutzer ausdrucken“ einen Sinn ergibt, muss angenommen werden, dass vorher „Registrierungsdaten eines neuen Bibliothekskunden gespeichert“ wurden. Genau diese Funktionalität muss über eine andere Anforderung beschrieben sein. Ist dies nicht der Fall, haben Sie eine implizite Annahme gefunden, die im Anforderungsdokument durch eine neue Funktionsbeschreibung spezifiziert werden muss: „Das Bibliothekssystem muss Registrierungsdaten eines neuen Bibliothekskunden speichern“. Diese zusätzliche Anforderung muss anschließend natürlich noch hinsichtlich semantischer Effekte, idealerweise beginnend bei Regel 1, detailliert analysiert werden.

Gibt es eine Anforderung, die beschreibt, dass Registrierungsdaten eingegeben werden können?

Auch Substantive, die durch ein Bezugswort detailliert werden, sind Indikatoren für implizite Annahmen. Betrachten wir die obige Beispielanforderung, so stellen wir schnell fest, dass das Bezugswort „eingegebene“ vor dem Substantiv „Registrierungsdaten“ darauf hinweist, dass es eine Funktionalität zum Eingeben von Registrierungsdaten geben muss. Falls diese Funktionalität noch nicht über mindestens eine Anforderung beschrieben ist, wurde eine Information getilgt: „Das Bibliothekssystem muss dem Bibliothekar die Möglichkeit bieten, Registrierungsdaten für einen neuen Bibliothekskunden einzugeben.“

## Der drei Schritte zum Prüfen des Gesamtbildes auf einen Blick

Eine qualitativ hochwertige Anforderung hat Ihnen nun dabei geholfen, selbst komplett getilgte Informationen, wie Ausnahmeverhalten, vergessene zeitliche oder logische Bedingungen oder gar implizite Annahmen zu erkennen und gezielt aufzuspüren.

RE-Bauernregel:  
Ein einzelner Satz  
kann zeigen, dass  
Fragen im Gesamtbild  
unbeantwortet  
bleiben.

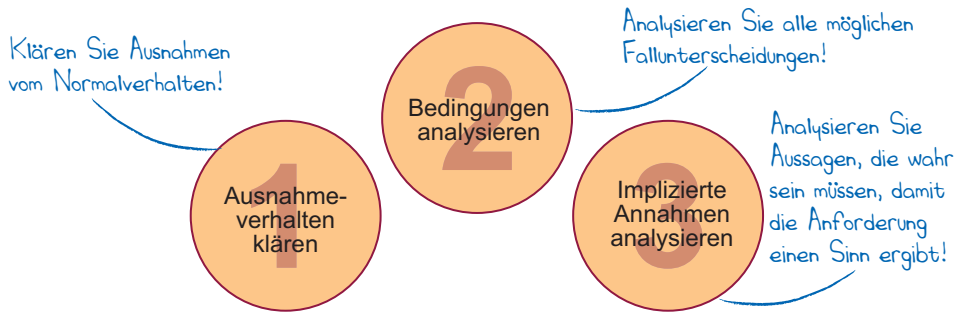
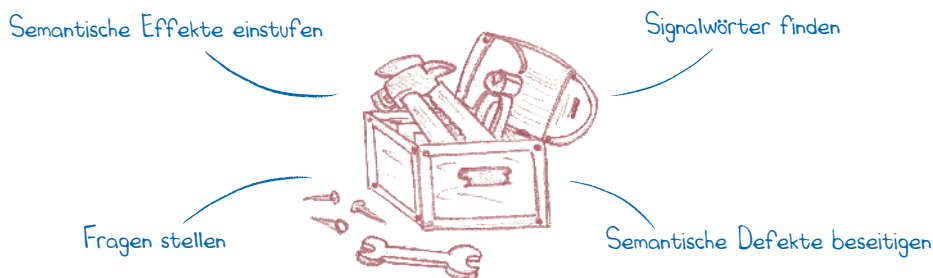


Abbildung 6.9: Die drei Schritte zum Prüfen des Gesamtbildes

Wenn Sie diese systematische Vorgehensweise Anforderung für Anforderung weiter fortsetzen, werden Sie Schritt für Schritt zu einer vollständigen und qualitativ hochwertigen Gesamtspezifikation gelangen.

## 6.8 Anwendung des SOPHIST-REgelwerks

Das SOPHIST-REgelwerk stellt Ihnen einen Werkzeugkasten zur Verfügung, um Anforderungen systematisch zu analysieren. Wir haben es bereits in vielen Industrieprojekten erfolgreich eingesetzt, die unterschiedlichste fachliche Domänen thematisierten.



Häufig werden zusätzlich Schablonen für die Formulierung von Anforderungen angewendet, die eine klare Vorgabe dafür geben, wie ein Anforderungssatz genau zu formulieren ist (siehe Kapitel 7 „Schablonen“). Dadurch kann man einige Schritte des SOPHIST-REgelwerks (z. B. Regel 1, Aktivformulierung) einsparen.

## 6 Das SOPHIST-REgelwerk

Sie können sich zu unterschiedlichen Zeitpunkten auf die Suche nach Effekten machen (vgl. Kapitel 7 „Schablonen“, Kapitel 10 „Prüftechniken für Anforderungen“ und Kapitel 11 „Qualitätsmetriken“):

Das erfordert natürlich etwas Übung - ist aber sehr effektiv.

- **Konstruktiv direkt im Interview:** Dabei prüft der Requirements-Engineer jede Aussage, die er im Interview von einem Stakeholder erhält, sofort auf mögliche sprachliche Effekte. Er hinterfragt direkt im Gespräch die fehlenden Sachverhalte, die ihm als wichtig erscheinen. So wird das nötige Wissen im Interview sehr effizient erhoben, da Wissenslücken umgehend aufgedeckt werden. Ist der Requirements-Engineer geübt, nimmt der Befragte gar nicht wahr, dass seine Aussagen systematisch nach psychotherapeutischen Gesichtspunkten untersucht und gezielt hinterfragt werden. Der Befragte merkt nur, dass das Gespräch sehr schnell auf den Punkt kommt. Auch während der Dokumentation, also dem Schreiben von Anforderungen, hilft ihm die Regeln konstruktiv weiter. So formuliert er beispielsweise die Funktionalität direkt durch ein Vollverb oder jede Anforderung direkt im Aktiv. Der konstruktive Einsatz der Regeln zum Auffinden semantischer Effekte erfordert natürlich schon ausreichend Praxiserfahrung mit den Frage-techniken des SOPHIST-REgelwerks.
- **Analytisch als Prüfkriterien auf der Basis erstellter Anforderungen:** Liegen bereits Anforderungen in schriftlicher Form vor, so werden diese mittels der sprachlichen Regeln überprüft. Auslassungen, Unklarheiten usw. werden dann hinterfragt. Hier dient das REgelwerk als eine Technik zur analytischen Qualitätssicherung.

Da Papier ja geduldig ist, lässt sich auf der Basis schriftlicher Anforderungen die Technik gut üben.

RE-Bauernregel:  
Ist die Zeit dir weggerannt, nimm nur ein paar Regeln in die Hand.

Natürlich sind auch Requirements-Engineers nicht vor den unerwünschten Defekten der Fokussierung und Vereinfachung gefeit, so dass man sich dementsprechend vorbereiten muss.

Für eine erste Anwendung der Methode empfiehlt es sich zunächst, bereits dokumentierte Anforderungen zu analysieren. Nach einiger Zeit Übung sind die psychotherapeutischen Ansätze mit dem SOPHIST-REgelwerk so in das Denken integriert, dass Aussagen eines Stakeholders direkt im Gespräch untersucht und hinterfragt werden können.

Wenden Sie zu Beginn auch nur einige wesentliche Regeln an, durch die Sie Ihre Aufmerksamkeit auf die Gruppe der gefährlichen Wiederholungstäter konzentrieren können.

Regel 1

Regel 2,3 oder 4

Regel 6

Regel 10 und 12

Regel 17

Nach unserer Erfahrung sollten Sie die folgenden Regeln des SOPHIST-REgelwerks mit sehr hoher Priorität beachten:

- Formulieren Sie jede Anforderung im **Aktiv**.
- Lösen Sie **Prozesse** auf und drücken Sie diese durch **Vollverben** aus.
- Analysieren Sie **fehlende Informationen zum Prozesswort**.
- Klären Sie **Mengen- und Häufigkeitsangaben**.
- Klären Sie **fehlende Bedingungen**.



Am besten können Sie genannten Regeln des SOPHIST-REgelwerks verinnerlichen, wenn Sie diese mit den zugehörigen Fragen auf ein Blatt Papier notieren. Hängen Sie dieses in Augenhöhe an Ihren Arbeitsplatz und werfen Sie jeden Tag bewusst einen Blick darauf.

Versuchen Sie auch nicht krampfhaft, auf jede Anforderung jede Regel anzuwenden. Wenn Ihre Anforderung beispielsweise nicht das Eintreten einer fachlich oder technisch motivierten Bedingung erfordert, dann müssen Sie selbstverständlich auch keine künstliche Bedingung konstruieren oder Fallunterscheidungen analysieren.

Wenn Sie sich sicher genug fühlen in der Anwendung der hoch priorisierten Regeln, dann greifen Sie noch etwas tiefer in die Trickkiste.

Als geübter Anwender der wesentlichen Regeln des SOPHIST-REgelwerks beachten Sie zusätzlich unbedingt die folgenden Regeln:

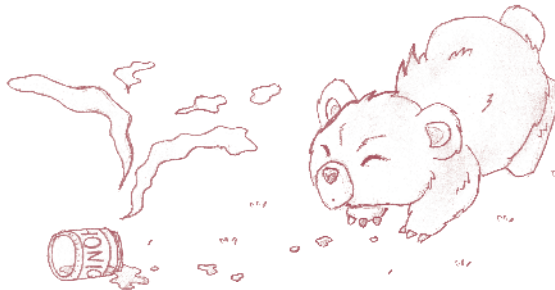
- Klären Sie **Ausnahmen vom Normalverhalten**.
- Analysieren Sie **implizite Annahmen**.
- Formulieren Sie **Vergleiche und Steigerungen** mess- bzw. testbar.

Regel 16

Regel 18

Regel 8

Und wenn Sie dann ein richtiger Experte sein wollen (oder sein müssen), dann wenden Sie die noch verbleibenden Regeln auf Ihre Anforderungen an.



Aber auch als richtiger Experte sollten Sie nie zum Perfektionisten mutieren. Streben Sie nicht völlig unreflektiert nach absolut defektfreien Anforderungen, sondern berücksichtigen Sie immer Ihre Projekttrandbedingungen. Bei dem Thema Qualität liegen Theorie und Praxis häufig weit voneinander. Qualität bei der Analyse und Dokumentation von Anforderungen ist mit viel Aufwand und hohen Kosten verbunden. Perfektionismus wird somit sehr schnell zu einem Luxus, den Sie sich in den meisten Projekten nicht leisten können.

Die Erkenntnis, dass perfekte Kommunikation unmöglich ist, wird Ihnen den Projektalltag ganz erheblich erleichtern. Der möglichst kompetente Umgang mit dieser Tatsache ist einer der Faktoren, der den Erfolg im Bereich Requirements-Engineering ausmacht.

Hier gilt der  
Grundsatz:  
Angemessenheit  
statt Perfektion!

### 6.9 Ene, mene, muh – und defektbereinigt bist du!

#### Checkliste SOPHIST-REgelwerk

- Ich **verstehe** die **Transformationsprozesse**, die zwischen innerer Vorstellung und sprachlichem Ausdruck zum Tragen kommen.
- Ich kann die am häufigsten auftretenden **sprachlichen Effekte** in Anforderungen **identifizieren**.
- Ich weiß, auf welchem **Spezifikationslevel** ich welches Vorgehen **beachten** sollte.
- Ich kenne die wichtigsten **Signalwörter** und mir ist klar, bei welchem Signalwort ich welche **Fragen stellen** muss.
- Ich **verstehe** die unterschiedlichen **Betrachtungswinkel** (Satzbestandteile, Satz, Gesamtbild), aus denen eine Anforderung zu analysieren ist.
- Ich kann das **SOPHIST- REgelwerk als analytische Prüftechnik** auf dokumentierte Anforderungen **anwenden**.
- Ich habe ausreichend Übung in der Anwendung des SOPHIST-REgelwerks und kann die wichtigsten **Regeln** nun auch **konstruktiv** während eines Stakeholder-Interviews **anwenden**.