

© Copyright Galileo Press 2002

Georg Erwin Thaller

# Software-Anforderungen für Webprojekte

Pflichtenheft, Lastenheft, Prozesse



# Inhalt

Vorwort 9

Danksagung 11

---

1	Bedeutung der Spezifikation	13
1.1	Software-Fehler	15
1.2	Warum scheitern Software-Projekte?	28
1.3	Machen wir Fortschritte?	30
2	Die Spezifikation im Entwicklungsprozess	33
2.1	Standard-Software oder individuelle Entwicklung	33
2.2	Machbarkeitsstudie	35
2.3	Projektmanagement und Spezifikation	39
2.3.1	Zeitplan und Aufwand	40
2.4	Software als Subsystem	43
2.5	Vorgehensmodelle	45
2.6	Die Phase im Detail	53
2.6.1	Modell des IEEE	53
2.6.2	Modell der NASA	55
2.6.3	Software Engineering Lab	57
2.6.4	Volere-Modell	60
2.6.5	Modell von Young	62
2.6.6	Die Spezifikation im Rahmen des Capability Maturity Models	71
2.6.7	Ziele	72
2.6.8	Fähigkeit zur Leistung	72
2.6.9	Durchzuführende Tätigkeiten	73
2.6.10	Messung und Analyse	74
2.6.11	Überprüfung der Implementierung	74
2.6.12	Der eigene Prozess	75
2.7	Wer sind die Stakeholder?	76
2.8	Wer soll die Spezifikation erstellen?	77
2.9	Spezifische Software	80
2.10	Rolle des Managements	81
2.11	Risiken	82
2.12	Vorteile einer guten Spezifikation	86

---

<b>3</b>	<b>Der Inhalt der Spezifikation</b>	<b>89</b>
3.1	Inhalte finden	89
3.1.1	Ableitung aus der System-Spezifikation	89
3.1.2	Kunde und Anwender	91
3.1.3	Innovationen und neue Technologie	103
3.2	Tücken der Sprache	105
3.3	Der Inhalt im Detail	105
3.3.1	Funktionen	107
3.3.2	Nichtfunktionelle und abgeleitete Anforderungen	110
3.3.3	Daten	119
3.3.4	Leistungsmerkmale	120
3.3.5	Ereignisse	121
3.3.6	Prozesse	121
3.3.7	Qualitätsattribute	122
3.3.8	Fehlerbehandlung	122
3.3.9	Restriktionen	123
3.3.10	Weiteres Material	124
3.4	Forderungen an eine gute Spezifikation	125
3.5	Grafiken	128
3.6	Mit Erfolg schreiben	129
3.7	Gliederung und Struktur	131
3.8	Operationelles Konzept	138
3.9	Abgeleitete und verwandte Dokumente	142
3.9.1	Die Schnittstellen-Spezifikation	143
3.9.2	Die Test-Spezifikation	144
3.9.3	Planungsinstrumente	145
3.10	Die Website	147
<b>4</b>	<b>Verifikation und Validation</b>	<b>161</b>
4.1	Qualitätsmanagement und Definitionsphase	161
4.2	Metriken zur Spezifikation	162
4.3	Reviews	165
4.4	Änderungen im Entwicklungsverlauf	168
4.5	Den Prozess beherrschen	170

---

---

5	Der Übergang zum Design	171
5.1	Vom Abstrakten zum Konkreten	171
5.2	Kreative Entwickler und Analysten	172
5.3	Design als Experiment	173
5.3.1	Konzeptuelles Design	173
5.3.2	Prototypen bauen	178

A	Anhang	181
A.1	Literaturverzeichnis	181
A.2	Verzeichnis der Akronyme	183
A.3	Glossar	186
A.4	Produktmuster	187
A.5	Fragebögen	242
A.6	Ressourcen im Internet	245

	Index	247
--	-------	-----

## 4 Verifikation und Validation

*Nachdem ein erster Entwurf der Software-Spezifikation vorliegt, müssen wir uns natürlich fragen, wie gut dieses Dokument ist. Sind wirklich alle Forderungen des Kunden und der Anwender erfasst? Auf der anderen Seite dürfen wir die Leute nicht vergessen, die das Dokument in Programmcode umsetzen sollen: Ist es realistisch, was da zu Papier gebracht wurde? Lässt es sich umsetzen? Wichtiger noch: Lässt sich das Projekt mit dieser Spezifikation in der vorgegebenen Zeit realisieren?*

### 4.1 Qualitätsmanagement und Definitionsphase

In der ersten Phase des Projekts werden Weichen gestellt, und deshalb ist es wichtig, dass das Qualitätsmanagement von Anfang an dabei ist. Wenn in diesen Wochen und Monaten etwas falsch läuft, kann diese Fehlentwicklung später oft nur mit erheblichem Aufwand korrigiert werden.

Wenn das Qualitätsmanagement der Firma sich als Anwalt des Kunden versteht, muss es dafür sorgen, dass der Kunde und die Anwender im Entwicklungsprozess angemessen vertreten sind. Falls der Kunde ein eigenes Qualitätsmanagement besitzt, sollte der Kontakt mit diesen Fachleuten gesucht werden.

Der Prozess zur Erstellung der Spezifikation wird bereits im QM-Handbuch definiert sein. Falls es bei dem Projekt spezifische Besonderheiten zu berücksichtigen gibt, ist jetzt der Zeitpunkt gekommen, um im Sinne von Tailoring den Prozess anzupassen.

Das Qualitätsmanagement kann in dieser Zeit, wenn es um ein umfangreiches Projekt geht, den Vertrag mit dem Kunden überprüfen. Bei großen EDV-Vorhaben ist das manchmal ein Dokument, das mehrere Ordner umfasst. In der Hektik vor der Vertragsunterzeichnung können da leicht Klauseln in den Vertrag geraten sein, die sehr schwer zu erfüllen sind. Bei großen Firmen oder Konzernen arbeiten manchmal mehrere Abteilungen dem Projektmanager zu, und dieser kann nicht Spezialist auf allen Gebieten sein. Deshalb ist nicht ausgeschlossen, dass einzelne Vertragsbestimmungen sich widersprechen.

Der Planungsprozess sollte nun aktiv beginnen, und das bezieht sich in erster Linie auf den Entwicklungsplan. Das Qualitätsmanagement [14,21] wird auch fragen, ob alle Ressourcen zur Verfügung stehen. Oder sind im Vertrag Zusagen enthalten, die nicht zu erfüllen sind?

Für die Bewertung der Produkthanforderungen müssen geeignete Maßnahmen geplant und durchgeführt werden, etwa Reviews. Hier wird das Qualitätsmanage-

ment einen wesentlichen Anteil haben. Nicht zuletzt muss über die Bewertung der Anforderungen ein Protokoll angefertigt werden. Dies ist primär eine Aufgabe der Entwicklung, aber das Qualitätsmanagement als unabhängige Instanz sollte seine eigenen Aufzeichnungen erstellen.

Management im Bereich der Software-Entwicklung gleicht einem Herumirren im Nebel, wenn das Management seine Entscheidungen nicht auf Zahlen stützen kann. Deshalb sollten bereits in dieser ersten Phase der Entwicklung Kennzahlen erfasst und dazu Metriken [22] erstellt werden.

## 4.2 Metriken zur Spezifikation

*You can't control what you can't measure.*

*Tom deMarco*

In der Neuauflage der DIN EN ISO 9001 vom Dezember 2000 hat der Bereich Messung, Analyse und Verbesserung eine bedeutende Aufwertung erfahren. Kennzahlen zur Bewertung der Qualität von Software in allen ihren Ausprägungen und in allen Phasen der Entwicklung sind jetzt unumgänglich.

In der Wahl der Metriken, die ein Unternehmen einsetzt, ist es natürlich frei. Es muss deshalb das Ziel des Managements sein, solche Metriken auszuwählen, die mit einem Minimum an Aufwand zu Einsichten in den Entwicklungsprozess führen.

Die ersten Wochen eines Projekts sind oftmals ziemlich chaotisch. Es kommen Mitarbeiter aus vielen Bereichen des Unternehmens zusammen, die vorher nie miteinander gearbeitet haben und unterschiedliche Erfahrungen mitbringen. Zwar scheint der Auslieferungstermin für die Software noch weit in der Ferne zu liegen, aber ein erfahrenes Management weiß, dass Zeit, die jetzt verschwendet wird, später nicht mehr einzuholen ist.

In dieser Hektik müssen mit Hilfe des Kunden und der Anwender die Software-Anforderungen gefunden und zu Papier gebracht werden. Dabei werden im Dokument am Anfang notwendigerweise Lücken sein, weil ein Gesprächspartner im Interview mit dem Autor der Spezifikation zu einem bestimmten Punkt nichts weiß oder weil aus anderen Gründen Informationen derzeit nicht verfügbar sind.

Am Anfang weist eine Spezifikation immer weiße Flecken auf, die der Autor mit dem Kürzel TBD (*to be defined*) kennzeichnet. Das ist Routine, aber im Lauf der Wochen müssen diese TBDs durch Inhalte ersetzt werden.

Für das Management ist die Zahl der TBDs im Dokument ein guter Gradmesser für die Vollständigkeit der Spezifikation. Deshalb sollte daraus eine Metrik gemacht werden. Wenn wir davon ausgehen, dass die Phase »Ermittlung der

Anforderungen« in unserem Fall 16 Wochen dauert, und wir jede Woche die Zahl der TBDs im Dokument messen, können wir die folgende Metrik ermitteln (siehe Abbildung 4.1).

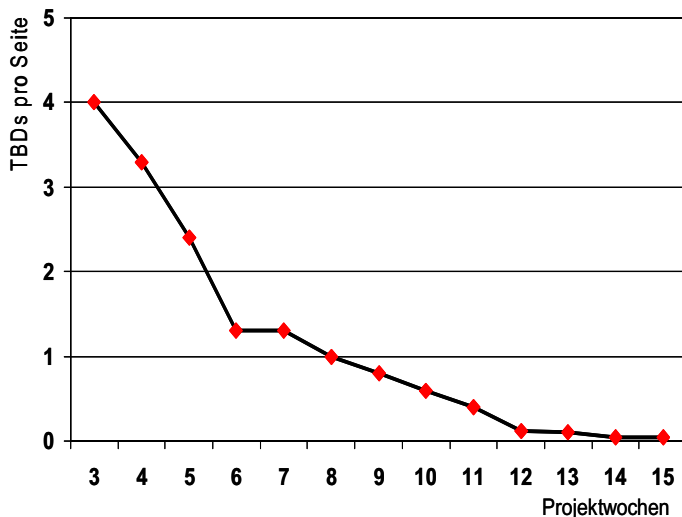


Abbildung 4.1 TBDs über der Projektlaufzeit

In unserem Fall wurden die TBDs pro Seite als Kennzahl verwendet. Es besteht auch die Möglichkeit, die absolute Zahl der TBDs über der Zeit einzutragen. Weil aber gerade in den ersten Projektwochen das Dokument in Sprüngen wachsen wird, ist es besser, wenn man die TBDs pro Seite ermittelt. Ein Dokument wächst besonders dann recht ungleichmäßig, wenn mehrere Autoren daran arbeiten. Dann bringt unter Umständen ein Autor an einem Tag 30 oder 50 Seiten ein, und damit ändern sich Charakter und Qualität des Dokuments signifikant.

In unserem Fall geht die Zahl der weißen Stellen im Dokument gegen Mitte der Phase signifikant zurück. Hier hat also das Management dafür gesorgt, dass das Wissen der Fachleute über die Applikation in die Spezifikation eingeflossen ist.

Interessant ist in dieser Anfangsphase noch eine andere Art der Statistik. Eine solche Metrik ist in Abbildung 4.2 zu sehen.

Eine der wichtigsten Erkenntnisse bei den ersten großen Software-Projekten fasst Brooks mit den Worten zusammen: »Adding manpower to a late project makes it later.« Mit anderen Worten: Es bringt nichts, wenn man relativ spät im Projekt neue Mitarbeiter einstellt. Weil sie mit den Gegebenheiten der Applikation nicht vertraut sind, beanspruchen sie zunächst einmal die Zeit ihrer Kollegen, die ihnen alles erklären sollen. Dadurch sinkt deren Produktivität.

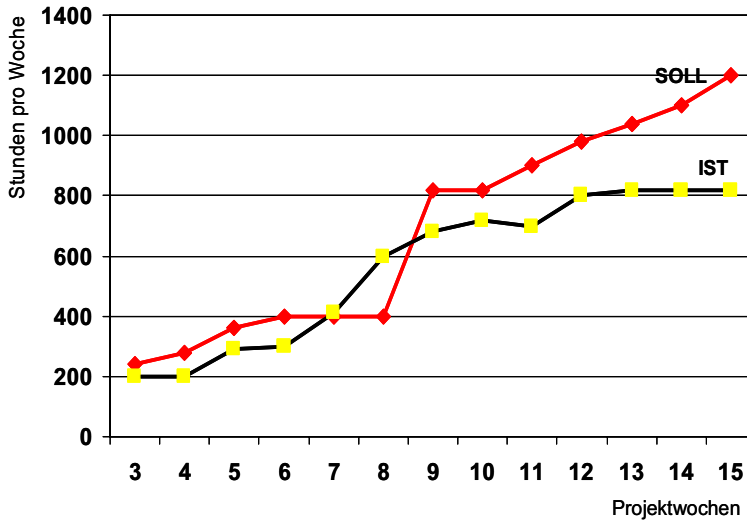


Abbildung 4.2 Personeller Aufwand

Es ist also für einen erfahrenen Projektmanager wichtig, dass er mit der mehr oder minder gleichen Mannschaft das gesamte Projekt durchzieht. Deswegen ist die erste Projektphase auch der Zeitraum, in der diese Mannschaft zusammenkommt.

Unsere Metrik zeigt deutlich, dass der Projektmanager ab Mitte der Phase mit der Rekrutierung neuer Mitarbeiter nicht mehr nachkommt. Er fällt hinter die Planzahlen zurück. In den ersten Wochen geht es darum, die Spezifikation zu erstellen. Aber bereits ab Woche 8 wandelt sich das Bild. Nun sind die Entwickler als Kritiker der Software-Anforderungen zunehmend gefragt. Und gerade hier scheint es Schwierigkeiten zu geben. Es kommen nicht genug Fachleute an Bord.

Diese Metrik gibt also Grund zur Beunruhigung im Management. Es müssen Maßnahmen ergriffen werden, um diesem Projekt mehr personelle Ressourcen zur Verfügung zu stellen. Wenn es nicht gelingt, dieses Defizit in den ersten Wochen der Designphase zu beheben, könnte das Projekt ernsthaft in Schwierigkeiten geraten.

## 4.3 Reviews

Das Wort *Review* ist ein schillernder Begriff, und deswegen müssen wir ihn richtig einordnen. Er taucht in Qualitätsstandards wie DIN EN ISO 9001 häufig auf und hat dadurch Eingang in die deutsche Sprache gefunden.

Grundsätzlich geht es darum, etwas zu überprüfen. Diese Prüfung kann allerdings sehr verschiedene Formen annehmen. Das Review eines Dokuments kann von einem einzelnen Prüfer an seinem Schreibtisch vorgenommen werden, etwa mit einer Checkliste. Oder es kann sich um eine Veranstaltung handeln, zu der drei Dutzend Teilnehmer aus zwei Kontinenten angereist kommen.

Der Grund für die Bedeutung von Prüfungen relativ früh im Entwicklungszyklus, wenn erst wenige Software-Produkte vorliegen, ist in erster Linie finanzieller Natur. Zwar gelingt es durch das Testen der Software vielfach, die meisten Fehler zu finden und sie mit einer geringen Restfehlerrate auszuliefern. Es ist jedoch relativ teuer, Fehler zu beseitigen, wenn sie bis zum Systemtest im Produkt verbleiben.

Die relativen Kosten der Fehlerbeseitigung sind in Tabelle 4.1 aufgelistet.

Phase	Relative Kosten (Faktor)
Ermittlung der Anforderungen	1
Design	3-6
Kodierung	10
Unit Test	15-40
Akzeptanztest	30-70
Einsatz und Betrieb	40-1000

**Tabelle 4.1** Relative Kosten der Fehlerbeseitigung [17]

Wenn man den Faktor 1 mit rund 200 Euro ansetzt, lässt sich leicht errechnen, welche enormen Summen sich einsparen lassen, wenn Fehler noch in der ersten Phase der Entwicklung gefunden und beseitigt werden. Betrachten wir dazu zunächst die möglichen Reviews im Überblick (siehe Abbildung 4.3).

Bei den internen Reviews ist zunächst das Management Review zu nennen. Natürlich hat das eigene Management des Unternehmens jederzeit das Recht, sich über den Status eines Projekts in einem Review informieren zu lassen.

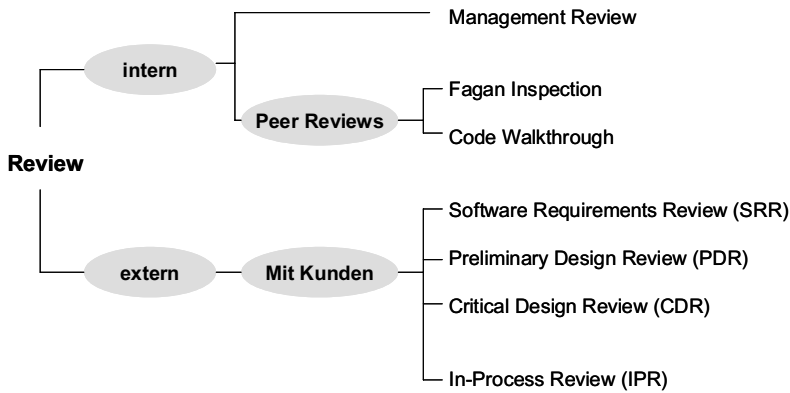


Abbildung 4.3 Mögliche Reviews

Die Peer Reviews tragen ihren Namen deshalb, weil die Reviewer [29] mit dem Entwickler, dessen Produkt einer Prüfung unterzogen wird, auf einer Stufe der Hierarchie stehen. Manager sind in derartigen Veranstaltungen ausdrücklich unerwünscht. Es geht darum, das Produkt gründlich zu untersuchen, und deshalb darf keine Frage ausgespart werden. Für Dokumente, und dazu gehört in erster Linie die Spezifikation, kommt als Methode in erster Linie die Fagan Inspection in Frage.

Bei den Veranstaltungen mit dem Kunden geht es um externe Reviews, weil firmenfremde Personen anwesend sind. Bezogen auf den Zeitplan sehen wir diese Reviews noch einmal in Abbildung 4.4.

Das für die Software-Spezifikation entscheidende Review ist das Software Requirements Review (SRR). Es findet gegen Ende der ersten Phase statt. Wenn dieses Review bestanden wurde, haben sich Auftraggeber und -nehmer auf eine gemeinsame Grundlage für das Projekt geeinigt. Das Dokument dazu ist die Software-Spezifikation.

Für die Designphase sieht man in der Regel zwei Reviews vor, das Preliminary Design Review (PDR) und das Critical Design Review (CDR). Das erste Review bildet den Abschluss der Phase »Grobentwurf«, befasst sich also im Wesentlichen mit der Architektur der Software. Das CDR hat nicht umsonst das Wort *critical* im Namen: Nach diesem Review hat der Kunde keine Chance mehr, das Projekt abzubrechen. Er muss bis zum Ende durchhalten, wenn er ein ausführbares Programm erhalten will.

Weil unter Umständen die Auslieferung der Software nach dem CDR noch Jahre in der Ferne liegt, können so genannte *In-Process Reviews* vereinbart werden. Dadurch bekommt der Kunde Gelegenheit, sich über den Projektfortschritt zu unterrichten.

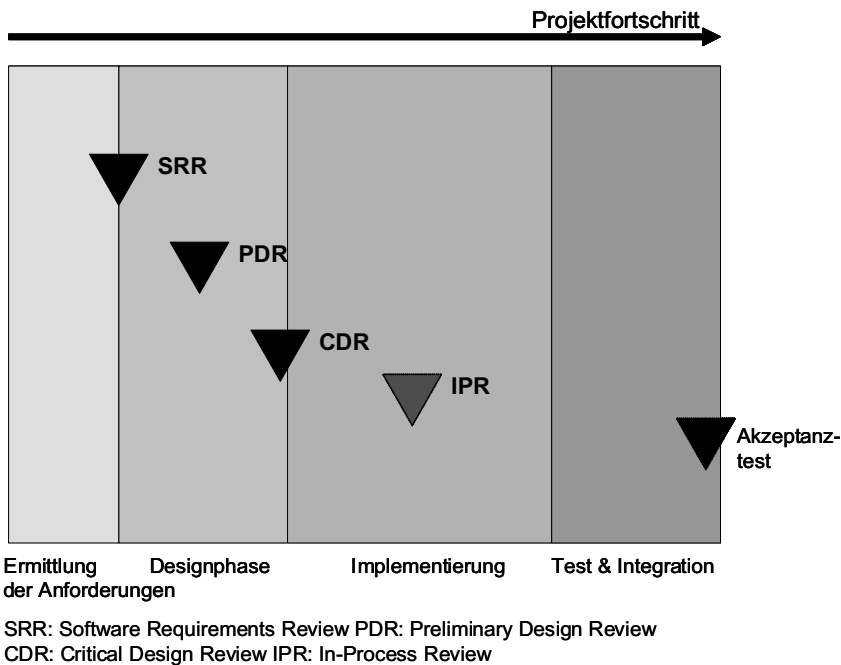


Abbildung 4.4 Reviews mit dem Kunden

Das Ende des Projekts bildet in der Regel ein Akzeptanztest. Er markiert den Abschluss der Entwicklung. Nach diesem Test liegt die Verantwortung für die Software und deren Einsatz beim Kunden.

Auch beim Erstellen von Webseiten halten sich die Beteiligten im Wesentlichen an derartige Modelle. Nicht immer ist allerdings das Projekt so groß, dass ein Review den großen Rahmen rechtfertigen würde. Auch bei solcher Software muss die Spezifikation überprüft werden.

Das kann auch durch einen einzelnen Mitarbeiter, etwa einen Qualitätssicherer, an seinem Schreibtisch geschehen. Das Review des Dokuments sollte sich dabei auf die in Tabelle 4.2 aufgeführten Bereiche beziehen.

Bei dem Review sollten Kommentare entstehen, die durchnummeriert sind und sich auf die Kapitelnummern des Dokuments beziehen. Falls generell Kritik am Dokument zu üben ist, kann der Reviewer einen globalen Kommentar verfassen. Je genauer und auf den Punkt gebracht ein Kommentar ist, desto eher ist der Autor in der Lage, darauf einzugehen und das Dokument zu verbessern. Kommentare der Art: »Manche Begriffe sind im Glossar nicht erklärt«, helfen nicht weiter, weil der Autor der Spezifikation nicht weiß, welche Begriffe der Leser vermisst. Deswegen kann er nichts für diesen Leser tun.

Bereich	Fragestellung
Leser	Ist das Dokument in einer Sprache geschrieben, die Kunde und Anwender verstehen? Ist der Text weder zu leicht noch zu schwierig?
Wortwahl, Terminologie	Passt die Terminologie zum Leserkreis? Werden bestimmte Ausdrücke über das gesamte Dokument hinweg einheitlich benutzt? Werden Ausdrücke so definiert, wie es in der Fachwelt üblich ist? Werden abweichende Definitionen erklärt? Sind Akronyme erklärt worden, und gibt es eine Liste mit Akronymen?
Inhalt	Sind die wichtigsten Themen behandelt worden? Gibt es Themen, die fehlen, aber in das Dokument gehören? Werden alle Inhalte in der notwendigen Tiefe behandelt?
Fakten	Sind die Informationen faktisch und technisch korrekt? Sind bestimmte Daten veraltet? Stimmt das Inhaltsverzeichnis mit den Kapitelüberschriften überein?
Grafiken	Sind die Grafiken korrekt, und stehen sie an der richtigen Stelle im Text? Stehen Bildunterschriften immer bei den zugehörigen Grafiken? Sind alle Texte in Grafiken lesbar?
Rechtschreibung	Entspricht der Text der deutschen Rechtschreibung? Werden Anglizismen über das gesamte Dokument hinweg in einheitlicher Schreibweise gebraucht? Sind ungewöhnliche Begriffe erklärt worden, etwa in einem Glossar?

**Tabelle 4.2** Fragen zur Überprüfung der Spezifikation

Es wird nicht ausbleiben, dass der Kunde oder die Anwender im Laufe des Projekts Änderungswünsche vorbringen. Es stellt sich so die Frage, wie der Auftragnehmer damit umgeht.

## 4.4 Änderungen im Entwicklungsverlauf

*Change is inevitable; growth is optional.*

*Walt Disney*

Es ist bestimmt nicht die Absicht des Managements, und das gilt sowohl für den Kunden als auch für den Auftragnehmer, nach der Annahme der Spezifikation das Dokument zu ändern. Insofern werden beide Seiten Anstrengungen machen, Änderungen zu verhindern oder zu erschweren. Manchmal sind sie jedoch unvermeidlich.

Wenn bei unserer Website der Kunde zum Beispiel das Kreuzfahrerschiff für den Indischen Ozean verkauft und nur noch Fluss-Kreuzfahrten auf dem Nil betreibt, wäre es töricht, diese Seiten zu erstellen. Wenn er andererseits die Kalahari-Wüste neu aufnimmt in sein Programm, müssen wir auch darauf reagieren. Wir

können keine Software ausliefern, die schon bei der Auslieferung nicht mehr aktuell ist. Das gilt für das WWW mehr als für jede andere Art der Software.

Die Frage ist, wie wir mit Änderungen umgehen. Die NASA sieht bei jedem Auftrag für Änderungen ein gewisses Polster an Stunden vor. Ist dieses aufgebraucht, hat der Kunde die Wahl: Entweder kommt das vorgeschlagene Feature nicht in das aktuell bearbeitete Release der Software oder der Auslieferungstermin verschiebt sich.

Um Änderungen an der Software bearbeiten zu können, hat sich die Einrichtung eines Software Change Control Boards (SCCB) bewährt. Seine Arbeit [14,21] ist in Abbildung 4.5 dargestellt.

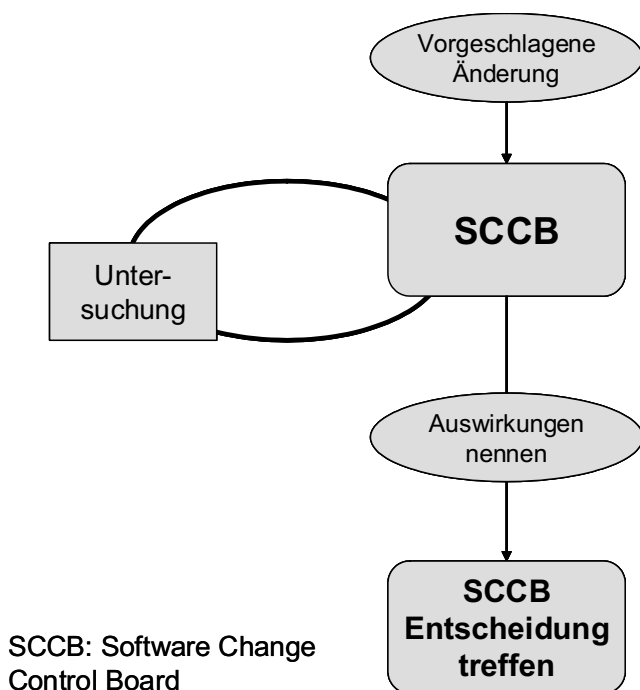


Abbildung 4.5 Software Change Control Board

Das SCCB ist mit Fachleuten besetzt, die aus dem Projekt stammen und in der Regel dem mittleren Management zugerechnet werden. Dazu gehören zum Beispiel der Gruppenleiter der Software-Entwicklung im Projekt, ein Vertreter des Qualitätsmanagements und möglicherweise auch ein Vertreter des Konfigurationsmanagements. Es ist auch denkbar, einen Vertreter des Kunden zu allen Sitzungen einzuladen, wenn es um Änderungen geht, die der Kunde zu vertreten hat.

Das SCCB bewertet jede vorgeschlagene Änderung – nachdem ein Produkt erstmals unter Konfigurationskontrolle gestellt wurde – nach seinen fachlichen, finanziellen und terminlichen Auswirkungen. Es gibt also bei jeder vorgeschlagenen Änderung an, was sie kosten und ob sich der Auslieferungstermin der Software dadurch verzögern würde.

Man ist möglicherweise geneigt, eine Änderung mit einem Aufwand von einem Manntag als geringfügig und verkraftbar zu betrachten. Aber bereits fünf solcher Änderungen verursachen einen Aufwand von einer Woche. Die Erfahrung hat jedenfalls gezeigt, dass das Management nur solche Projekte unter Kontrolle hat, bei denen ein SCCB vorhanden ist. Deswegen kann auf ein solches Instrument zur Steuerung der Änderungen im Projektverlauf nicht verzichtet werden.

## 4.5 Den Prozess beherrschen

*At Group L. Stoffel oversees six first-rate programmers, a managerial challenge roughly comparable to herding cats.*

*The Washington Post Magazine, 9. Juni 1985*

In großen Firmen und Konzernen wird die Firmenleitung nicht in der Lage sein, sich täglich um jedes Projekt zu kümmern. Umso wichtiger ist es, einen Prozess vorzuschreiben, der die Entwicklung beherrschbar macht. Das beginnt mit einer genauen Definition der Schritte, die zur Ermittlung der Software-Anforderungen notwendig sind. Dazu gehört auch eine kritische Prüfung des Dokuments durch Fachleute, die für die Realisierung der Software verantwortlich sind.

Ebenso zählen Vorgaben in der Form von Produktmustern für Dokumente dazu, und nicht zuletzt ist der Kunde immer einzubinden. Wer den Prozess beherrschen will, der kann sich nicht allein auf sein Gefühl, seine Erfahrung oder sein Gespür verlassen. Nur wer Zahlen vorweisen kann, darf sich mit Fug und Recht Manager nennen. Deswegen sind bereits in der ersten Phase der Software-Entwicklung ein paar Metriken unverzichtbar.

Wenn im Management diese Maßnahmen ergriffen werden, hat das Projekt eine gute Chance, zu einem Erfolg zu werden. Gerade in jungen Märkten, und dazu zählt der Bau von Webseiten innerhalb der EDV, tummeln sich Scharlatane und Glücksritter. Hier kann ein Unternehmen glänzen, das nicht nur Versprechungen macht, sondern jedes Projekt mit Elan und Enthusiasmus, aber auch mit einem gesunden Sinn für die Realität angeht.