Untersuchung von natürlichsprachlichen Analyse- und Spezifikationsdokumenten

Ralf Melchisedech
Institut für Informatik, Abteilung Software Engineering, Universität Stuttgart
E-Mail: melchise@informatik.uni-stuttgart.de

Zusammenfassung: In diesem Artikel werden die Ergebnisse einer Untersuchung natürlichsprachlicher Spezifikationen vorgestellt, die in industriellen Projekten erstellt wurde. Ziel der Untersuchung war es, einen Einblick in den Stand der Praxis zu erhalten. Gegenstand der Untersuchung waren die Spezifikationsdokumente und der Spezifikationsprozeß. Die Spezifikationsdokumente wurden auf Aufbau, Inhalt und verwendete Notationen hin untersucht, beim Spezifikationsprozeß standen der Werkzeugeinsatz, die Kommunikation mit den Kunden und die Prüfung der Dokumente im Vordergrund. Soweit wie möglich wurden die Informationen quantitativ erhoben.

Schlüsselwörter: Requirements Engineering, Spezifikation, natürliche Sprache, informal, Pflichtenheft, Lastenheft, empirische Untersuchung, Fallstudie.

1 Einführung

In den letzten Jahren wurden sehr viele neue Methoden zur Erstellung von Spezifikationen publiziert. Viele dieser Methoden verwenden halb-formale oder formale Notationen (siehe z.B. Rumbaugh (1997), Gaudel (1994)).

Die industrielle Praxis sieht dagegen anders aus. In den meisten Unternehmen wird, soweit überhaupt Spezifikationen erstellt werden, als Notation hauptsächlich natürliche Sprache verwendet, oftmals ergänzt durch informale Notationen (z.B. Skizzen) oder halb-formale Notationen (z.B. Datenflußdiagramme, ER-Diagramme).

Im folgenden wird eine Untersuchung von Spezifikationen, die in drei Projekten in drei Unternehmen entstanden ist, beschrieben. Zum Erstellungszeitpunkt war den Systemanalytikern nicht bekannt, daß die Spezifikationen und der Spezifikationsprozeß untersucht werden würden. Deswegen ist die Wahrscheinlichkeit sehr hoch, *typische* Spezifikationen erhalten zu haben.

2 Untersuchungsgegenstand

In diesem Kapitel werden die Projekte und die in den Projekten realisierten Systeme kurz charakterisiert. Aus Vertraulichkeitsgründen können in diesem Rahmen weder die Unternehmen genannt noch die Projekte genauer beschrieben werden. Die Projekte werden im folgenden mit Projekt A, B und C bezeichnet.

2.1 Projektkenndaten

Im Projekt A wurde eine Cross-Plattform-Entwicklungsumgebung für eingebettete Software realisiert. Das Entwicklungssystem soll auf IBM PC's unter Windows laufen. Bei der Zielumgebung, in der die von dem System erzeugte Software ausgeführt werden soll, handelt es sich um ein Spezialbetriebssystem für Echtzeitanwendungen. Die Benutzerinteraktion erfolgt über eine menügesteuerte, textuelle Schnittstelle. Mit dem Entwicklungssystem können Programme geschrieben werden, die einem vorgegebenen, auf Wiederverwendbarkeit ausgerichteten Programmierparadigma folgen. Dieses wurde im Rahmen einer Vorstudie entwickelt und war zu Projektbeginn fest. Das Projekt wurde inzwischen abgeschlossen. Das entstandene System wird bei den Kunden eingesetzt.

Im Projekt B wurde ein graphischer Editor für Netzwerktopologien entwickelt. Ein Netzwerk besteht aus Symbolen und Verbindungen zwischen diesen Symbolen. Jedes Symbol wiederum kann durch ein Netzwerk verfeinert werden. Die Anwendung soll an eine bereits vorhandene Datenbank angebunden werden. Ein altes System, welches bisher die Aufgaben erfüllt hat, soll durch das neue System teilweise ersetzt werden. Das Projekt wurde inzwischen abgeschlossen. Das entstandene System wird in Kundenprojekten zu Evaluationszwecken eingesetzt.

Im Projekt C wird ein System entwickelt, welches im Bankenbereich eingesetzt wird. Angestellte sollen von ihrem Terminal aus Anfragen an eine zentrale Datenbank eines Drittanbieters durchführen können. Die Anfragen sollen online erfolgen. Innerhalb weniger Sekunden sollen die Antworten verfügbar sein. Das System soll in die schon vorhandene Arbeitsumgebung der Bankangestellten eingebunden werden. Die Anfragen bei dem Drittanbieter wurden bisher manuell durchgeführt. Diese manuelle Tätigkeit soll durch das neue System automatisiert werden. Das Projekt wurde noch nicht beendet, die Spezifikationsphase ist jedoch schon abgeschlossen.

Projektkenndaten	Projekt A (Entw. System)	Projekt B (graph. Front-End)	Project C (Online-System)
Dauer des Projekts (in Monaten)	26	29	(noch nicht beendet)
Aufwand des Projekts (in MM)	48	155	(noch nicht beendet)
Anzahl Projektmitarbeiter	2-7	8-13	2-3
Dauer der Spez.-Phase (in Monaten)	5	4	12
Aufwand der Spez.-Phase (in MM)	13	2,5	8
Anzahl Mitarbeiter in der Spez.-Phase	2-3	1	1
Finanzierung	Festpreisprojekt	Eigenfinanzierung	Eigenfinanzierung
Prototyp erstellt?	GUI-Prototyp	Vorversion	Prototyp
Wartung	Kunde	Entwickler	Entwickler

Tabelle 1 : Vergleich der Projektkenndaten der untersuchten Projekte

2.2 Systemkenndaten

Um besser abschätzen zu können, inwieweit die untersuchten Systeme Ähnlichkeiten aufweisen, wurden sie anhand der in Tabelle 2 beschriebenen Kriterien klassifiziert.

Durch die Datenkomplexität soll beschrieben werden, wie komplex ein einzelnes Datum aufgebaut ist. Es macht durchaus einen Unterschied, ob es sich bei den Daten um viele, einfach strukturierte Einzeldaten oder um einige wenige, dafür aber sehr komplex aufgebaute Daten handelt.

Unter Metadaten werden Daten zur Beschreibung von Daten verstanden. Ein System mit einer hohen Metadatenkomplexität verfügt über eine Datenbeschreibungssprache, mit der der Aufbau von Daten definiert werden kann. Bei einem System mit niedriger Metadatenkomplexität sind die Datentypen vorgegeben und können nur eingeschränkt angepaßt werden.

Aufgrund der vorliegenden Spezifikationen wurde geschätzt, inwieweit die Kriterien auf die einzelnen Projekte zutreffen. Hierbei handelt es sich nicht um absolute Angaben. Die Angaben dienen lediglich dazu, die Systeme untereinander vergleichen zu können.

Gemeinsam ist allen Systemen, daß sie keine harten Zeitanforderungen haben. Das Entwicklungssystem (Projekt A) und das graphische Front-End (Projekt B) weisen einige Ähnlichkeiten auf. Die wesentlichen Unterschiede liegen darin, daß es im Entwicklungssystem eine komplexe Datenbeschreibungssprache gibt und daß die Algorithmen komplexer sind. Im graphischen Front-End hingegen ist die Benutzungsschnittstelle komplexer. Das Online-System (Projekt C) unterscheidet sich in fast allen Kriterien von den anderen beiden Systemen.

3 Untersuchungsergebnisse - Spezifikationsdokumente

In diesem und im nächsten Kapitel werden die Untersuchungsergebnisse beschrieben. In Kapitel 3 stehen die Spezifikationsdokumente im Vordergrund, während in Kapitel 4 der Prozeß genauer beschrieben wird.

Systemkenndaten	Projekt A (Entw. System)	Projekt B (graph. Front-End)	Project C (Online-System)
Datenkomplexität	hoch	hoch	niedrig
Metadatenkomplexität	hoch	niedrig	niedrig
Datenmenge	niedrig	niedrig	mittel
Anzahl Funktionen	mittel	mittel	niedrig
Komplexität der Algorithmen	hoch	mittel	mittel
Komplexität der Benutzungsschnittstellen	mittel	hoch	niedrig
Komplexität der Systemschnittstellen	mittel	mittel	hoch
Zeitanforderungen	nicht hart	nicht hart	nicht hart
Anzahl gleichzeitiger Benutzer	viele	viele	sehr viele

Tabelle 2 : Vergleich der Systemkenndaten der untersuchten Projekte

3.1 Aufbau der Spezifikationsdokumente

Im Projekt A (Entwicklungssystem) wurden zwei Dokumente erstellt: ein Lastenheft (LH) und ein Pflichtenheft (PH). Der Aufbau der Dokumente wird in Tabelle 3 beschrieben

Hierbei ist bemerkenswert, daß in beiden Dokumenten die gleichen Kapitel vorkommen. Fast alle Kapitel sind im Pflichtenheft ausführlicher als im Lastenheft. Bei dem Lastenheft scheint es sich um eine Vorversion des Pflichtenhefts zu handeln. Ausnahmen sind das Kapitel »Gesamtübersicht« im Pflichtenheft, welches nur Verweise auf das gleiche Kapitel im Lastenheft enthält, und das Kapitel »Datenmodell«, welches nur im Lastenheft auftaucht. Im Pflichtenheft wird das Datenmodell durch die Angabe von Dateiformaten präzisiert. Das Datenmodell wird also aus zwei verschiedenen Blickwinkeln betrachtet: im Lastenheft werden die Ideen und der prinzipielle Aufbau beschrieben, im Pflichtenheft die zugehörigen Dateischnittstellen.

Kapitel	LH*	PH*
Allgemeines (Abk., Def., Literatur)	3	4
Projektkenndaten (Probleme, Lieferumfang, geplante Versionen)	2	2
Gesamtübersicht (Produktstruktur)	3	1
Vorgaben und Randbedingungen (Entwurfsvorgaben, nicht-fkt. Anforderungen, QS und CM-Plan)	3	8
Datenmodell (Metadaten)	11	-
Funktion der Werkzeuge	16	40
Schnittstellen (DB, Dateien, BS)	1	59
Benutzungsschnittstellen	1	43

Tabelle 3: Struktur von Lastenheft (LH) und Pflichtenheft (PH) des Entwicklungssystems (*: Umfang in Seiten)

Im Projekt B (graphisches Front-End) wurden ebenfalls zwei Spezifikationsdokumente erstellt: eine Requirements Specification und eine Functional Specification. Beide beschreiben das System, aus unterschiedlichen Blickwinkeln.

Die Requirements Specification enthält die Anforderungen aus Benutzersicht. In der Functional Specification wurde ein Modell des Systems erstellt. Die Funktionalität wird durch Szenarien beschrieben, das Datenmodell durch Klassen.

In Tabelle 4 werden Anzahl und Umfang der Anforderungen, Szenarien und Klassen aufgezählt. Angaben in Klammern stehen für Unterpunkte. So gibt es in den Dokumenten z.B. 43 funktionale Anforderungen. Diese haben insgesamt 68 Unterpunkte.

Im Gegensatz zu den beiden anderen Projekten, bei denen je ein Dokument als Ergebnis der Analyse- und Spezifikationsphase erstellt wurde, sind in diesem Projekt sehr viele kurze Dokumente entstanden. Jedes Dokument beschreibt genau einen Aspekt. Querverweise zwischen den einzelnen Dokumenten wurden teilweise durch Matrizen hergestellt. So gibt es z.B. eine Matrix, in der Ziele und Lösungen einander gegenübergestellt werden. Zu jeder Lösung wurde angegeben, welche der Ziele durch diese Lösung erreicht werden. Folgende Dokumente wurden erzeugt:

Kapitel	Anzahl	Seiten
Funktionalität		
- fkt. Anforderungen	43 (+68)	15,0
- fkt. Anf. an Version 2	5 (+4)	2,0
Szenarien	40	24,0
Klassen	31	32,0
Leistungsanforderungen	5 (+2)	0,5
Entwurfsvorgaben	4	0,5
Produktionsaspekte	6 (+10)	2,0
Wartungsaspekte	3	0,5
Sonstige Einschränkungen	3	0,5

Tabelle 4: Struktur von Requirements und Functional Specification des graph. Front-Ends

- Betroffene Organisationseinheiten (1 Seite),
- Ziele (3 Seiten),
- Probleme (1 Seite),
- Ist-Situation (1 Seite),
- Fkt. und nicht-fkt. Anforderungen (4 Seiten),
- Lösungsansätze (3 Seiten),
- Nutzen und Kosten (5 Seiten),
- Dienste des Systems (18 Seiten),
- Kommunikationsprotokoll (2 Seiten),
- Szenarien u. Benutzungsoberfläche (9 Seiten),
- DV-Ist-Zustand (4 Seiten),
- Daten- und Zugriffsschutz (1 Seite) und
- Datenmodell (28 Seiten).

3.2 Klassifikation der Informationen

Eine weitere Frage der Untersuchung war, welche Arten von Informationen in welchem Umfang in den Spezifikationen enthalten sind. Um diese Frage quantitativ zu beantworten, wurde ein Klassifikationsschema (siehe Tabelle 5) für Informationsarten aufgestellt und auf die Spezifikationen angewendet. Grundlage für das Klassifikationsschema waren Requirements-Engineering-Standards aus Dorfman (1990) und IEEE (1993).

In Tabelle 5 wird angegeben, welchen Umfang (in Zeilen[1] und in Prozent vom Gesamtumfang) die einzelnen Informationsarten in den Spezifikationen einnehmen. Können einem Absatz mehrere

1. Bei graphischen Notationen wurde die eingenommene Fläche in Zeilen umgerechnet.

Kategorie	Projekt A (Entw.-System)		Projekt B (Graph. Front-End)		Projekt C (Online-System)	
	Zeilen	%	Zeilen	%	Zeilen	%
Begriffsdefinitionen, Abkürzungen	178	2,4	29	0,7	20	1,1
Problembeschreibung, Ziele	134	1,8	116	2,6	116	6,1
Ist-Zustand	-	-	41	0,9	194	10,2
Lösungsansätze, Alternativen	-	-	-	-	65	3,4
Systemumgebung	46	0,6	20	0,5	50	2,6
Schnittstellen	3540	47,3	677	15,1	334	17,4
interne Struktur	1077	14,8	-	-	-	-
interner Zustand, Daten	213	2,9	1126	25,4	455	24,0
funktionale Anforderungen	1306	17,9	1861	42,1	510	26,9
nicht-funktionale Anforderungen	77	1,1	67	1,5	35	1,8
Entwurfseinschränkungen	101	1,4	43	1,0	-	-
Zukunft des Systems	18	0,2	47	1,0	-	-
Projektplanung, Kosten, Nutzen	134	1,8	133	3,0	106	5,6
Meta-Informationen	467	6,4	265	6,0	8	0,4

Tabelle 5: Klassifikation der Informationen in den Spezifikationsdokumenten

Kategorien zugeordnet werden, wurde jeder Kategorie ein entsprechender Anteil angerechnet.

Mit Meta-Informationen sind alle Arten von Informationen gemeint, die der Spezifikation zwar keine neuen Inhalte hinzufügen, aber für das Verständnis oder das Auffinden von Informationen notwendig oder hilfreich sind, z.B. Inhaltsverzeichnis, Index, Definition der verwendeten Notation usw.

Hierbei fällt auf, daß die Beschreibung der funktionalen Anforderungen, der internen Zustände und der Systemschnittstellen bei allen drei Systemen den größten Teil des Umfangs einnehmen (68,1%, 82,6% und 68,3%). Im Projekt A spielt zusätzlich die interne Struktur eine wichtige Rolle (14,8%). Das System wird also nicht als eine Black-Box betrachtet, deren interne Struktur »reine Entwurfssache« ist. Im Projekt C nimmt die Beschreibung des Ist-Zustands mit 10.2% einen wesentlichen Anteil ein.

3.3 Verwendete Notationen

Allen untersuchten Spezifikationen ist gemeinsam, daß hauptsächlich natürliche Sprache als Notation eingesetzt wurde. Andere Notationen wurden in geringerem Umfang ebenfalls verwendet.

In Tabelle 6 werden alle verwendeten Notationen aufgezählt. Für jedes Projekt wird angegeben, wie oft die Notation verwendet wird [1] und welchen Umfang (in Seiten und Prozent des Gesamtumfangs) die Notation einnimmt.

Folgende Notationen wurden als natürlichsprachlich klassifiziert: Freitext, numerierte und punktierte Aufzählungen und Überschriften. Bei Freitext mit Schlüsselwörtern wird jeder Absatz durch ein Schlüsselwort eingeleitet, z.B. *Name*, *Attribute* usw. Der Rest des Absatzes ist Freitext.

Hierbei fällt auf, daß in Projekt A sehr viele verschiedene Notationen verwendet wurden. Teilweise wurden unterschiedliche Notationen zur Beschreibung der gleichen Arten von Informationen eingesetzt (z.B. EBNF und Pseudocode zur Beschreibung des Aufbaus von Dateien). Andererseits gibt es Notationen, die für viele verschiedene Zwecke eingesetzt wurden (z.B. Baumdiagramme).

Die Spezifikation von Projekt B enthält mit Ausnahme von vier Skizzen nur natürliche Sprache (bzw. sehr ähnliche Notationen wie Freitext mit Schlüsselwörtern und Tabellen). Das gilt auch für die Beschreibung von Klassen und Szenarien.

In den Spezifikationen von Projekt C spielen Tabellen eine sehr wichtige Rolle. Die Tabellen wurden verwendet, um natürlichsprachliche Aussagen zu strukturieren und damit klarer lesbar zu machen.

In Tabelle 7 wird beschrieben, welche Informationen durch die in Tabelle 6 aufgelisteten Notationen ausgedrückt wurden.

1. Bei der natürlichen Sprache macht diese Angabe keinen Sinn. Es kann z.B. gezählt werden, wieviele Tabellen oder Skizzen verwendet werden, aber nicht, an wievielen Stellen natürliche Sprache verwendet wird.

Notation	Projekt A: (Entw.-System)			Projekt B (graph. Front-End)			Projekt C: (Online-System)		
	Anzahl	Seiten	%	Anzahl	Seiten	%	Anzahl	Seiten	%
natürliche Sprache (Freitext)	(-)	160,30	73,6	(-)	59,75	43,9	(-)	38,00	39,6
Freitext mit Schlüsselwörtern	-	-	-	31	25,00	18,2	18	22,5	18,0
hierarchische Aufzählung	32	10,75	5,0	-	-	-	-	-	-
Tabelle	30	8,75	4,1	85	50,25	36,9	80	31,5	39,4
Skizze	14	4,25	1,9	4	1,25	1,0	1	0,3	0,25
Bildschirmabzug	-	-	-	-	-	-	3	1,9	1,5
Baumdiagramm	22	9,00	4,0	-	-	-	1	0,3	0,25
Datenflußdiagramm	4	4,00	1,8	-	-	-	1	0,6	0,5
Flußdiagramm	1	1,00	0,5	-	-	-	-	-	-
Pseudocode	13	19,25	8,9	-	-	-	-	-	-
EBNF	1	0,50	0,2	-	-	-	-	-	-
ER-Diagramm	-	-	-	-	-	-	1	0,6	0,5

Tabelle 6 : Verwendete Notationen in den Spezifikationsdokumenten

4 Untersuchungsergebnisse - Spezifikationsprozeß

Die Spezifikationsprozesse wurden auf folgende Aspekte hin untersucht: Einsatz von Werkzeugen, Kommunikation mit den Kunden und Prüfung der Dokumente.

4.1 Werkzeugeinsatz

In jedem Projekt wurden Textverarbeitungsprogramme eingesetzt, um die Spezifikationen zu erstellen und zu verwalten. In Projekt C wurde zusätzlich eine Datenbank für einen Teil der Spezifikationen eingesetzt.

Textverarbeitungsprogramme bieten keine Unterstützung darin, die Arbeit zwischen verschiedenen Personen am gleichen Dokument zu koordinieren. Im Projekt A z.B. haben bis zu drei Personen gleichzeitig an den Dokumenten gearbeitet. Schreib-/Lesekonflikte wurden dadurch verhindert, daß jeder Bearbeiter für einen Teil des Dokuments zuständig war und in den anderen Teilen keine Änderungen vornehmen durfte.

In allen Projekten wurden Vorlagen (Templates) eingesetzt, in denen festgelegt wird, in welcher Reihenfolge welche Kapitel vorkommen sollen, welche Arten von Informationen in welche Kapitel gehören und welche Notationen verwendet werden sollen. Die Spezifizierer in den Projekten A und C kannten sich schon zu Projektbeginn mit den Werkzeugen und Vorlagen aus. Projekt B jedoch war das erste Projekt in der Firma, in dem Szenarien eingesetzt wurden. Laut Aussagen der Projektverantwortlichen hat sich der Ansatz bewährt, auch wenn z.B. beim Management der Szenarien Verbesserungen denkbar sind.

4.2 Kommunikation mit den Kunden

In allen Projekten spielte die Kommunikation mit den Kunden eine wichtige Rolle. In Projekt A gab es insgesamt zehn zweitägige Treffen am späteren Einsatzort, der mehrere hundert Kilometer vom Entwicklungsort entfernt liegt. In Projekt B wurde der Aufwand für die Kommunikation mit den Kunden auf 5 - 10h pro Woche geschätzt. Der Gesamtkommunikationsaufwand der Projektmitglieder mit den Kunden wird auf max. 10% des Gesamtaufwands geschätzt. In Projekt C war neben den Kunden der Datenmodellierer als Informationsquelle an dem Projekt beteiligt. Aufgabe des Datenmodellierers ist es, das unternehmensweite Datenmodell zu verwalten und zu pflegen.

4.3 Prüfung der Dokumente

Das Lastenheft von Projekt A wurde nicht formal geprüft. Für das Pflichtenheft wurde ein Review veranstaltet, an dem vier Personen aus dem Anwendungsbereich und das Entwicklerteam beteiligt waren. Die Benutzungsoberfläche wurde von den gleichen Personen anhand des Prototypen geprüft. Nach dem Review wurde das Pflichtenheft entsprechend angepaßt. Alle Fehler, die danach in dem Pflichtenheft gefunden wurden, wurden in Form von Änderungsmeldungen festgehalten. Das Pflichtenheft selbst wurde nicht mehr geändert.

Die Requirements Specification von Projekt B wurde von den Kunden geprüft. Da sich die meisten Prüfer in anderen Ländern aufhielten, wurden die Dokumente an die Prüfer verschickt. Die Befunde wurden an das Entwicklungsteam zurückgeschickt. Die Functional Specification wurde durch ein Review des Entwicklerteams ohne Betei-

Notation	Information, die durch die Notation ausgedrückt wird
Freitext mit Schlüsselwörtern	Beschreibung von Klassen interner Zustand
hierarch. Aufz.	hierarchischer Aufbau von Dialogfenstern
Tabelle	Aufwandsschätzung, Projektplan, CM- und QS-Plan; Problem-, Ziel- und Ist-Analyse; Begriffsdefinitionen, Abkürzungen; Systemumgebung; Lösungsansätze; fkt/nicht-fkt Anforderungen, Mengengerüst; Änderungen von Anforderungen; Schnittstellen, Aufbau von (Objekt-)Menüs; Liste von Dialogfenstern; Zusammenhang zwischen Szenarien und Klassen; Plausibilitätsregeln für Attribute
Skizze	Beispiele für die graphische Eingabe; prinzipieller Aufbau der graphischen Ausgabe; Architekturvorgaben; funktionale Anforderungen
Bildschirmabzug	Benutzungsschnittstelle
Baumdiagramm	Hierarchien von Zielen, Funktionen, Programm- und Modell-Komponenten; Verzeichnisstruktur; Aufbau von Dateien; Zuordnung von Programmkomponenten zu Funktionen; Zuordnung von Funktionen zu Programmkomponenten
Datenflußdiagr.	Zusammenspiel zwischen den Komponenten des Ist- und Soll-Zustands
Flußdiagramm	Beschreibung von Algorithmen
Pseudocode	Aufbau einer Datei (in Form eines unvollständigen Beispiels)
EBNF	Aufbau einer Datei
ER-Diagramm	Interner Zustand

Tabelle 7 : Informationen, die durch die Notationen ausgedrückt wurden

ligung des Kunden geprüft. Bei den Prüfungen wurden eine Reihe von Fehlern gefunden. Diese wurden jedoch nicht in die Spezifikationen integriert. Ebenso wurden Anforderungsänderungen, die im weiteren Projektverlauf aufgetreten sind, in den Spezifikationen nicht berücksichtigt.

Jedes in Projekt C erstellte Dokument wurde von den späteren Benutzern aus dem Fachbereich geprüft, das ER-Datenmodell wurde zusätzlich von dem Datenmodellierer abgenommen. Bei den Prüfungen handelte es sich um informale Prüfungen und Abnahmen, die parallel zur Erstellung der Dokumente vorgenommen wurden. Nach Abschluß der Spezifikationsphase wurden die Dokumente nicht mehr geändert.

5 Konsequenzen

In diesem Kapitel werden einige Konsequenzen beschrieben, die aus der Untersuchung gezogen werden können.

5.1 Kommunikation mit den Kunden

Obwohl in allen Projekten eine klare Kommunikationsschnittstelle zwischen dem Entwicklungsteam und den Kunden definiert war, sind Schwierigkeiten aufgetreten. So gab es z.B. in Projekt A das Problem, daß eine Entwicklungsumgebung realisiert werden sollte, welche ein bestimmtes Programmierparadigma unterstützt und auch erzwingt. Die Konsequenzen, die sich dadurch auf die Arbeitsprozesse der Benutzer ergaben, waren nicht gründlich durchdacht. Darüberhinaus gab es Akzeptanzprobleme bei der Auslieferung. So war z.B. im Pflichtenheft eine textuelle Benutzungsschnittstelle vereinbart, mit der die Benutzer trotz des GUI-Prototyps nicht zufrieden waren. Hieraus folgt, daß die Kunden noch stärker in den Spezifikationsprozeß einbezogen werden müssen. Ein Indiz dafür, daß die Kunden sich nicht ausreichend mit den Spezifikationen auseinandergesetzt haben, ist, daß in dem Review viele formale, aber nur wenige inhaltliche Fehler gefunden wurden.

5.2 Änderungsmanagement

Eines der größten Probleme bei der Projektdurchführung war, daß sich auch nach Abschluß der Spezifikationsphase Anforderungen änderten und neue Anforderungen hinzukamen. Trotzdem wurden die Spezifikationen nicht an die Änderungen angepaßt. Zwar wurden in Projekt A Änderungsmeldungen gesammelt, sie wurden aber nicht in die Dokumente eingearbeitet. Es ist sehr schwer festzustellen, ob diese Änderungen nicht doch weitreichende Konsequenzen für das zu entwickelnde System hatten.

Die Konsequenz daraus ist, daß die realisierten Systeme nur unzureichend durch die Spezifikationen beschrieben werden. Durch folgende Maßnahmen können die Probleme vermindert werden:

- Es muß ein Verfahren zur Initiierung und Durchführung von Änderungen an den Spezifikationen und zur Behandlung von evtl. daraus resultierenden Konflikten eingeführt werden.

- Es muß ein Konfigurationsmanangement eingeführt werden. Dies ist insbesondere dann notwendig, wenn es sich bei den Spezifikationen um viele einzelne Dokumente handelt, die in unterschiedlichen Versionen vorliegen können.

5.3 Fehlende Informationen

Folgende Informationen wurden in den Spezifikationen teilweise zu knapp beschrieben:

- Begriffsdefinitionen: Das Begriffslexikon ist ein für das gesamte Projekt zentrales Ergebnis der Spezifikationsphase. In dem Begriffslexikon wird die für das Projekt relevante Terminologie festgelegt. Neben den für das System relevanten technischen Begriffen und Abkürzungen müssen auch Begriffe aus dem Anwendungsbereich definiert werden. Die Begriffsdefinitionen sollten in einem Kapitel gesammelt und nicht erst an den Stellen eingetragen werden, an denen sie zum ersten Mal verwendet werden.
- Nicht-funktionale Anforderungen: Zu den nicht-funktionalen Anforderungen gehören u.a. Aussagen über das Laufzeitverhalten, Datenmengen und Datengrößen. Im Projekt B fehlten z.B. Aussagen über die Komplexität der zu verwaltenden Daten. Sie wurde von den Entwicklern zu Beginn deutlich unterschätzt. Dies führte zu ernsthaften Performance-Problemen im weiteren Projektverlauf.
- Objekte der Systemumgebung: Zur Systemumgebung gehören u.a. die unterschiedlichen Benutzer, die mit dem System interagieren. Informationen über die Benutzer können z.B. herangezogen werden, um zu prüfen, ob die Benutzungsschnittstelle für Benutzer mit den dokumentierten Vorkenntnissen geeignet ist.
- Beschreibung des Ist-Zustands: In vielen Projekten soll ein vorhandenes System durch ein neues System ersetzt werden. In Spezifikationen zu solchen Systemen kommen oft Anforderungen der Art »so wie bisher, aber ...« vor. Solche Anforderungen sind sinnvoll. Um sie verstehen zu können, müssen jedoch die Mängel und die wichtigsten Eigenschaften des zu ersetzenden Systems beschrieben werden.
- Zu jeder Anforderung sollte angegeben werden, von wem die Anforderung erhoben wurde. Wenn Konflikte erkannt werden, dann können alle betroffenen Personen identifiziert und an der Lösung des Konflikts beteiligt werden.
- Während der Erstellung der Spezifikation kann es vorkommen, daß zu einem bestimmten Zeitpunkt noch einige Punkte ungeklärt sind. Es ist nicht immer möglich, das sofort zu ändern. weil z.B. noch bestimmte Informationen eingeholt werden müssen. Diese offenen Punkte sollten jedoch explizit dokumentiert sein, z.B. durch eine einheitliche Markierung im Text und einen Index, um sie schnell auffinden zu können. Es reicht nicht aus, wenn gut versteckt im Fließtext steht, daß an einigen Stellen noch Informationen nachzutragen sind.

5.4 Informale oder formale Notation?

In den untersuchten Spezifikationen konnten Effekte beobachtet werden, die typisch für natürlichsprachliche oder informale Spezifikationen sind:

- Anforderungen sind unpräzise formuliert.
- Es treten Konflikte in scheinbar nicht zusammenhängenden, weit auseinanderliegenden Textstellen auf.
- Anforderungen werden nicht klar von anderen Informationen wie z.B. Erklärungen getrennt.
- Anforderungen können nicht klar von anderen Anforderungen getrennt werden.

Eine Möglichkeit, diese Probleme zu umgehen besteht darin, halb-formale oder formale Notationen zu verwenden. Jedoch kam auf die Frage, ob sich die Projektbeteiligten einen verstärkten Einsatz halb-formaler oder formaler Notationen vorstellen könnten, ein klares *Nein* als Antwort. Hauptgründe dagegen waren:

- sehr hoher Ein- bzw. Umstiegsaufwand,
- sehr hoher Schulungsaufwand,
- fehlende Vorstellung, inwiefern (halb-)formale Methoden von Nutzen sein können (Projekt A),
- kein Bedarf – die bisherige Vorgehensweise ist auch für größere Projekte geeignet (Projekt C).

In der Firma, die Projekt A durchführte, wurden objektorientierte Analysemethoden (Yourdon, 1991) und SA (DeMarco, 1982) evaluiert. Die Evaluation führte jedoch zu dem Ergebnis, daß diese Methoden in dem Projektumfeld nicht in Frage kommen.

Durch folgende Maßnahmen kann man die o.g. Probleme besser in den Griff bekommen, ohne auf die informalen Notationen zu verzichten:

- Es sollten Anleitungen und Checklisten erstellt und verwendet werden, durch die o.g. Probleme aufgedeckt und vermindert werden können.
- Die Spezifikationen müssen auf die o.g. Probleme geprüft werden. Diese Prüfung kann intern im Entwicklerteam durchgeführt werden. Die inhaltlichen Prüfungen durch die Kunden können hierdurch natürlich nicht ersetzt werden.

Literatur

Coad, P.; Yourdon, E.: Object-Oriented Analysis. 2nd edition. Prentice Hall, 1991.

DeMarco, T.: Structured Analysis and System Specification. Prentice Hall, 1982.

Dorfman, M.; Thayer, R.: Standards, Guidelines and Examples on System and Software Requirements Engineering. IEEE Computer Society (New York), 1990.

Gaudel, M.-C.: Formal Specification Techniques. Proceedings of the 16th International Conference on Software Engineering. 16. - 21. Mai 1994, Sorrento (Italien), 1994, S. 223-227.

IEEE: IEEE Recommended Practice for Software Requirements Specifications. IEEE STD 830-1993, IEEE Computer Society (New York), 1993.

Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language, Documentation Set 1.0, Rational Software Corporation, Santa Clara, 1997.