

MAGAZIN FÜR SOFTWARE-ENTWICKLUNG

sw development

SCHWERPUNKT:

E-COMMERCE VS. M-COMMERCE

SW-INTERVIEW MIT ERIK KICK, FEEDBACK AG

SCHWERPUNKTTHEMEN:

MEGATREND E-BUSINESS – WER HAT EIN E-IMAGE?

MOBILE COMPUTING – SIND SIE BEWEGLICH GENUG

TIPPS & TRICKS:

MOGELPACKUNG:

WENN DER HERSTELLER
SEINE PRODUKTE
VERSCHENKT

FACHTHEMEN:

KOMPLEXE SYSTEME

SICHER SPEZIFIZIEREN

UMSTEIGEN VON COBOL AUF OOP

REQUIREMENTS MANAGEMENT –
ANFORDERUNGEN IM GRIFF

ISSN 1616-4970

SONDERDRUCK FÜR

applied technologies GmbH

Kruppstraße 82-100, D-45145 Essen, Tel.: 02 01 / 8 12 73 34

REQUIREMENTS MANAGEMENT –

Fehlendes Requirements Engineering und Management führen oftmals zum Scheitern von Projekten. Nicht unberechtigt haben sich das Requirements Engineering und -Management als „best practices“ herauskristallisiert. Dieser Artikel soll aufzeigen, welches Potenzial das Anforderungsmanagement birgt und wie es mit einfachen Mitteln zu einer soliden Grundlage für eine erfolgreiche Projektentwicklung führen kann.

Die folgende Situation wird Ihnen bekannt vorkommen: In diversen Projektmeetings mit dem Auftraggeber versucht der Entwickler Verständnis für die Anforderungen des Kunden zu gewinnen. Schnell entsteht eine Vorstellung davon, wie ein Lösungsansatz aussehen könnte. Der Entwickler zieht sich mit seinen Aufzeichnungen zurück und erstellt ein Lösungskonzept, das dem Auftraggeber zur Abstimmung vorgelegt wird (siehe **Abbildung 1**).

Der Auftraggeber bewertet das Konzept und stellt fest, dass es nicht seinen Anforderungen entspricht. Das Lösungskonzept wird überarbeitet und erneut vorgelegt; dieser Vorgang kann sich beliebig häufig wiederholen. Endlich wird – so meint man – ein gemeinsames Verständnis erreicht, jedoch geht der Auftraggeber implizit davon aus, dass mit einzelnen, sprachlich ungenauen Aussagen des Lösungskonzepts verschiedene weitere Anforderungen abgedeckt werden.

Erst während der Realisierung des Konzepts, spätestens aber nach Auslieferung der Software, stellt sich heraus, dass der Lösungsvorschlag nicht vollständig ist, Anforderungen falsch verstanden oder auch gar nicht berücksichtigt wurden. Da aber die konkreten Anforderungen bis dahin nicht dokumentiert wurden, ist die Situation für beide Seiten unbefriedigend. Durch gegenseitige Schuldzuweisungen erzeugt man möglicherweise ein für den weiteren Projektverlauf schädliches Klima.

Hier hat sich das Requirements Engineering als „best practice“ durchgesetzt, um ein gemeinsames Verständnis von Anforderungen zu erreichen und damit letztendlich

auch negative Einwirkung auf das Projekt zu verhindern – ein nicht zu unterschätzender Faktor auf dem Weg zu einem erfolgreichen Projektabschluss!

Abbildung 2 zeigt den Weg zur Erstellung eines Lösungskonzepts auf der Basis einer Anforderungsspezifikation. Durch die Trennung von Anforderungen und Lösungskonzept (Anwendungsfälle, Objekt- und/oder Datenmodell etc.) ist ein Umdenken bei Auftraggeber und IT-Spezialisten notwendig, denn zunächst wird mit der Anforderungsspezifikation auf ein gemeinsames Verständnis hin gearbeitet.

Gause und Weinberg waren die ersten, die dem Requirements Engineering entsprechende Bedeutung beimessen [Gau89]. Mit den aktuellen Veröffentlichungen zum Thema (z. B. [Wie99], [Lef00] und [Rup01a]) werden einfach anzuwendende Vorgehensweisen zum Ermitteln und Verwalten von Anforderungen beschrieben.

Eine software-technische Unterstützung für die Erfassung, Dokumentation und das Management von Anforderungen ist dringend angeraten. Für kleinere Systeme mit einem überschaubaren Umfang an Anforderungen kann dies unter Zuhilfenahme eines Tabellenkalkulations- oder Textverarbeitungssystems erfolgen (siehe **Abbildung 3**).

Während [Gau89], [Rup01a] und [Rup01b] u. a. Wege zeigen, die durch das Requirements Engineering zu einem gemeinsamen Verständnis von korrekten, vollständigen Anforderungen führen, werden im Folgenden die Attribute einer Anforderung und deren Verwendung als zentrales Instrument des Requirements Management beschrieben.



Abbildung 1: der herkömmliche Weg

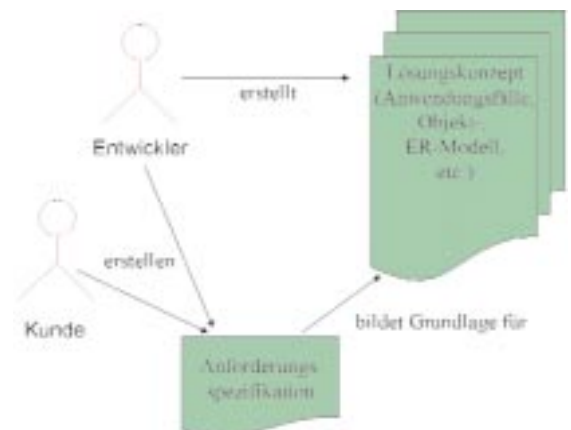


Abbildung 2: Requirements als Basis des Lösungskonzepts

ANFORDERUNGEN IM GRIFF

Feature	Identität	Beschreibung	Priorität	Quelle	Erst.-Datum	Autor	Status	Komplexität	Stabilität	Version	letzte Änd. am
Stationsauswertung	SA-001	Das System soll die in einem bestimmten Zeitraum gemessene Menge einer Station unter Verwendung der Berechnungsverfahren, die die Meßtechnik vorgibt, bestimmen. (Verfahren siehe in "Meßtechnik - Berechnungsverfahren.doc")	hoch	Protokoll v. 22.01.01 Punkt 2, 3, 4; Herr Schulze	13.02.01	Leibecke	abgestimmt	mittel	hoch	1	
	SA-002	Eine Station soll von mehreren Verträgen abgerechnet werden können. Wird eine Station über mehrere Verträge abgerechnet, so sollen die Verträge optional unterschiedliche Berechnungsverfahren zur Mengenermittlung bereitstellen.	hoch	Protokoll v. 22.01.01 Punkt 2; Herr Schulze	13.02.01	Leibecke	abgestimmt	mittel	hoch	1	
	SA-003	Das System soll Stationsauswertungen über beliebige Zeiträume durchführen (z. B. Auswertung für einen Kalendermonat, Auswertung für eine Kalenderwoche, beliebiger Beginn, beliebiges Ende)	hoch	H. Resser	15.02.01	Gerwert	abgestimmt	mittel	hoch	1	
	SA-004	Das System soll den Zugriff auf die vertraglichen Berechnungsverfahren über das Berechtigungskonzept schützen.	hoch	H. Resser	15.02.01	Gerwert	abgestimmt	niedrig	hoch	2	05.03.01

Abbildung 3: Anforderungsattribute

In einem systematischen Requirements Management kann jede Anforderung mit den folgenden Attributen dokumentiert werden:

- Feature: Funktionale Anforderungen werden nach Charakteristiken des Systems organisiert, z. B. „Grafische Auswertung“ oder „E-Mail Integration“.
- Identität: Jede Anforderung erhält einen eindeutigen Bezeichner. Dieser kann sich aus einem Feature-Kürzel und einer laufenden Nummer zusammensetzen, z. B. „GA-001“ (grafische Auswertung, lfd. Nr. 1).
- Die Identität wird u. a. für die Nachvollziehbarkeit von Anforderungen verwendet.
- Beschreibung: Die ausformulierte Anforderung, die verschiedene Charakteristiken erfüllen muss. Sie muss vollständig, korrekt, machbar, notwendig, eindeutig, priorisier- und prüfbar sein.
- Priorität: Die Priorität dient zur Festlegung der dem Kunden am wichtigsten erscheinenden Anforderungen.
- Quelle: Wer stellte diese Anforderung? (z. B. „Protokoll vom 23.05.01, Herr Meier“)
- Erstellungsdatum: Wann wurde die Anforderung dokumentiert?
- Autor: Wer hat die Anforderung dokumentiert?
- Status: Erlaubt die Nachverfolgung einer Anforderung; eine Anforderung kann die folgenden Stadien durchlaufen: „vorgeschlagen“, „abgestimmt“, „implementiert“, „geprüft“, „gelöscht“.
- Eine „gelöschte“ Anforderung sollte in der Anforderungsdokumentation bestehen bleiben; damit dient das Dokument zur Pflege der Projekthistorie.
- Komplexität: Schwierigkeitsgrad der Implementierung („hoch“, „mittel“, „niedrig“).

- Stabilität: Die Stabilität beschreibt die Wahrscheinlichkeit, dass eine Anforderung im Laufe der Zeit stabil bleibt („hoch“, „mittel“, „niedrig“).
- Version: Die Versionsnummer einer Anforderung.
- letzte Änderung am: Datum der letzten Änderung.
- letzte Änderung durch: Autor der Änderung.

Abhängig von Größe und Komplexität des Projekts können durchaus weitere Attribute zu einer Anforderung dokumentiert werden, z. B. das Subsystem, dem eine Anforderung zugeordnet wird, das Release, mit dem eine Anforderung realisiert werden soll, Kriterien für einen Akzeptanztest etc..

Mit diesem Dokument steht dem Projektmanager ein mächtiges Werkzeug mit großem Potenzial zur Verfügung; werden die Anforderungen gar in einer Datenbank-Tabelle verwaltet, stehen beliebige Auswerte- und Sortiermöglichkeiten zur Verfügung (z. B. „alle Anforderungen mit geringer Stabilität“, „alle nicht implementierten Anforderungen“ etc.).

Mit einem Anforderungsdokument existiert ein zentrales Instrument, das alle Anforderungen an das zu erstellende System aufnimmt. Da sich die Anforderungen an ein System bekanntermaßen ändern, handelt es sich um ein „lebendes“ Dokument.

Die üblichen Probleme, die durch die Verteilung der Anforderungen auf verschiedene Dokumente (Anwendungsfälle, Constraints im Objekt- und Datenmodell, Protokolle, Notizen von Projektmitarbeitern, nicht zu vergessen die gar nicht schriftlich fixierten Anforderungen etc.) entstanden, gehören damit der Vergangenheit an. Neuen Mitarbeitern ermöglicht das Dokument einen einfacheren Einstieg in das Projekt und Fluktuation im Team ist einfacher zu verkraften.

Die Priorität einer Anforderung kann ebenso wie die Stabilität zur Steuerung der Realisierungsreihenfolge herangezogen werden. Wird z. B. das Timebox-Verfahren ([Mar91]) angewendet, können in Abstimmung mit ▶

dem Kunden bestimmte Anforderungen aus einem Release herausgenommen werden, um den angestrebten Termin einhalten zu können, anstatt eine Terminverschiebung durchzuführen.

Weisen Anforderungen Qualitätsmängel auf (unvollständig, fehlerhaft etc.), dient der Hinweis auf die Quelle zur schnellen Fehlerbehebung. Oftmals findet sich zu einer Anforderung kein Verantwortlicher; das Requirements-Dokument hilft bei der „Fahndung“ nach dem Verursacher.

Die im erstgenannten Beispiel aufgeführten Iterationen in der Abstimmung von Lösungsansätzen werden nahezu vollständig entfallen, denn die strikte Trennung zwischen Anforderungsdokumentation und Lösungskonzept ermöglicht die Erstellung einer Nachvollziehbarkeitsmatrix, über die sichergestellt werden kann, dass alle Anforderungen im Lösungskonzept berücksichtigt werden.

Die Verwaltung expliziter, vollständiger Anforderungen erlaubt die effiziente Erstellung von Abnahmespezifikation und Testfällen.

Unter dem Druck des time-to-market und dem Zwang, Projekte „in time“ und „in budget“ abzuwickeln, sollte es unbedingt vermieden werden, dass Entwickler „goldene Klinken“ in ein System hineinbauen. Eine der wichtigsten Fragen des Projektmanagers zur Einhaltung der Disziplin seiner Entwickler wird lauten „Auf welche Anforderung hin wird dieses Stück Software entwickelt?“. Ohne ein entsprechendes Requirements Management ist auch dies ein schwieriges Unterfangen.

FAZIT

Unabhängig von der Technik zur Ermittlung der Anforderungen (Anwendungsfälle, Interviews, Beobachtung etc.) stellt das Requirements Management ein mächtiges Werkzeug zur Projektsteuerung und Sicherstellung einer hohen Qualität dar.

Der Aufwand zur Dokumentation von Anforderungen in kleinen Systemen ist durch die Einfachheit der Requirements-Struktur vertretbar; die damit zu erzielenden Erfolge rechtfertigen allerdings auch einen deutlich höheren Aufwand.

In größeren Systemen bietet sich jedoch der Einsatz professioneller Requirements Management Software an (z. B. Rational RequisitePro [Rat], DOORS [Doo] oder Caliber-RM [Cal]).

Dass ein professionelles und systematisches Requirements Engineering und Management erforderlich ist, scheint unumstößlich, denn wer in seinem Projekt jemals Requirements Engineering und -Management-Praktiken angewendet hat, wird sich fragen, wie er zuvor ohne diese Technik auskommen konnte.

Dietmar Leibecke (E-Mail: dleibecke@appliedtechnologies.de) ist Geschäftsführer der applied technologies GmbH, Essen. Er beschäftigt sich seit vielen Jahren mit objektorientierter Softwareentwicklung und Software Engineering Themen.

VERWEISE

- [Cal] Technology Builders Incorporated; Caliber-RM <http://www.tbi.com/>
- [Doo] Telelogic DOORS Enterprise Requirements Suite <http://www.telelogic.com>
- [Gau89] Donald C. Gause, Gerald M. Weinberg: »Exploring Requirements: Quality before Design«, Dorset House 1989
- [Lef00] Dean Leffingwell, Don Widring: »Managing Software Requirements. A Unified Approach«, Addison-Wesley 2000
- [Mar91] James Martin: »Rapid Application Development«, Macmillan Publishing Company 1991
- [Rat] Rational Software: RequisitePro; <http://www.rational.com/products/reqpro/index.jsp>
- [Rup01a] Chris Rupp: »Requirements-Engineering und -Management«, Carl Hanser Verlag 2001
- [Rup01b] Chris Rupp, Jürgen Dallner: »Mustergültige Anforderungen«, Objektspektrum Mai/Juni 2001
- [Wie99] Karl Wiegers: »Software Requirements«, Microsoft Press 1999