

Chris Rupp

**Von Ideen und Wünschen zu dokumentierten
Anforderungen**

25.-27. März 2002 in Tutzing

SOPHIST GmbH

Von Ideen und Wünschen zu dokumentierten Anforderungen

Chris Rupp

SOPHIST Gesellschaft
für innovatives Software-Engineering mbH
Vordere Cramergasse 11-13
D-90478 Nürnberg
info@sophist.de

Abstract: Die richtigen Kundenwünsche zu kennen ist heute der entscheidende Faktor in der Systementwicklung. Typischerweise werden Stakeholder nach Ihren Wünschen in einem Interview befragt. Doch nicht immer sind die Interviewergebnisse das, was sich Designer und Entwickler wünschen. Oftmals kommt dabei nicht mal das zu Tage, was der Kunde selbst wünscht. Haben Sie sich schon mal alternative Wege überlegt um an bewusste aber auch unbewusste und unterbewusste Wünsche Ihres Kunden heranzukommen? Gerade das Erfinden von Requirements statt dem mühevollen Aufsammeln vorhandener Requirements gewinnt in einer Zeit rascher technischer Innovationen mehr und mehr an Bedeutung. Die wirklichen Hotseller sind die Produkte, die die geheimen Wünsche der Kunden befriedigen. Moderne Requirements Engineering-Methoden bieten Ihnen zahlreiche Alternativen zum klassischen Interview. Im Folgenden reißen wir lediglich einige der aktuellen Methoden an.

1 Einleitung

Warum Anforderungen analysieren? Eine provokante aber nicht seltene Frage, die man erfahrungsgemäß immer wieder beantworten muss. Verschiedene Untersuchungen zeigen, dass die Mehrzahl aller Fehler in Softwareprodukten in der Analysephase entstehen. Fehler, die in einer frühen Phase der Softwareentwicklung gemacht und erst in späteren Phasen behoben werden (wenn es dafür eigentlich schon zu spät ist), sorgen für einen Summationseffekt, bei dem diese sich in spätere Phasen fortpflanzen und potenzieren. Es ist schwierig vorzustellen wie jemand ein Haus zu bauen beginnt ohne Pläne zu haben. Überdies ist es leicht einzusehen, dass Änderungen während der Bauphase oder gar erst nach der Fertigstellung wesentlich teurer sind als die selben Änderungen in der Planungsphase.

Dass dies nicht nur auf den Hausbau sondern auch auf die Entwicklung von Software zutrifft wurde von verschiedensten Institutionen und Firmen nachhaltig erwiesen. Tatsächlich lassen sich ca. 65% der schwerwiegenden Fehler in Programmen auf Unzulänglichkeiten in der Analysephase zurückführen. Darum sollten auch Softwareprojekte gründlich analysiert und geplant werden, damit nicht teure Änderungen durch mangelnde Funktionalität oder schlechte Performance zu unzureichenden Arbeitsbedingungen führen. Resümee: Um eine Anforderungsanalyse kommt man nicht herum und je besser,

spricht effizienter durchgeführt wird, desto größer sind die gewonnenen Vorteile. "Worüber ich mich immer wieder wundere ist, dass viele Forscher nichts Neues zustande bringen. Ich glaube, dass dies zum Teil davon herrührt, dass sie sich zu stark an den Weg halten, welchen ihre Vorgänger schon beschritten haben:

„Man muss sich vom großen Weg entfernen, um Neues zu finden.“

(Gottfried Wilhelm Leibniz).

Wir wollten uns hier mal einen Schritt vom normalen Weg der Requirementsermittlung - der aus vielen zähen Interviews besteht - wegbewegen. Mal sehen was es neben diesem klassischen, durchaus häufig auch erfolgreichen Vorgehen noch an Möglichkeiten existiert.

2 Die Klassiker: Befragungstechniken

Eine erste einfache und daher auch sehr verbreitete Analysemethode stellen die Befragungstechniken dar. Es wird dabei versucht, direkt vom Stakeholder eine möglichst genaue und unverfälschte Aussage über die Anforderungen des zu untersuchenden Systems zu erhalten.

Typische Vertreter von Befragungstechniken sind das freie Interview, das fragebogengestützte Interview und Interviews in Gruppensitzungen oder auch die Selbstaufschreibung durch die Stakeholder.

Problematisch sind hierbei zwei Punkte. Befragungstechniken sind tendenziell analytikergetrieben. D. h. die Aktivität, welche die Richtung bestimmt, in der die Anforderungsanalyse fortschreitet, geht überwiegend vom Analytiker aus. Dadurch kann es dazu kommen, dass Anliegen der Stakeholder verdrängt, vergessen oder vernachlässigt werden.

Zudem setzen die Befragungstechniken voraus, dass der Befragte sein Wissen explizieren kann, d.h. nicht nur im Kopf hat sondern auch kommunizieren und auf den Punkt bringen kann und dazu auch noch Zeit und Lust hat. Leider trifft das nicht immer zu.

3 Viel zu selten praktiziert: Beobachtungstechniken

Neben den Befragungstechniken sind v. a. auch die Beobachtungstechniken von industrieller Bedeutung.

Der Analytiker macht sich selbst ein Bild der Tätigkeiten und Arbeitsabläufe durch die Beobachtung der Stakeholder an ihren Arbeitsplätzen und kann ungünstige Arbeitsabläufe und Fehler der Stakeholder selber erkennen. Bei den Beobachtungstechniken besteht die Möglichkeit ungünstige Arbeitsabläufe zu erkennen und diese zu verbessern, sowie Fehler der Stakeholder aufzudecken. Beides kann dem Stakeholder unbewusst sein, da er viele Handlungen und Tätigkeiten seiner Arbeit gewohnheitsmäßig automatisch ausführt ohne weiter darüber nachzudenken. Auch liefern die Beobachtungstechniken einen unverfälschten Eindruck der Sprache und Terminologie des entsprechenden Fachgebietes, was weitere Befragungen erleichtern kann und Missverständnisse reduziert. Zuletzt besteht ein Vorteil darin, dass der Stakeholder seine normale Tätigkeit nicht unterbre-

chen muss, somit sein Zeitaufwand minimiert wird. Der Zeitaufwand für die Dokumentation des gewonnen Wissens liegt beim Analytiker.

Zukünftige Anforderungen können nur in einem geringeren Maße durch die Feststellung ungünstiger oder unzureichender Sachverhalte erhoben werden und müssen vom Analytiker abgeleitet und dann noch einmal zusammen mit dem Stakeholder reflektiert werden. Bekannte Vertreter sind Videoanalyse, Feldbeobachtung oder ‚In die Lehre‘ (Apprenticing) gehen.

4 Eine Quelle für Innovationen: Brainstorming, Mind Mapping

Neben den Befragungs- und Beobachtungstechniken gibt es eine Reihe von Gruppentechniken.

Zwei der wohl bekanntesten Gruppenkreativitätstechniken sind Brainstorming und Mind Mapping. Es lohnt sich diese Techniken bei der Erhebung grober Anforderungen und Ziele des Systems zu nutzen. Da es reichlich Literatur zu beiden Techniken gibt, empfehlen wir ein Studium eines der reichhaltig vorhandenen Werke.

5 Für Mutige: Der Wechsel der Perspektive

Ein weiterer Ansatz basiert auf der Annahme, dass jedes Problem aus unterschiedlichen Richtungen betrachtet werden sollte. Ein bekannter Vertreter dieser Mehr-Sichten-Modelle ist Edward de Bonos 6-Hut-Denken. Hierbei steht jeder der sechs Hüte für eine andere Sichtweise, die von jedem Teilnehmer einmal eingenommen wird. Die auf dieser Grundlage basierenden Lösungsansätze betrachten das Problem aus allen erdenklichen Sichten.

Diese Methode wird angewendet um die Systemziele und Anforderungen auf einem sehr groben Abstraktionsniveau zu formulieren. Hier werden sehr viele unterschiedliche Sichten generiert und die entsprechenden Anforderungen abgeleitet.

Diese Methode ermöglicht es sogar manchen in seiner Sicht sehr festgefahrenen Stakeholder seine Denkweise zu verlassen und in eine andere Rolle zu steigen. Diese Technik produziert Ideen aus allen erdenklichen Sichten heraus und erzeugt damit ein sehr rundes Bild der zu analysierenden Realität. Wichtig ist ein guter Moderator, der alles mitdokumentiert, und die Akzeptanz der Stakeholder.

Als Nachteil erweist sich diese Technik für viele eher introvertierte oder konservative Stakeholder als ziemlich abgehoben Technik. Sie muss sehr vorsichtig eingeführt werden um die Beteiligten nicht zu verschrecken. Der Analytiker läuft Gefahr mit der Psychoecke assoziiert zu werden..

6 Des Pudels Kern: Essenzbildung

Die Essenzbildung reduziert alle derzeit gültigen und geforderten Arbeitsabläufe auf ihre fachliche Essenz. Ziel ist eine deutliche Minimierung der Komplexität eines Systems.

Bei der Erhebung von Anforderungen (z.B. mittels „in die Lehre gehen“, Selbstaufschreibung, ...) besteht die Gefahr, dass einzelne derzeit gültige Arbeitsabläufe einschließlich der derzeit getroffenen technologischen Ablaufentscheidungen falsch dokumentiert werden. Es werden unnötige Vorgänge beschrieben, die weder direkte noch indirekte Auswirkungen auf die Anforderungen haben. Bevor darauf basierend eine Neuentwicklung eines Systems starten kann, ist es daher wichtig, diese Abläufe auf Ihre fachliche Essenz zurückzuführen. Ziel ist es durch Abstraktion technologie- und lösungsneutrale Anforderungen zu finden. Diese müssen dann evtl. sofort wieder in eine pragmatische Form umgesetzt werden. Dennoch ist es unabdingbar vorher die Essenz des fachlichen Sachverhalts herauszuarbeiten. Dadurch kann die Komplexität eines Systems deutlich minimiert werden. Häufig werden mehrere Lösungen (die von unterschiedlichen Stakeholdern gefordert werden) innerhalb eines Systems implementiert. Wären diese Lösungsansätze auf ihre jeweilige Essenz zurückgeführt worden, hätte man das Potenzial besessen evtl. eine einzige, optimierte Lösung, statt vieler weniger optimaler Lösungen zu implementieren. Durch Abstraktion findet man allgemeingültige universelle Regeln, die man bei verschiedenen Projekten nutzen kann. Aufgrund dieser Tatsache können alte Lösungen leicht abgeändert bei neuen Projekten angewandt werden. Durch die Generalisierung kommen ungeahnte Zusammenhänge zum Vorschein. Durch die Trennung von Lösung und fachlicher Essenz wird der Blick des Analytikers geschärft, das Problem tritt in den Vordergrund, nicht die (falsche?) Lösung. Die Ermittlung der Essenz fällt jedoch vielen Stakeholdern sehr schwer, da sie sehr stark im Konkreten verhaftet sind. Meist ist dazu ein externer Analytiker nötig, der sehr gut abstrahieren kann. Ideal sind dabei Personen, die nicht mit der Materie behaftet sind, da sie den nötigen Abstand besitzen.

7 Nicht nur für UML-Fans: Analyse der Business-Events (Use Case Modellierung)

Um nicht sofort in der Komplexität von Detailaufgaben zu versinken, ist es sinnvoll zuerst eine Außenaufsicht auf das zu erstellende System zu dokumentieren. Bei der Analyse der Business-Events (Anwendungsfallmodellierung) stehen zwischen dem auslösenden Ereignis und dem wirtschaftlich wertvollen Ergebnis dann jeweils ein Use Case, dessen Bearbeitung durch das System unterstützt werden muss. Details hierzu sind in vielen UML-Werken, z.B. [Oesterreich01] zu finden.

8 Nicht kommuniziertes zur Sprache gebracht: Neurolinguistische Programmierung

Die Neurolinguistische Programmierung (NLP) bietet neben den genannten Analysemethoden eine Reihe neuer Elemente, Modelle und Techniken. Mit diesen kann eine effektive, zielgerichtete Kommunikation gefördert und Interviews erfolgreich durchgeführt werden. Die Techniken dienen insbesondere dem Lesen der Körpersprache und dem richtigen Stellen von relevanten Fragen im zwischenmenschlichen Informationsaustausch, um ungeahnte oder ungenaue Anforderungen des Betroffenen zu identifizieren.

Um Gesagtes besser verstehen zu können, bedient sich die NLP eines sog. „Metamodells“ der Sprache. Durch Anwendung dieses Modells können somit evtl. ungeahnte Anforderungen des Betroffenen gefunden werden. Es können aber auch mehrdeutige und ungenaue Anforderungen identifiziert werden.

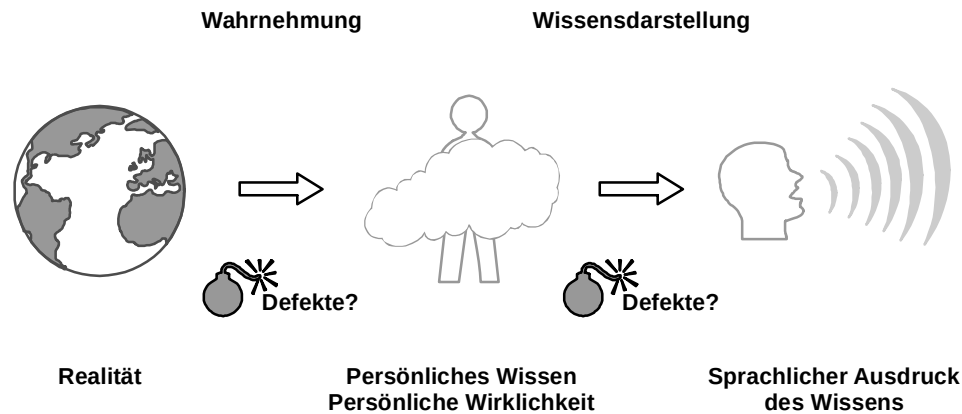


Abb. 1: Prinzip des Metamodells der Sprache aus dem NLP

Das Metamodell der Sprache ist ein Modell zur Gesprächsführung, das verwendet werden kann, um Probleme in der Kommunikation gezielt zu hinterfragen. Es beruht auf der Annahme, dass beim Explizieren von Wissen ex- und implizite Transformationen durchgeführt werden. Bei der Darstellung und Formulierung von Realität wirken unterschiedliche Einflüsse auf den Menschen ein. Das jeder Person eigene Bewusstsein, seine individuelle Umgebung und sein fachlicher Hintergrund verfälschen oder beeinflussen seine Wahrnehmung. Dieser Transformationsprozess ist unterbewusst und nicht zu ändern. Eine weitere Verfälschung der Realitäten (in der Analyse übertragen: eine Verfälschung des gewünschten Systems) ist beim Vermitteln von Informationen zu beobachten. Fragen Sie zehn Nutzer eines Systems zu einem bestimmten Sachverhalt – sie werden zehn unterschiedliche Schilderungen zu hören bekommen. Dies ist zum Glück behebbar, indem man das Wissen mehrerer Köpfe sammelt und geeignet verarbeitet oder durch NLP Fragetechniken versucht möglichst viel Informationen von wenigen Personen zu erhalten. D. h. es wird versucht die Darstellungstransformationen zu untersuchen und rückgängig zu machen. Die drei wichtigsten Transformationen sind Tilgung, Generalisierung und Verzerrung. Ein mit NLP-Techniken geschulter Analytiker kann diese durch die Auswahl bestimmter Fragen erkennen und beheben.

NLP-Techniken können in Kombination mit anderen Erhebungstechniken eingesetzt werden. So ist es möglich sie im Rahmen eines Interviews bezüglich der Anforderungen als Gesprächsleitfaden zu verwenden oder sie bei Reviews von Anforderungen einzusetzen. Sie bilden überdies ein gutes Hilfsmittel bei der Erstellung von Anforderungen, indem bereits beim Schreiben darauf geachtet wird, dass verlustbehaftete Transformationen vermieden werden. Behält man dies im Auge, lässt sich sukzessive ein mächtiges Regelwerk erstellen.

Während andere Techniken wie Interviews oder Fragebögen eigentlich nur an der „Oberfläche“ des vorhandenen Wissens eines Stakeholders kratzen, geht die NLP weiter. Hier wird nach verborgenem Wissen gegraben. Gerade das unterbewusste, versteckte Wissen würde ein Stakeholder vermutlich erst sehr spät und per Zufall in den Entwicklungsprozess einbringen, mit NLP – Techniken hingegen wird er angehalten dies sofort zutun. Erfahrungen zeigen sogar, dass durch einen kleinen Anstoß der berühmte Stein ins Rollen gebracht wird. Gerade bei der Befragung von introvertierten oder schüchternen Menschen sind diese Techniken sehr effektiv.

Als Nachteil der NLP-Techniken erweist sich das Erlernen und die korrekte Anwendungen, die sehr stark auf Erfahrungen beruht. Die verwendeten Methoden lassen sich nicht „von heute auf morgen“ lernen, geschweige denn korrekt anwenden. Bei der Verwendung dieser Technik müssen daher externe Berater und Analytiker herangezogen werden, die eine solche Erfahrung aufweisen können. Zudem müssen die verwendeten Techniken so transformiert werden, damit sie von Stakeholdern akzeptiert werden. Insbesondere darf nicht der Eindruck entstehen, dass zwischen Analytiker und Befragtem ein Arzt – Patient Verhältnis existiert. Die Anpassung der Techniken an informations-technische Gegebenheiten ist aber in aller Regel problemlos.

9 Für Spielernaturen: Simulationsmodelle - Szenarien und Prototypen

Eine gängige Technik, die auch in der Industrie häufig verwendet wird sind Simulationsmodelle.

Manche Anforderungen an ein System werden dem Benutzer erst in dem Moment klar, in dem er es zum erstenmal bedient. Um diese Vorstellung bereits im Vorfeld so konkret wie möglich zu gestalten, werden daher Prototypen erstellt.

10 Der harte Weg: Systemarchäologie

Als Systemarchäologie bezeichnet man die Technik, die Informationen bezüglich eines neuen Systems aus der Dokumentation oder der Implementierung eines Altsystems herauszuholen. Diese Technik wird häufig dann angewendet, wenn das Wissen über die im System implementierte Fachlogik verloren gegangen ist oder nur noch in den Köpfen sehr weniger Mitarbeiter steckt. Bei diesen Verfahren wird durch einer Analyse des bestehenden Codes sichergestellt, dass keine der bereits implementierten Funktionalitäten vergessen wird und zumindest die Fachlogik des Altsystems neu erhoben. Somit stellt dieses Verfahren sicherlich von allen vorgestellten Techniken, das einzige dar, dass die vollständige Umsetzung in ein Neusystem implizit sicherstellt.

Dennoch ist Systemarchäologie ein sehr aufwendiges Verfahren! Zudem wird lediglich der Funktionsumfang des Altsystems ermittelt. Dieser kann jedoch deutlich von den Anforderungen an ein Neusystem abweichen.

11 Für „Extremisten“: User Stories nach XP

Als letzte Analysemethode möchten wir auf eine relativ neue Technik eingehen, den User Stories aus Extreme Programming (kurz XP).

Bei dieser Technik wird sehr großer Wert auf die enge Einbindung des Kunden in den *gesamten* Entwicklungsprozess gelegt. Der Kunde selbst schreibt durch sogenannte User Stories seine Wünsche an sein System auf, die dann im Sinne des „on-site-customer“-Prinzips gemeinsam mit den Entwicklern implementiert werden.

Der Hauptvorteil von XP ist natürlich der hohe Grad an Kommunikation zwischen Entwickler und Kunde. Aufgrund der engen Einbindung des Kunden in den Entwicklungsprozess, sind Fehlentwicklungen und Missverständnisse äußerst selten. Überdies tragen besonders die vielen Acceptance Tests und die kurzen Iterationen positiv zu einer fehlerfreien Entwicklung bei,

Des weiteren wird durch die ständige Kommunikation innerhalb des Teams eine positive Arbeitsatmosphäre mit motivierten Mitarbeitern geschaffen.

Wie man sich vorstellen kann, mangelt es nicht an Kritik an XP. Ein großer Nachteil ist sicherlich das Fehlen einer geeigneten Dokumentation, da ein Grossteil des Wissens lediglich in der persönlichen Kommunikation ausgetauscht wird. Dies kann besonders bei Erweiterungen und Wartung des Systems zu Schwierigkeiten führen. Es existiert zwar Dokumentation in Form der Acceptance Tests, jedoch ist es fraglich, ob diese auch für die Entwickler ausreichend sind, die nicht Mitglied des ursprünglichen XP-Teams waren. Somit wird eine starke Abhängigkeit von den aktuellen Wissensträgern erzeugt.

Weiterhin erfordert die durchgehende, enge Einbindung des Kunden einen hohen Ressourcenaufwand, da der Kunde selbst sehr viel Zeit mit dem Entwicklungsteam verbringen muss, um ständig für Fragen zur Verfügung zu stehen. Bei einer begrenzten Anzahl von Know-how-Trägern ist dies unmöglich. Ein weiteres Problem stellt eine verteilte Entwicklung dar. XP geht immer von einer Ausgangssituation aus, bei der alle Teammitglieder an einem Ort zusammengefasst werden. Auch das ist eine Ausgangssituation, die in der Industrie kaum anzutreffen ist.

12 Und was lernen wir daraus?

Wir hoffen, wir konnten Ihnen aufzeigen, auf welche unterschiedliche Arten man an bewusste, unbewusste und unterbewusste Requirements herankommen kann. Überlegen Sie für Ihr Unternehmen welche Arten von Anforderungen Sie eigentlich erheben wollen und welche Techniken aus finanziellen, kulturellen, technischen Möglichkeiten machbar und geeignet sind. Requirements Engineering bietet Ihnen eine breite Auswahl unterschiedlichster Methoden – nutzen Sie sie!

Literatur:

- [Bandler94] **Bandler, R.:** Metasprache und Psychotherapie: Die Struktur der Magie I. Paderborn, Junfermann 1994. ISBN 3-87387-186-6
- [Beck01] **Kent B. und Fowler, M.:** Planning Extreme Programming, Addison Wesley 2001. ISBN 0-201-71091-9
- [Mehrmann 95] **Mehrmann, E.:** Vom Konzept zum Interview. Informationsgespräche richtig planen und führen. München, Econ TB Verlag 1995. ISBN 3-61221-218-4
- [Mucchielli 73] **Mucchielli, R.:** Das Gruppeninterview. Praktische Übungen / Theoretische Einführung. Salzburg, Otto Müller 1973. ISBN 3-70130-490-4
- [Oesterreich01] **Oesterreich, B.:** Objektorientierte Softwareentwicklung: Analyse und Design mit der Unified Modeling Language. München, Oldenbourg, 2001. ISBN 3-486-25573-8
- [Robertson 01] **Robertson, S.:** Techniken für die Anforderungsanalyse. In: OBJEKTSpektrum Januar/Februar 2001, S. 58-65
- [Rupp00] **Rupp, C.:** Requirements-Engineering – der Einsatz einer natürlichsprachlichen Methode bei der Ermittlung und Qualitätsprüfung von Anforderungen. In: OBJEKTSpektrum März/April 2000, S. 22-26
- [Rupp01] **Rupp, C.; SOPHIST GROUP.:** Requirements-Engineering und -Management – Professionelle, iterative Anforderungsanalyse für die Praxis. München, Carl Hanser Verlag 2001. ISBN 3-446-21664-2
- [Wiley99] **Wiley, B.:** Essential System Requirements, A Practical Guide to Event-Driven Methods, Dec 1999



Falls ein Lebenslauf von mir benötigt wird, bitte diesen verwenden: Chris Rupp legte das Fundament des NLP-basierten Requirements Engineering, erarbeitete und veröffentlichte pattern orientierte Requirements Engineering Ansätze. Sie ist geschäftsführende Gesellschafterin der SOPHIST GmbH und SOPHIST Technologies GmbH und Autorin zahlreicher international verlegter Publikationen.

Chris ist spezialisiert auf die Analyse sicherheitskritischer technischer Großprojekte internationaler Kunden. Ihre Arbeitsmethodik umfasst u.a. natürlichsprachliche und objektorientierte Analysen sowie Bereiche der Organisationspsychologie und des Neuro-Linguistischen Programmierens (NLP). Ihr Unternehmen SOPHIST Technologies GmbH ist der Hersteller des Requirements Engineering und Management Tools C.A.R.E.