

Diplomarbeit Nr. 2174

**Empirische Untersuchung von
Vorgehensweisen in der
Softwarewartung**

Dražen Pavlaković

Studiengang:	Softwaretechnik
Prüfer:	Prof. Dr. rer. nat. Jochen Ludewig
Betreuer:	Dipl.-Inf. Stefan Opferkuch
Beginn am:	26.01.2004
Beendet am:	27.07.2004
CR-Klassifikation:	D2.9, K6.3

Fakultät Informatik, Elektrotechnik, und Informationstechnik
Universität Stuttgart
Institut für Softwaretechnologie
Abteilung Software Engineering
Universitätsstraße 38
70569 Stuttgart

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Hintergrund.....	1
1.2	Aufgabenstellung.....	1
1.3	Ziele.....	1
1.4	Untersuchte Systeme.....	2
1.5	Gliederung des Dokuments.....	4
2	Software-Wartung.....	6
2.1	Probleme.....	7
2.2	Definitionen.....	10
2.3	Industrienormen und Standards.....	11
2.3.1	ISO/IEC 12207: 1995 Software life cycle processes.....	12
2.3.2	IEEE 1219-1998 Standard for Software Maintenance.....	15
2.3.3	CMMI.....	19
3	Methodik.....	23
4	Die untersuchten Systeme.....	25
4.1	System A.....	26
4.1.1	Vorstellung des Systems.....	26
4.1.2	Die Rollen.....	26
4.1.3	Der Wartungsprozess.....	28
4.1.4	Besonderheiten.....	29
4.2	System B.....	29
4.2.1	Vorstellung des Systems.....	29
4.2.2	Die Rollen.....	30
4.2.3	Der Wartungsprozess.....	31
4.2.4	Besonderheiten.....	32
4.3	System C.....	33
4.3.1	Vorstellung des Systems.....	33
4.3.2	Die Rollen.....	34
4.3.3	Der Wartungsprozess.....	36
4.3.4	Besonderheiten.....	36
4.4	System D.....	37
4.4.1	Vorstellung des Systems.....	37

4.4.2 Die Rollen.....	38
4.4.3 Der Wartungsprozess.....	38
4.4.4 Besonderheiten.....	40
4.5 System E.....	41
4.5.1 Vorstellung des Systems.....	41
4.5.2 Die Rollen.....	42
4.5.3 Der Wartungsprozess.....	43
4.5.4 Besonderheiten.....	44
5 Auswertung.....	45
5.1 Vergleich der Standards.....	45
5.2 Vergleich der Wartungs-Projekte.....	46
5.3 Vergleich der Praxis mit der Literatur.....	48
6 Zusammenfassung.....	51
7 Fazit und Ausblick.....	52
Anhang A – Fragebogen.....	54
Zeitachse der Softwareentwicklung und -Wartung.....	54
Produkt.....	55
Wartungsmodell.....	57
Projektleitung.....	59
Personal.....	60
Begriffslexikon.....	61
Erklärung.....	62

Abbildungsverzeichnis

Abbildung 1 Alter der Systeme.....	3
Abbildung 2 Anzahl der Benutzer.....	4
Abbildung 3 Anzahl der Kunden.....	4
Abbildung 4 ISO/IEC Maintenance Process Activities.....	12
Abbildung 5 IEEE Maintenance Process.....	16
Abbildung 6 Kommunikationswege System A.....	27
Abbildung 7 Kommunikationswege System B.....	31
Abbildung 8 Kommunikationswege System C.....	35
Abbildung 9 Kommunikationswege System D.....	39
Abbildung 10 Kommunikationswege System E.....	42

Tabellenverzeichnis

Tabelle 1 Umfang der Systeme.....	2
Tabelle 2 Übersicht der Aufwände.....	3
Tabelle 3 Übersicht der Systeme.....	25
Tabelle 4 Vergleich der Standards.....	45
Tabelle 5 Vergleich aller Projekte.....	46

1 Einleitung

1.1 Hintergrund

Bei der Wartung langjährig eingesetzter Software stehen Unternehmen häufig vor großen Problemen. Eine Dokumentation ist oftmals nicht vorhanden, die ursprünglichen Entwickler der Software sind nicht mehr verfügbar und eine weitere Wartung der Software ist nicht oder nur unter erheblichen Schwierigkeiten möglich. Dieses Phänomen tritt bei allen Arten von heute eingesetzter Software auf.

1.2 Aufgabenstellung

In dieser Diplomarbeit soll das Vorgehen verschiedener Unternehmen bei der Wartung ihrer Software untersucht werden.

Die Ergebnisse der Untersuchung sollen mit bekannten Modellen zur Einstufung der Prozessreife verglichen werden. Für diesen Vergleich sollen das Capability Maturity Model Integration (CMMI) des Software Engineering Institute (SEI) der Carnegie-Mellon Universität [Kneuper 2003], der ISO/IEC Standard 12207 (1995) der International Organisation for Standardization (ISO) und der International Electrotechnical Commission (IEC) [ISO/IEC 1995], [Pigoski 1996] und der IEEE 1219-1998 des Institute of Electrical and Electronics Engineers (IEEE) [IEEE 1998] herangezogen werden. Dabei soll festgestellt werden, welche Übereinstimmungen und Abweichungen es zwischen der Beschreibung der Wartung in der Literatur und der Situation bei den Industriepartnern gibt.

Des Weiteren sollen Gemeinsamkeiten und Unterschiede bei der Durchführung der Wartung zwischen den Industriepartnern festgestellt, herausgearbeitet und präsentiert werden.

Die Daten für die Untersuchung bei den Industriepartnern werden anhand von Interviews erhoben. Um die Interviews vergleichbar und planbar zu gestalten wird ein Fragebogen erstellt.

1.3 Ziele

In dieser empirischen Untersuchung werden Daten aus fünf verschiedenen Projekten erhoben. Die relevanten Aspekte werden mit Hilfe der Literatur ermittelt. Die aus der Untersuchung resultierenden Daten sollen die Situation in den Projekten widerspiegeln. Es wird versucht auch Informationen über die Erstentwicklung zu erhalten um Veränderungen im Laufe der Zeit festzustellen.

Ziel dieser Arbeit ist es die Vorgehensweisen der Industriepartner untereinander zu vergleichen und falls möglich einen Ist-Zustand in der Industrie zu erhalten. Neben der Industrie wird auch ein Vergleich zwischen den schon genannten Vorgehensmodellen erstellt, um einen Soll-Zustand in der Literatur zu erhalten. Nach Erfassung dieser zwei Zustände werden diese auf Unterschiede und Gemeinsamkeiten untersucht.

Die Datenbasis dieser empirischen Untersuchung ist zu klein, als dass allgemeingültige Aussagen gemacht werden könnten. Da alle Industriepartner im gleichen Dienstleistungsbereich, nämlich dem Finanzsektor, tätig sind, könnten eingeschränkt Aussagen zur Softwarewartung in diesem Wirtschaftsbereich gemacht werden.

Ziel dieser Diplomarbeit ist es den Ist-Zustand in der Industrie zu erfassen und zu beschreiben. Es sollen keine Vorschläge zur Prozessverbesserung in den untersuchten Projekten erstellt werden.

1.4 Untersuchte Systeme

Bei den untersuchten Systemen handelt es sich um zwei Produkte für den Bankensektor, ein prozessorientiertes Vertriebssystem, eine Kontenverwaltung und ein System für die Hauptbuchhaltung. Bei den Systemen mit der kürzesten Zeitspanne sind fast schon zehn Jahre seit Entwicklungsbeginn vergangen, bei den ältesten sind es schon 30 oder mehr Jahre seit Entwicklungsbeginn. In *Abbildung 1* ist eine Übersicht des Alters der Systeme zum Zeitpunkt der Untersuchung dargestellt.

Neben dem Alter der betrachteten Systeme gibt es auch eine große Spanne in der Größe der Implementierung. Das älteste System ist nicht das Größte und das kleinste System ist auch nicht das jüngste. *Tabelle 1* vergleicht den Umfang der System zum Zeitpunkt der Erstentwicklung und zum Zeitpunkt der Untersuchung. Alle Angaben sind in kLOC (Tausend Lines of code). Einige Interviewpartner waren nicht in der Lage diese Angaben zu machen, da sie nie erfasst wurden.

Größe in kLOC	System A	System B	System C	System D	System E
Erstentwicklung	-	500	190	-	5
Heute	-	25000	250	-	10

Tabelle 1 Umfang der Systeme

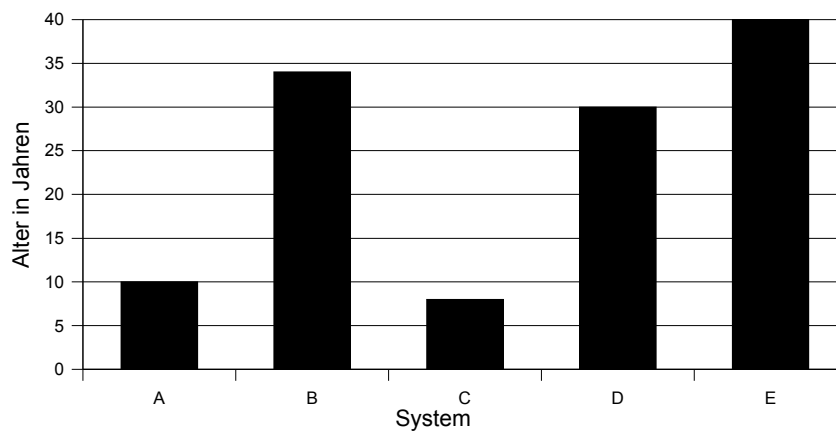


Abbildung 1 Alter der Systeme

Von besonderem Interesse war bei dieser Untersuchung der Vergleich der geleisteten Aufwände für die Entwicklung und die darauf folgende Wartung. In *Tabelle 2* sind die Systeme mit ihren Aufwänden für die Erstentwicklung und die Aufwände für die Wartung dargestellt. Alle Angaben aus den Interviews wurden in Tausend Personentage (kPT) umgerechnet.

Aufwand in kPT	System A	System B	System C	System D	System E
Erstentwicklung	80	-	6	-	0,2
Wartung	80	1380	3	1	40

Tabelle 2 Übersicht der Aufwände

Zuletzt folgt eine Übersicht zur Anzahl der Benutzer und Kunden der einzelnen Systeme. In *Abbildung 2* sind die Anzahl der Benutzer und in *Abbildung 3* die Anzahl der Kunden aufgeführt. Auffällig dabei ist die ungleiche Verteilung zwischen Kunden- und Benutzerzahl. System A wird für zehn externe Kunden mit ungefähr 3000 Benutzern entwickelt und gewartet. Die Systeme B, C, D und E werden entweder für das eigene Unternehmen oder den einzigen externen Kunde gewartet. Bei System B gibt es zusätzlich noch die Besonderheit, dass die Wartung vom Produkteigentümer an ein eigenständiges Tochterunternehmen übergeben wurde. Für das Wartungsunternehmen gibt es somit nur einen Kunden, dieser bietet sein System aber vielen weiteren externen Kunden mit der entsprechenden Anzahl von Benutzern an.

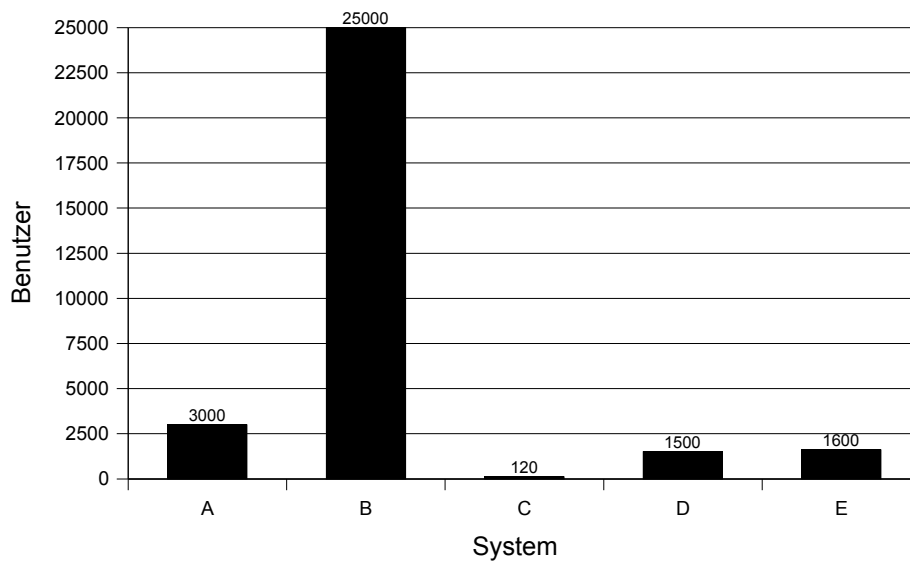


Abbildung 2 Anzahl der Benutzer

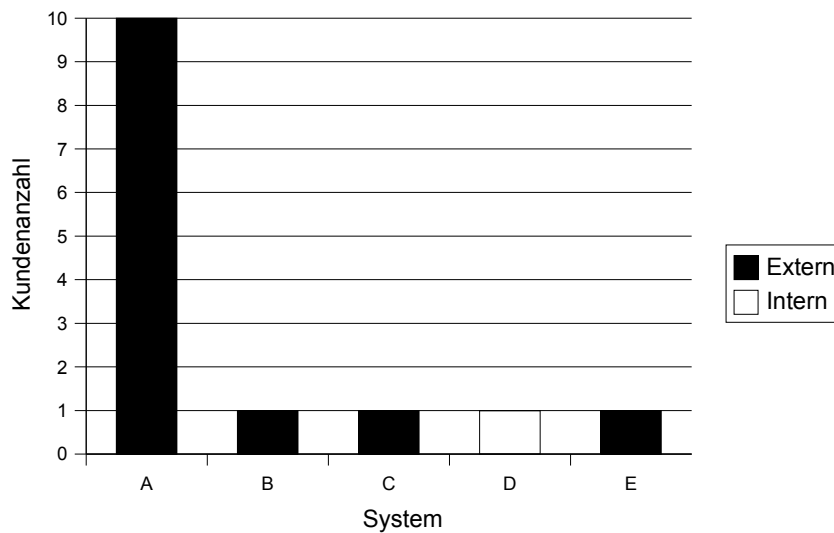


Abbildung 3 Anzahl der Kunden

Weitere Eigenschaften werden in den Vorstellungen der einzelnen Systeme im *Kapitel 4* beschrieben. Zusätzliche Angaben zu den befragten Unternehmen und ihrer Produkte gibt es nicht und sie werden in dieser Arbeit ansonsten anonym behandelt.

1.5 Gliederung des Dokuments

Der Aufbau des Dokuments orientiert sich an der Vorgehensweise bei der Bearbeitung der Diplomarbeit. Im zweiten Kapitel wird die Problematik der Software-Wartung und mögliche Lösungsansätze der Literatur vorgestellt. Im Rahmen dieses Kapitels werden drei Vorgehens-

modelle beschrieben. Im folgenden Kapitel wird auf die Datenerhebung bei den Industriepartnern eingegangen. Fragenkatalog und die Durchführung der Interviews werden beschrieben. Kapitel vier dient der Präsentation der erhobenen Daten in den betrachteten Systemen. Danach folgt der Vergleich der Literatur mit der Praxis. Mit dem Fazit und dem Ausblick wird die Arbeit abgeschlossen. Im Anhang befindet sich der Fragebogen für die Interviews.

2 Software-Wartung

Betrachtet man die einschlägige Fachliteratur so entsteht der Eindruck, dass Wartung kein größeres Problem im Software-Lebenszyklus darstellt. Dies lässt sich exemplarisch an den Umfängen der Wartung in Standardwerken des Software Engineering zeigen. In Sommervilles Software Engineering [Sommerville 1996] wird auf weniger als 2% des gesamten Buches auf die Problematik der Wartung eingegangen. Auch in Pressmans Software Engineering – A practitioner's approach [Pressman 2001] stellt dieser zwar fest, dass Wartung einen sehr hohen Anteil an den Gesamtkosten für die Entwicklung eines Softwaresystems belegt, aber er erwähnt Wartung aber nur als einen Aspekt des Reengineering.

Die Anzahl von Büchern welche sich mit Wartung beschäftigen spiegelt die seit Jahren vermittelte große Bedeutung von Wartung im Bereich der Softwareentwicklung nicht wieder.

Bei der Suche nach Vorgehensmodellen für die Wartung sieht das Bild ebenso unbefriedigend aus. Es gibt sehr viele Vorgehensmodelle zur Softwareentwicklung: CMM, PSP Bootstrap, GQM, V-Modell, SPICE, u.a.. Die Suche nach alternativen Vorgehensmodellen für die Wartung wurde mit ernüchterndem Ergebnis abgeschlossen. So wie sich die Literatur kaum über die Wartung äußert gibt es noch weniger Vorschläge wie Wartungsprozesse abzulaufen haben. Als probates Mittel bleibt nur der Griff zu den Normen und Standards der Softwareentwicklung IEEE 1219 und ISO/IEC 12207.

Um das Ausmaß der Problematik zu verdeutlichen bieten sich zwei Metaphern an. Eine von [Canning 1972] und eine Abwandlung dieser von [Martin, McClure 1983]. Canning beschreibt die Problematik der Software-Wartung ähnlich einem Eisberg. Die Gefahren gehen dabei nicht vom sichtbaren Teil des Eisbergs respektive der Wartung aus, sondern vom bedeutend größeren nicht sofort ersichtlichen Teil. Martin und McClure wandelten die Metapher leicht ab und deklarierten den sichtbaren Teil des Eisbergs als Entwicklung und der unsichtbaren als Wartung. Diese Metapher soll das Verhältnis der Aufwände für Entwicklung und Wartung eines Systems beschreiben.

Passender beschreibt [Sneed 1991] die Situation der Wartung im Vergleich zur Erstentwicklung: *Es ist zweierlei, eine neue Strasse zu bauen oder eine bestehende Strasse zu reparieren oder zu erweitern. Im zweiten Fall kommt es vor allem darauf an, so viel wie möglich von der*

alten Bausubstanz zu bewahren und so wenig wie möglich durch die Wartungsmaßnahmen zu zerstören, bei gleichzeitiger Aufrechterhaltung des Betriebes.

Der letzte Nebensatz zeigt besonders deutlich worauf es bei der Software-Wartung ankommt: Die Benutzer verlangen nach Anpassungen und Erweiterungen ohne in ihrer täglichen Arbeit eingeschränkt zu werden.

2.1 Probleme

Was ist Wartung?

Unter Wartung im allgemeinen Sprachgebrauch wird die Wiederherstellung eines Urzustandes verstanden. Wartung im Sinne der Softwareentwicklung stellt nicht den Ursprungszustand her, sondern wandelt das Produkt in einen neuen vom Benutzer gewünschten Zustand. Anhand von IEEE1219 und ISO/IEC 12207 wird der Begriff der Wartung wie folgt definiert:

- Nach IEEE 1219 ist *Wartung die Veränderung eines Softwareprodukts nach der Auslieferung, um Fehler zu beheben, Performance oder andere Attribute zu verbessern oder Anpassungen an die veränderte Umgebung vorzunehmen.*

- ISO/IEC 12207 definiert Wartung als: *... Software unterliegt Veränderungen bei Code und anhängender Dokumentation aufgrund von Problemen oder der Notwendigkeit von Verbesserungen. Ziel bleibt die Änderung der bestehenden Software und dabei die Unversehrtheit zu erhalten.*

Des Weiteren definieren [Martin, McClure 1983] Wartung als:

- *... Änderungen, die an Computerprogrammen durchgeführt werden müssen nach deren Auslieferung an den Kunden*

[Mayrhauser 1990] definiert Wartung wie folgt:

- *Wartung deckt den Lebenszyklus eines Softwaresystems von der Installation bis zum Abschalten dieses.*

Welche Arten von Wartung gibt es?

Nach dem Wartungsbegriff müssen jetzt die einzelnen Kategorien von Wartung definiert werden. [Swanson 1980] unterteilt Wartung in vier Kategorien: Korrektive oder adaptive Wartung, Erweiterungen oder Optimierungen.

- Korrektive Wartung
Änderungen die notwendig sind um konkrete Fehler zu beheben.
- Adaptive Wartung
Jegliche Bemühungen um ein Softwaresystem an veränderte Umgebungsbedingungen anzupassen und dadurch weiter funktionsfähig zu halten.
- Erweiterung
Alle Änderungen, Löschungen, Einfügungen, Veränderungen, Erweiterungen und Verbesserungen am System um es an die gewachsenen oder erweiterten Bedürfnisse der Benutzer anzugleichen.
- Optimierung
Alle Änderungen um das System performanter zu machen oder Restrukturierungsmaßnahmen um die Wartung zukünftig zu erleichtern.

ISO/IEC 12207 unterscheidet:

- Korrektive oder präventive Wartung, Verbesserung oder Anpassung

IEEE 1219 unterscheidet bei der Wartung:

- Korrektive, adaptive oder perfektive Wartung und den Notfall

[Ludewig, Opferkuch 2004] unterteilen Wartung in vier Kriterien: Anlass der Wartung, der Ausgangszustand der Wartung, Art der Anforderungen und die Prozesse nach denen Wartung abläuft. Bemerkenswert ist dabei die Definition zum Anlass der Wartung:

Zu Beginn der Wartung einer Software, wenn vor allem Korrekturen erforderlich sind und wenn erst nach Auftreten von Fehlern und Problemen agiert wird, wird dies als *reaktive* Wartung bezeichnet. Im Gegensatz dazu ist die *proaktive* Wartung, Anpassungen und Erweiterungen werden in der Regel proaktiv bearbeitet.

Als wohl berühmtestes Beispiel für proaktive Wartung kann das Jahr-2000-Problem mit seinen Aktivitäten zur Behebung betrachtet werden.

Weiter erklären sie: *In der Literatur wird an Stelle von „proaktiv“ das Wort „präventiv“ verwendet. Wir folgen dem nicht, weil die Prävention durch die Medizin eine ganz andere Konnotation hat. Prävention soll Defekte (Krankheit) verhindern, während sich proaktive Wartung mit Defekten befasst, bevor sie Schaden angerichtet haben.*

Ein Beispiel für Literatur, die präventiv verwendet, wäre der auch für diese Untersuchung verwendete ISO/IEC 12207.

Wer ist zuständig für Wartung?

Eine Frage die nicht eindeutig geklärt ist. Soll der Entwickler sein System nach der Auslieferung selbst warten oder soll das System nach der Auslieferung an eine Wartungsorganisation übergeben werden?

Für die Wartung durch den Entwickler sprechen folgenden Vorteile:

- Der Entwickler kennt das System sehr gut
- Es wird keine umfangreiche Dokumentation benötigt
- Es muss keine Schnittstelle zwischen Entwicklern und Wartungsingenieuren definiert sein
- Die Benutzer müssen nur mit einer Abteilung bzw. Organisation zusammenarbeiten
- Die Entwickler sind zufriedener, da ihre Tätigkeit abwechslungsreicher ist

Die ersten drei Punkte bergen die Gefahr, dass es nie eine adäquate Dokumentation zu einem Softwaresystem gibt. In der heutigen Zeit von Ausgliederung und Subunternehmern verliert der Produkteigentümer sehr schnell Wissen über das eigene System, wenn die Dokumentation nur in den Köpfen der Entwickler steckt und nicht schriftlich niedergelegt ist.

Wenn die Entwickler nicht die Wartung durchführen, ergeben sich folgende Vorteile:

- Es gibt eine Dokumentation
- Die Qualität der Dokumentation ist meist besser, da die Wartungsingenieure diese zur Einarbeitung in das für sie neue System benötigen
- Es werden Prozesse zur Implementierung von Änderungen definiert

Für den Fall, dass die Entwickler die Wartung durchführen gibt es auch Nachteile:

- Falls der Anteil an Wartung größer als die Mitwirkung in Neuentwicklungen ist kann dies dazu führen, dass Mitarbeiter die Stelle oder sogar das Unternehmen wechseln
- Jungentwickler wären von dem Umfang an Wartung abgeschreckt
- Während der Durchführung von Wartung neigen die Entwickler zu unwirtschaftlichen Optimierungen des Systems
- Entwickler werden in der Praxis aus Wartungsprojekten abgezogen um sehr wichtige neue Projekte zu entwickeln

Wenn die Entwickler nicht die Wartung durchführen ergeben sich folgende Nachteile:

- Es muss ein bedeutender Aufwand in Schulung der Wartungsingenieure investiert werden
- Wenn Benutzer für die Wartung nicht bezahlen müssen, können diese versuchen nachträglich nicht genehmigte Wünsche aus der Erstentwicklung zu realisieren
- Wenn die Entwickler mehrere Jahre für die Entwicklung benötigt haben, kann niemand erwarten dass die Wartungsingenieure nach wenigen Wochen diesen Wissenstransfer durchgeführt und bewältigt haben, dies führt dazu, dass die Qualität der Wartung zu Beginn niedriger als erwartet ausfallen kann.

Was ist erfolgreiche Wartung?

Harry Sneed hat in [ICSM 1996] folgende Kriterien für den Erfolg von Wartung benannt:

- Funktionalität: Wartungstätigkeiten sollten wenigstens die Funktionalität des gewarteten Systems erhalten, wenn nicht sogar steigern.
- Qualität: Sie soll durch Wartung im System erhalten oder sogar gesteigert werden.
- Komplexität: Durch Wartung soll diese im Verhältnis zur Größe des Systems nicht ansteigen.
- Unbeständigkeit: Die Wartung soll die Unbeständigkeit des Systems nicht erhöhen.
- Kosten: Die Kosten pro Wartungsauftrag sollen nicht ansteigen
- Releasezeitpunkte: Verzögerungen bei der Auslieferung von neuen Versionen sollen vermieden werden.
- Benutzerzufriedenheit: Sie soll mindestens auf gleicher Höhe verbleiben.
- Rentabilität: Die Wartungsaktivitäten sollen rentabel oder zumindest kostendeckend durchgeführt werden.

2.2 Definitionen

In dieser Arbeit werden die folgende Definitionen verwendet.

Softwarewartung

Wartung ist die Veränderung eines Softwaresystems nach der Auslieferung, um Fehler zu beheben, Performance oder andere Attribute zu verbessern oder Anpassungen an die veränderte Umgebung vorzunehmen [Pressman 2001], [IEEE 1998].

Korrektive Wartung

Die korrektive Wartung beschäftigt sich mit der Behebung von Fehlern, die nach der Auslieferung der Software auftreten [IEEE 1998].

Adaptive Wartung

Die Lebensdauer von Software überschreitet oft eine Dauer von 10 Jahren. Es ist nicht außergewöhnlich, dass sich in dieser Zeit die ursprüngliche Systemumgebung verändert. Das hat zur Folge, dass die Software diesen Veränderungen angepasst und weiterentwickelt werden muss. Alle Tätigkeiten, die in diesen Bereichen fallen werden als adaptive Wartung bezeichnet [Pressman 2001], [IEEE 1998].

Perfektive Wartung

Perfektive Wartung bedeutet die Verbesserung der Software. Dies betrifft die Performance, Wartbarkeit oder andere Eigenschaften der Software [IEEE 1998].

Change Control Board

Aufgabe dieses Gremiums ist die Beurteilung von Änderungsanforderungen auf Auswirkungen auf das laufende System. Es beurteilt ob Änderungen wirtschaftlich sinnvoll und durchgeführt oder abgelehnt werden [Sommerville 1996].

2.3 Industrienormen und Standards

So wie die Wartung, im Vergleich zur Entwicklung, in der Literatur unterrepräsentiert ist, wird ihre Bedeutung auch in den Normen und Standards nicht deutlich.

Für den Vergleich zwischen Literatur und Praxis wurden CMMI, IEEE 1219-1998 Standard for Software Maintenance und ISO/IEC 12207: 1995 Standard for Information Technology als Prozessmodelle für die Wartung ausgewählt.

Capability Maturity Model Integration (CMMI)

Das Capability Maturity Model Integration (CMMI) des Software Engineering Institute (SEI) der Carnegie-Mellon Universität wurde im Jahre 2002 in der Version 1.1 veröffentlicht. Es ist das Nachfolgemodell des 1991 veröffentlichten Capability Maturity Model (CMM) ebenfalls vom SEI.

Das SEI wurde vom amerikanischen Verteidigungsministerium, Department of Defense (DoD,) beauftragt Lösungsansätze für die Entwicklung von komplexen Softwaresystemen zu suchen. Als Ergebnis dieser Arbeit entstand das schon erwähnte CMM.

IEEE

Das Software Engineering Standards Committee der Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS) veröffentlichte 1992 einen Standard über Software-Wartung. In dieser Arbeit wird die aktuellste Version von 1998 verwendet.

ISO/IEC

Das Joint Technical Committee 1 (JTC1) des International Organization for Standardization (ISO) und das International Electrotechnical Commission (IEC) förderten die Erstellung des ISO/IEC 12207 und veröffentlichten 1995 diesen Standard für den Software-Lebenszyklus. Ein Teil dieses Standards beschäftigt sich mit Wartung als Bestandteil des Software-Lebenszyklus und ist dadurch von Bedeutung für diese Diplomarbeit.

Bevor nun die einzelnen Vorgehensmodelle vorgestellt werden, soll eine Gemeinsamkeit noch hervorgehoben werden: Alle Modelle verlangen in jeder Phase die Erfassung von Metriken, deshalb wird dieser Aspekt hier kurz genannt.

2.3.1 ISO/IEC 12207: 1995 Software life cycle processes

Software-Wartung ist nach ISO/IEC 12207 einer von fünf primären Lebenszyklusprozessen. Daneben gibt es noch zwei weitere Gruppen von primären Prozessen, nämlich unterstützende und organisatorische.

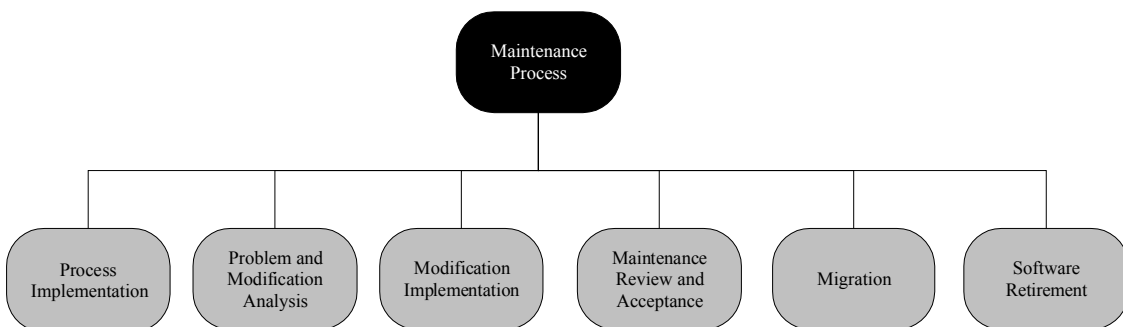


Abbildung 4 ISO/IEC Maintenance Process Activities

Der Wartungsprozess besteht nach ISO/IEC 12207 aus folgenden sechs Aktivitäten:

- Implementierung eines Wartungsprozesses
- Analyse von Problem und Änderung
- Implementierung der Änderung
- Review und Abnahme der Wartung

- Migration
- Außerbetriebnahme der Software

Angestoßen wird der Wartungsprozess durch einen Modification Request (MR). Dieser wird vom Entwickler in einen Problem Report (korrektive Wartung) oder ein Enhancement Request (adaptive oder perfektive Wartung) umgewandelt.

Die Aktivitäten des Wartungsprozess decken dabei im Einzelnen folgende Aufgaben ab:

Implementierung eines Wartungsprozesses

Die Norm schreibt vor, dass Pläne und Vorgehensweisen zur Steuerung von Aufgaben und Aktivitäten im Wartungsprozess entwickelt, dokumentiert und ausgeführt werden. Es müssen Mechanismen zum Empfang und Verfolgung von Problemmeldungen und Änderungsanforderungen erstellt werden. Daneben soll der Urheber über den Verlauf seiner Meldung informiert werden. Jedes gemeldete Problem muss aufgezeichnet und bearbeitet werden. Die Wartungsingenieure sollen den Konfigurationsmanagementprozess für Änderungen am bestehenden System implementieren oder eine Verbindung zum vorhandenen Konfigurationsmanagement schaffen.

Analyse von Problem und Änderung

Eine ankommende Problembeschreibung oder ein Änderungswunsch muss auf seine Auswirkungen für das bestehende System geprüft werden. Es sind dabei folgende Kriterien zu beachten:

Art, Umfang und Risiko der Wartung sowie alternative Lösungsansätze

- Art der Wartung
Korrektive oder präventive Wartung, Verbesserung oder Anpassung an eine neue Umgebung
- Umfang der Wartung
Größe der Änderungen, erwartete Kosten und zeitlicher Rahmen für die Realisierung
- Risiko der Wartung
Auswirkungen auf die Sicherheit oder Leistung des Systems

- Alternative Lösungsansätze

Basierend auf den Analysen müssen Optionen zur Realisierung der Veränderung erstellt werden

Das eingegangene Problem soll von der Wartung bestätigt oder als unbegründet abgewiesen werden. Der Wartungsingenieur muss alle Ergebnisse dokumentieren, und dann die Erlaubnis zur Implementierung der gewählten Lösung einholen.

Implementierung der Änderung

Vor der Realisierung der Änderungen müssen alle zu ändernden Teile und Versionen festgelegt und dokumentiert werden. Die Wartungsanforderung soll in den Entwicklungsprozess geleitet und implementiert werden. Es wird gefordert den Entwicklungsprozess, wie er in der selben Norm beschrieben wird, dabei zu verwenden. Der Entwicklungsprozess soll zusätzlich an die besonderen Anforderungen der Wartung angepasst werden, indem er mit folgenden Daten erweitert wird:

- Test- und Evaluierungskriterien für die geänderten und unveränderten Teile des Systems sollen definiert und dokumentiert werden.
- Vollständige und korrekte Implementierung von neuen und geänderten Anforderungen muss gewährleistet werden. Ebenfalls soll sichergestellt werden, dass die unveränderten Teile nicht betroffen sind. Die Testergebnisse müssen dokumentiert werden.

Review und Abnahme der Wartung

Es wird die Durchführung von Reviews mit den Verantwortlichen, die die Modifikationen genehmigen, verlangt. Als Ergebnis soll dabei die Unversehrtheit des Gesamtsystems festgestellt werden. Den Wartungsingenieuren wird empfohlen nach Bestätigung für die zufriedenstellende Erledigung der geforderten Änderungen zu verlangen.

Migration

Falls ein System aus einer alten in eine neue Umgebung migriert wird, muss ein Migrationsplan entworfen werden, der folgendes enthält:

- Analyse der Anforderungen und Definition der Migration
- Entwicklung von Migrationswerkzeugen
- Umwandlung der Daten und des Systems
- Ausführung der Migration
- Verifikation der Migration

- Unterstützung für das alte System in der Zukunft

Die Benutzer des Systems sollen über den Vorgang der Migration und die Zeitplänen informiert werden. Die Benachrichtigung muss enthalten warum das alte System nicht weiter unterstützt wird, wann das neue System verfügbar wird und wo es alternative Quellen der Unterstützung, für das zukünftig nicht mehr unterstützte System gibt. Zum Zeitpunkt der Migration ist es notwendig das alte System und alle zugehörige Dokumentation zu archivieren

Außerbetriebnahme der Software

Die Software wird auf Verlangen des Eigentümers außer Dienst gestellt.

Um eine Software geregelt aus dem Betrieb zu nehmen, bedarf es eines koordinierten Ablaufs.

Es sind dabei folgende Schritte zu beachten:

- Beendigung der Benutzerbetreuung nach einer gewissen Zeit
- Archivierung des Systems und aller zugehöriger Dokumentation
- Verantwortung für zukünftig verbleibende Supportanfragen festlegen
- Falls möglich, Übergang zu einem neuen Produkt
- Verfügbarkeit der Datenarchive gewährleisten

Die Benutzer müssen über die Abschaltung informiert werden und dabei folgende Informationen erhalten:

- Beschreibung des Ersatzsystems und Zeitpunkt der Verfügbarkeit
- Erklärung warum das bestehende System nicht weiter unterstützt wird
- Beschreibung von alternativen Quellen der Unterstützung, sobald diese eingestellt wird

Schulungsmaßnahmen werden begleitend zur Abschaltung des alten Systems und zur Einführung des Neuen für die Benutzer durchgeführt.

2.3.2 IEEE 1219-1998 Standard for Software Maintenance

Wartung wird nach IEEE 1219-1998 durch einen Modification Request (MR) angestoßen und dabei in folgende Phasen unterteilt:

- Problem/nderungswunsch identifizieren
- Analyse
- Entwurf
- Implementierung
- Systemtest

- Abnahmetest
- Auslieferung

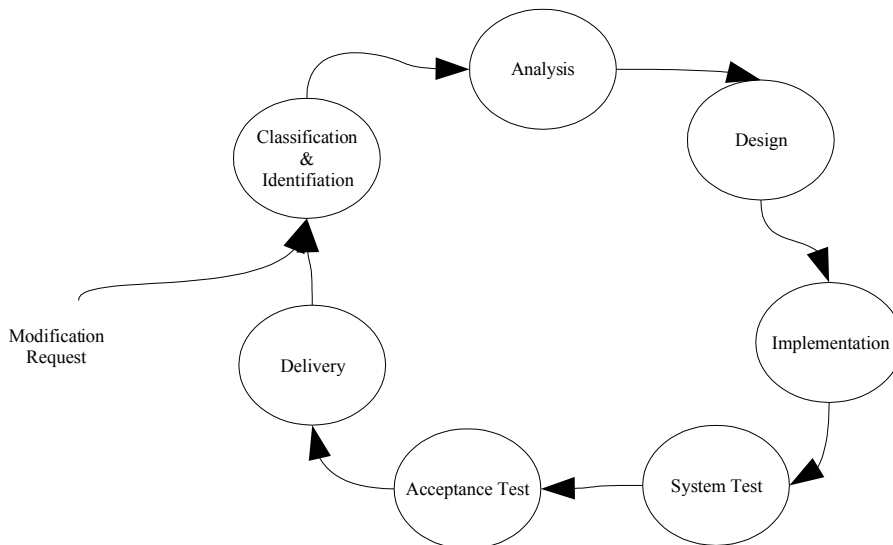


Abbildung 5 IEEE Maintenance Process

Problem/Änderungswunsch identifizieren

In dieser Phase müssen ankommende Probleme oder Änderungswünsche identifiziert, klassifiziert und die Priorität der Meldung festgelegt werden. Für die Klassifizierung stehen folgende Kategorien zur Verfügung:

- Korrektive Wartung
- Adaptive Wartung
- Perfektive Wartung
- Notfall

Neben dieser Einteilung müssen für jeden Änderungswunsch vom Urheber, einem erfahrenem Benutzer, den Entwicklern oder der Projektleitung Aussagen zur Wichtigkeit dieser Änderung gemacht werden. Nach der Einschätzung der Wichtigkeit muss die Änderung analysiert und dann akzeptiert, abgelehnt oder zur weiteren Untersuchung weitergeleitet werden. Falls die Änderung akzeptiert wird müssen Schätzungen zu folgenden Punkte abgegeben werden:

- Grobe Ressourcenplanung
- Schwierigkeitsgrad und Dauer der Implementierung, mit aktuell verfügbaren Mitteln
- Erwartete Auswirkungen für jetzige und zukünftige Nutzer
- Vorteile und Nachteile der Implementierung

Genauere Kosten und Auswirkungen sollten in der darauf folgenden Analyse festgestellt werden. Da die Schätzung ein iterativer Prozess ist, die Priorisierung aber vom Ergebnis dieses Prozesses abhängt, ist es möglich bzw. sogar notwendig vormals getroffene Priorisierungen bei genaueren Abschätzungen ggf. auch zu ändern.

Analyse

Ein einziger Änderungswunsch kann systemweit zu vielen unterschiedlichen Änderungen führen. Damit diese erfasst werden, müssen während der Analyse alle betroffenen Systemteile bestimmt und alle zusätzlichen Anforderungen generiert werden. Deshalb ist es notwendig während der Analyse alle betroffenen Personen/Gruppen (Urheber, Entwickler und Nutzer) zu beteiligen, um als Kompromiss eindeutige Anforderungen zu erhalten.

Auswirkungen sollen mit Hilfe von Dokumentation, welche aus dem Quellcode abstrahiert wurde, untersucht werden.

Analysen der Auswirkungen von Softwareänderung sollten:

- Potentielle Seiteneffekte aufzeigen
- Den Verlauf von erfolgreichen und erfolglosen Änderungen der Vergangenheit berücksichtigen

Außerdem sollte in einer detaillierten Analyse folgendes enthalten sein:

- Feststellung, ob weitere Problemanalysen notwendig sind
- Alternative Lösungsansätze
- Aufzeichnung, ob die vorgeschlagene Änderung akzeptiert oder verworfen wurde
- Schätzung, ob neue Beschränkungen durch die Implementierung der Änderung entstehen können, die während des folgenden Entwurfs zusätzlich beachten werden müssen.
- Dokumentation über jegliche Risiken für Projekt oder Software, welche bei der Analyse entstanden und in späteren Phasen des Software-Änderungszyklus beachtet werden müssen.

Entwurf

Die eigentliche Implementierung der vorgeschlagenen Änderungen beginnt schon während der Entwurfsphase. Es muss jetzt schon auf die Durchführbarkeit der Änderungen geachtet werden. Es ist beispielsweise möglich, dass die Entwickler bei der Implementierung die Aus-

wirkungen auf andere Teile des Systems selbst nicht mehr überblicken können, oder der Entwurf einer gewünschten Änderung zu komplex wird, um diesen zu implementieren.

Für den Entwurf sollen, neben dem bestehenden System, alle Ergebnisse der vorhergehenden Phasen verwendet werden. Als Folge des Entwurfs sollten alle bisherigen Planungen und Schätzungen der ersten beiden Phasen überarbeitet bzw. aktualisiert werden. Darunter fallen zum Beispiel eine überarbeitete Änderungsliste, aktualisierte Testpläne oder eine Liste von Beschränkungen und Risiken.

Implementierung

Neben den Ergebnissen der Entwurfsphase werden folgende Dokumente oder Informationen benötigt:

- Der aktuelle Quellcode mit Kommentaren
- Projekt- und Systemdokumentation
- Genehmigte Anforderungen
- Kodierrichtlinien für die Wartung
- Detaillierter Zeitplan, mit Anzahl der Reviews und deren Ebenen

Risikoanalysen und Reviews müssen regelmäßig stattfinden und nicht wie in der Analyse und dem Entwurf am Ende der jeweiligen Phase.

Modultest und die Integration der Änderungen sind Teil der Implementierung.

Systemtest

Der Systemtest soll von einer, den Entwicklern unabhängigen, Organisation durchgeführt werden. Diese darf auf die Menge der Entwickler zurückgreifen, um ihren Bestand an Personal für den Test zu erhöhen. Die Durchführung des Systemtests durch die Testorganisation soll von Kunden und Endbenutzer mit begleitet werden.

Falls möglich, sollte der Test über ein vollkommen integriertes System erfolgen, nur wenn es nicht möglich ist in der Testeinrichtung ein komplettes System zu testen, sind Simulationen erlaubt. Zum Systemtest abgelieferte Teile müssen in einem Konfigurationsmanagement fixiert werden.

Die Qualitätssicherung muss garantieren, dass die gelieferten Änderungen tatsächlich genehmigt und zutreffend sind.

Wurden Änderungen an den Testfällen oder der Software nach deren Auslieferung gemacht, kann es notwendig sein während der Analyse Regressionstests durchzuführen, um eine definierte Systembeschreibung zu erhalten.

Abnahmetest

Abnahmetests müssen von Kunden, den Benutzern des Systems oder von einer, durch den Kunden bestimmten, dritten Partei bzw. Organisation durchgeführt werden. Der Abnahmetest soll sicherstellen, dass das System den Anforderungen des Kunden entspricht. Im Testumfang befindet sich neben dem implementierten Code auch die zugehörige Dokumentation. Bei Wartungsreleases kann es notwendig sein, zusätzliche Tests auf dem System des Kunden durchzuführen, um sicherzustellen, dass keine Fehler durch die implementierten Änderungen entstanden sind. Besonders soll dabei auf die Interaktion mit anderen Systemteilen geachtet werden.

Auslieferung

Neben der Auslieferung des ausführbaren Code müssen noch einige weitere begleitende Tätigkeiten durchgeführt werden. Um das Risiko bei der Installation eines neuen Systems zu minimieren, muss die Projektleitung eine alternative Installationsroutine planen. Diese sollte minimale Auswirkungen auf die Benutzer des bisherigen Systems haben. So sollten wichtige Arbeitsabläufe der Benutzer wie Jahresabschluss oder Inventur als nicht verfügbare Zeiträume für die Installation eingeplant werden.

Falls es zu größeren Änderungen in der Benutzeroberfläche oder Funktionserweiterungen kommt, müssen Schulungen für die Benutzer vor der Installation des neuen Systems eingeplant werden. Das Konfigurationsmanagement ist verantwortlich für die Sicherung der Daten, um einen Verlust dieser beim Umstieg auf das neue System zu verhindern.

2.3.3 CMMI

CMMI ist im Gegensatz zu dem vorangegangenen IEEE 1219 und ISO/IEC 12207 kein Standard sondern ein Vorgehensmodell. Es beschreibt nicht wie Software-Wartung betrieben werden soll, sondern was getan werden muss, damit ein Software-Projekt erfolgreich durchgeführt werden kann. Wenn man Wartung als Weiterentwicklung betrachtet greifen dieselben Mechanismen, die für eine Erstentwicklung gelten. Bei Wartung, die Fehler beheben oder aufgetretene Probleme lösen soll, wird der Einsatz von CMMI schwierig.

CMMI soll die Entwicklung von Software planbar machen. Dies ist bei Wartung, wenn man sie als Fehlerbehebung betrachtet, schwierig. Einerseits kann der Wartungsprozess sehr wohl geplant werden, andererseits kann über den Umfang der Wartung im voraus keine Aussage gemacht werden. Niemand kann sagen wann wie viele Fehler auftreten, welche Zeit benötigt wird, diese wieder zu beheben, und welche Kosten entstehen.

Planung ist der Kernansatz von CMMI. Durch sorgfältige Planung sollen Projekte zu einem befriedigenden Abschluss geführt werden können. Wartung ist in dem Sinne nicht als Projekt planbar. Es ist trotzdem notwendig Wartung zu planen, da man Ressourcen dafür bereitstellen muss. Typischerweise werden die Ressourcen, basierend auf Erfahrungswerten, geschätzt. Diese sind dann jedoch meist unzureichend, so dass nicht alle Änderungswünsche und Fehler behoben werden können. Dies kann durch Priorisierung aller eingehender Meldungen so gesteuert werden, dass zunächst wichtige Änderungen behoben werden.

Falls man den Wartungsumfang zu niedrig eingeschätzt hat oder große Änderungen anstehen, kann es notwendig werden das Budget für die Wartung aufzustocken oder zusätzliche Wartungsprojekte einzurichten. Falls die Schätzung zu hoch waren könnten auch unrentable Fehler behoben werden. Dies wird in der Regel sehr schnell durch Reduzierung der Ressourcen korrigiert.

Vorgehensmodelle wie das CMMI werden üblicherweise für die Durchführung von Projekten angewendet. Diese haben die Eigenschaft, dass sie zeitlich begrenzt sind. Wartung ist auch zeitlich begrenzt, aber ihre Laufzeit ist um ein Vielfaches länger als die entsprechenden Entwicklungsprojekte. Dies hat zur Folge, dass Entwicklungsprojekte über die insgesamt kürzere Projektdauer gesehen bis zum Projektende und damit langfristig geplant werden. Im Gegensatz dazu Wartung mit einer Laufzeit von vielen Jahren oder gar Jahrzehnten üblicherweise für das nächste Jahr und damit kurzfristig geplant wird.

Deshalb werden an die Durchführung von Wartungsprojekte andere Anforderungen gestellt. Bei der Betrachtung der Prozessgebiete des CMMI wird schnell ersichtlich, welche für Wartungsprojekte von Bedeutung sind:

- Anforderungsmanagement/Anforderungsentwicklung
- Technische Umsetzung
- Projektverfolgung und -steuerung
- Qualitätssicherung von Prozessen und Produkten
- Konfigurationsmanagement
- Produktintegration

- Risikomanagement
- Verifikation
- Validierung

Anforderungsmanagement/Anforderungsentwicklung

Eingehende Änderungswünsche oder Fehler müssen durch ein ausgeprägtes Anforderungsmanagement beurteilt und verwaltet werden. Es muss entschieden werden, welche Anforderungen sofort, welche langfristig und welche überhaupt nicht umgesetzt werden. In der Anforderungsentwicklung sollen die erhaltenen Anforderungen weiter verfeinert und Lösungsalternativen gesucht werden.

Technische Umsetzung

Vor den ersten Entwurfsschritten ist eine Auswahl der Lösungen basierend auf Untersuchung der Lösungsalternativen zu treffen. Schwerpunkt bei diesem Prozessgebiet ist der Entwurf sowie dessen Implementierung. Da CMMI keines vorschreibt kann hier jedes Vorgehensmodell eingesetzt werden um die Implementierung durchzuführen.

Projektverfolgung und -steuerung

Für den Erfolg von Projekten ist die Planung sehr wichtig, da der Wartungsumfang nur eingeschränkt planbar ist, wird es umso wichtiger den Projektverlauf der Wartung zu verfolgen und entsprechend zu steuern. Die Verfolgung und Steuerung ist ähnlich zu einem Entwicklungsprojekt. Es gibt nur den Unterschied, dass bei Entwicklungsprojekten gegen einen, zu Projektbeginn erstellten und damit langfristigen Projektplan verglichen wird. Bei der Wartung muss man dabei zwischen zwei Planungsebenen unterscheiden:

- Auf Ebene der einzelnen Anforderungen ergibt sich eine kurzfristige Planung und diese wird laufend aktualisiert.
- Auf Ebene der Wartung für das Gesamtsystem ergibt sich eine langfristige Planung ähnlich der bei Entwicklungsprojekten

Qualitätssicherung von Prozessen und Produkten

Da bei der Wartung gemeldete Fehler behoben, oder zusätzliche Funktionalität implementiert werden soll, und dabei keine neuen Fehler entstehen sollen, ist es verständlich warum ein besonderes Augenmerk bei der Wartung auf den Prozessen liegt.

Konfigurationsmanagement

Während der Wartung wird es schwierig mit den vielen verschiedenen Versionen des Systems umzugehen. Da sich die Wartung an mehreren Stellen und in unterschiedlichen Phasen befinden kann, entsteht ein komplexes Geflecht von Versionen mit unterschiedlich bereinigten Fehlern. In größeren Projekten wird es kaum möglich sein dies ohne Konfigurationsmanagement durchzuführen.

Produktintegration

Mit Produktintegration wird in CMMI die Integration der einzelnen Komponenten zu einem Produkt bezeichnet. Auf Ebene des Gesamtproduktes werden damit auch die Dokumentation, Schulung und Einführung erfasst, bzw. diese Komponenten ergeben erst durch Integration das Gesamtprodukt.

Risikomanagement

Neben der Priorisierung im Anforderungsmanagement, muss auch eine Risikoanalyse stattfinden. Auch wenn eine Anpassung wünschenswert oder wirtschaftlich wäre kann es sein, dass die Anforderung abgelehnt werden muss. Fehlende oder unvollständige Dokumentation kann dazu führen, dass Auswirkungen auf andere Systeme nicht ausgeschlossen werden können, und damit das Risiko für eine Änderung nicht eingeschätzt werden kann.

Verifikation

Prüfung der Ergebnisse auf Übereinstimmung mit ihrer Spezifikation. CMMI fordert das zur Verifikation Reviews und Tests eingesetzt werden.

Validierung

Prüfung der Ergebnisse auf Übereinstimmung mit den Anforderungen. CMMI fordert keine besondere Vorgehensweise, sie soll nur erfolgen.

3 Methodik

Für diese Diplomarbeit wurde eine Befragung in der Industrie durchgeführt. Es wurde dabei der Ist-Zustand bei der Wartung von Softwaresystemen in der Industrie erfasst. Besonderes Augenmerk wurde bei der Auswahl geeigneter Unternehmen auf das Vorhandensein eines schon mehrere Jahre in der Wartung befindlichen Softwaresystems gelegt. Neben dem Alter der Software wurde bei der Auswahl auch auf das Marktsegment der Unternehmen geachtet. Die Wahl fiel auf den Finanzdienstleistungssektor. Da hier Software im Großrechnereinsatz dominiert und diese Einsatzzeiten von mehreren Jahrzehnten aufweisen können.

Als interessierte Partner fanden sich fünf Unternehmen aus dem Wirtschaftsraum Mittlerer Neckar. Die fünf Unternehmen teilen sich in drei Banken und zwei Unternehmen, die Softwaresysteme für Banken erstellen, auf.

Als Interviewpartner wurden von den Unternehmen immer Personen aus dem Leitungsbereich bzw. Projektverantwortliche benannt.

Für die Datenerhebung wurden ursprünglich zwei Fragerunden geplant. In der ersten Runde sollten die Industriepartner einen vollstandardisierten Fragebogen nach [Konrad 2001] erhalten und bearbeiten. Danach sollte ein Interview zur Verifizierung der Daten und Beseitigung von Unklarheiten folgen. Da bei der Erhebung durch Fragebögen die Gefahr besteht, dass der Ersteller durch die Formulierung der Fragen seine Vorstellungen von Wartung in den Fragebogen projiziert, wurde entschieden die Datensammlung mit Leitfadenterviews nach [Mayer 2004] durchzuführen.

Für die Interviews wurde der schon erstellte Fragebogen als Leitfaden verwendet. Da es sich bei den interviewten Personen meist um Projektverantwortliche handelte wurden die Interviews als Experteninterview geführt.

Ablauf der Interviews

Die Interviewpartner erhielten jeweils eine Woche vor dem Gesprächstermin den Interview-Leitfaden. Durch die vorherige Information konnte eine eventuelle Prüfungssituation für den Industriepartner durch das Interview vermieden werden.

Die Interviews begannen mit einer Vorstellung aller Teilnehmer und ihrer entsprechenden Organisationen. Die Interviewpartner stellten ihre Unternehmen und die zu betrachtenden Sys-

teme vor. Von uns wurde der Studiengang der Softwaretechnik und das Ziel dieser Untersuchung noch einmal erläutert. Nach dieser Einleitungsphase wurden die Fragen entsprechend der Reihenfolge im Leitfaden beantwortet. Zur Protokollierung der Gespräche wurde der ursprünglich erstellte vollstandardisierte Fragebogen verwendet. Die freien Ausführungen der Interviewpartner wurden nur unterbrochen wenn sie zu weit von der eigentlichen Frage abwichen oder das zur Sprache gekommene Thema von uns später noch detailliert betrachtet wurde.

Einige der Interviews fanden mit mehreren Vertretern der jeweiligen Unternehmen statt, dies führte vereinzelt zu kurzen Diskussionen über die mögliche Antwort.

Die Interviews wurden auf 2-2,5h geplant und außer zwei Interviews, diese fanden als Gruppeninterviews statt und dauerten etwas länger, wurde dieser Zeitrahmen auch eingehalten.

Auswertung

Nach der Durchführung der Interviews wurden die erfassten Daten ausgewertet. Dabei wurden alle Aussagen der Interviewpartner miteinander verglichen. Danach erfolgte ein Vergleich der Aussagen mit den Anforderungen in der Literatur.

Qualitätssicherung

Nach den Interviews erhielten alle Teilnehmer die Möglichkeit die Auswertung und die darin beschriebene Situation zu kommentieren und ggf. bestehenden Fehleinschätzungen des Bearbeiters zu korrigieren.

4 Die untersuchten Systeme

Das folgende Kapitel ist das Ergebnis der Interviews, die mit Hilfe des Fragebogens im *Anhang A* geführt wurden.

Bei den fünf im Kapitel 1.4 vorgestellten Systemen gibt es einige grundsätzliche Unterschiede:

Ein System wird vom Unternehmen entwickelt und an externe Kunden verkauft, damit sind die Entwickler Teil des Produkteigentümers.

Bei zwei Systemen waren die Entwickler, zu Beginn der Entwicklung, Teil des Unternehmens und wurden in der Zwischenzeit in jeweils eine selbstständige Gesellschaft ausgegliedert. Die Unternehmen entwickeln und warten nun für ihren einzigen externen Kunden und sind nicht Produkteigentümer.

Ein System wurde von einer zwischenzeitlich ebenfalls ausgegliederten Entwicklungsabteilung erstellt, aber das Produkt entstand im Auftrag für die Muttergesellschaft und diese hat auch die Produkteigentümerschaft.

Beim letzten der fünf Systeme ist die Entwicklung Teil des Unternehmens und erstellte das System für interne Bedürfnisse des Auftraggebers und ist damit Teil des Produkteigentümers.

	Produkt-eigentümer	Kunde	System	Größe in kLOC	Entwicklungs-/Wartungsmodell	Entwicklungsbeginn	Aufwand in kPT
System 1	entwickelndes Unternehmen	extern	Banken-SW	-/-	Ja	1994	80/80
System 2	Kunde	extern	Banken-SW	500/25000	Ja	1970	-/1380
System 3	Kunde	extern	Vertriebssystem	190/250	Ja	1996	6/3
System 4	entwickelndes Unternehmen	intern	Kontenverwaltung	-/-	Nein	1970er	-/1
System 5	Kunde	extern	Saldierung	5/10	Ja	1960er	0,2/40

Tabelle 3 Übersicht der Systeme

In der Spalte mit den Größenangaben in *Tabelle 3* ist der erste Wert die Größe des Systems nach der Erstentwicklung und der zweite Wert der aktuelle Stand. Der Aufwand hat einmal den Wert für die Erstentwicklung und dann für die Wartung. Die gewählte Reihenfolge für die nun folgende Vorstellung hat nichts mit der oben beschriebenen Reihenfolgen in der *Tabelle 3* zu tun und ist willkürlich gewählt.

4.1 System A

4.1.1 Vorstellung des Systems

Eines der betrachteten Systeme ist ein prozessorientiertes Vertriebssystem. Es handelt sich hierbei um eine Client-Server-Anwendung. 1996 wurde mit der Modellierung der Geschäftsprozesse begonnen. Bis Ende 1998 wurde die Konzeption abgeschlossen. Nach einer Machbarkeitsstudie wurde entschieden, die Entwicklung nicht an einen externen Anbieter zu vergeben, sondern im eigenen Haus zu entwickeln. Bei der Realisierung wurde bewusst die Entscheidung getroffen Java als Programmiersprache einzusetzen. Da es im Hause keine Erfahrungen mit objektorientierter Programmierung gab, wurde das prozessorientierte Vertriebssystem auch als Pilotprojekt für den Einsatz von objektorientierten Programmiersprachen angesehen.

Im März 2001 wurde eine erste Version in Betrieb genommen. Für die Entwicklung entstand ein Aufwand von 6100 Personentagen. Seither wurden 3000 Personentage in die Wartung des Systems investiert.

Da die Entwicklungsabteilungen inzwischen zu einer eigenständigen Gesellschaft umgewandelt wurden ergeben sich einige Besonderheiten:

- Die Gesellschaft ist nicht Produkteigner des Softwaresystems, sondern der Kunde.
- Die Gesellschaft bietet der Muttergesellschaft Entwicklungskapazität und nicht Produkte an
- Die Gesellschaft hat kein eigenes Budget zur Verfügung, alle Tätigkeiten müssen sich in Aufträgen der Fachabteilungen spiegeln, dies wird insbesondere bei perfekter Wartung zum Problem. Um diese durchführen zu können, muss der Kunde von deren Notwendigkeit überzeugt werden.

4.1.2 Die Rollen

Der Produktbetreuer

Der Produktbetreuer ist die einzige Schnittstelle zur Kommunikation zwischen Benutzern aus den Fachbereichen und Entwicklern in den Prozessbereichen. Jeder Produktbetreuer ist für mindestens ein Produkt verantwortlich. Produktbetreuer sind Mitarbeiter mit erweiterten Kenntnissen über das Produkt, sogenannte Poweruser. Sie haben neben ihren regulären Tätigkeiten in den Fachbereichen zusätzlich diese Schnittstellenfunktion.

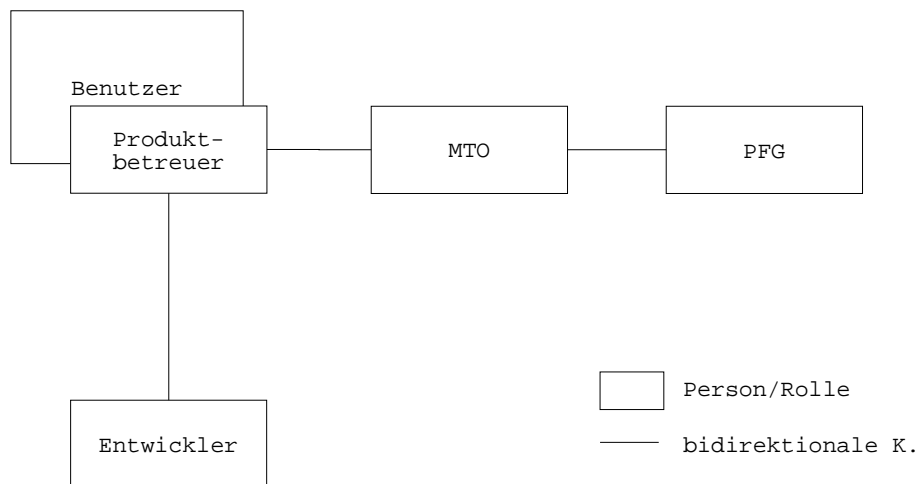


Abbildung 6 Kommunikationswege System A

Der Produktbetreuer kontaktiert den zuständigen Prozessbereich in der Entwicklung über eingerichtete Sammelrufnummern oder per Email. Für jedes Produkt wird vom Betreuer eine Produktakte geführt, in welcher anstehende Änderungen aufgelistet werden.

Die Verantwortung über die Leerung der Produktakte wechselt in regelmäßigen Abständen. Dieser Vorgang wird in Anlehnung an das regelmäßige Reinigen der Außenanlagen in Süddeutschland, „die Kehrwoche“ genannt.

Neben der Kommunikation vom Kunde, über den Produktbetreuer, zur Entwicklung, gibt es auch den umgekehrten Weg: Die Entwickler informieren den Produktbetreuer über neue Versionen und dieser übernimmt dann die Verteilung und Unterrichtung der Benutzer.

Management-Technologie-Organisation bzw. Projekt-Freigabe-Gremium

Neben den Produktbetreuern gibt es zwei weitere Gremien die Einfluss auf den Wartungsprozess haben. Das eine Gremium ist die Management-Technologie-Organisation (MTO) und das andere das Projekt-Freigabe-Gremium (PFG). An das MTO werden vom Produktbetreuer Meldungen über geplante Wartungsaktivitäten übermittelt. Falls diese Aufwände fünf Personentage überschreiten, werden sie von der MTO dem PFG zur Entscheidung vorgelegt. Das Projekt-Freigabe-Gremium besteht aus Mitgliedern des Vorstandes.

Neben der Genehmigung von laufenden Wartungsaktivitäten beschließt das PFG auch den Geschäftsjahresplan. Für diesen werden die Anforderungen der Fachbereiche gesammelt und dann einmal jährlich beschlossen, welche der Anforderungen realisiert werden.

Nach Aussage der Interviewpartner hat dies zur Folge, dass es nur in Ausnahmefällen zu unterjährigen Vorhaben kommt, die vorher nicht im Geschäftsjahresplan vorgesehen waren.

In *Abbildung 6* sind die Kommunikationswege zwischen Benutzer und Entwickler dargestellt.

4.1.3 Der Wartungsprozess

Bei einer Fehler- oder Problemmeldung wendet sich der Benutzer an den Produktbetreuer. Dieser versucht das aufgetretene Problem zu beheben oder eine provisorische Lösung zu erstellen. Falls dies nicht möglich ist, wird ein neuer Eintrag in der Produktakte angelegt. Falls das Problem schneller gelöst werden muss, wird der Produktbetreuer den Prozessbereich zusätzlich über die eingerichtete Sammelrufnummer kontaktieren.

Nach der Aufwandsschätzung durch die Entwickler gibt es abhängig von der Höhe verschiedene Vorgehensweisen:

- bis 1 Personentag

Die Fachabteilung vergibt eine Auftragsnummer und das aufgetretene Problem wird gelöst. Alle Aktivitäten, die darüber hinausgehen, müssen an die MTO gemeldet werden.

- bis 3 Personentage

Der Fachbereich entscheidet, ob das aufgetretene Problem gelöst wird und vergibt eine Auftragsnummer.

- bis 5 Personentage

Der Fachbereich übergibt die Entscheidung an die übergeordnete Fachorganisation.

- über 5 Personentage

Die MTO bereitet die Dokumentation vor und übergibt an das PFG. Dieses tagt im monatlichen Rhythmus.

Nach Erteilung des Auftrages wird vom Entwickler ein Fachkonzept erstellt und es folgt die Implementierung.

Danach erfolgt der Test. Die Entwickler führen Modultest durch und der Kunde ist für den Systemtest verantwortlich. Der Kunde bestimmt die Kriterien, die erfüllt werden müssen, um das System abzunehmen. Dies führt dazu, dass die Behebung von Fehlern nach der Abnahme, im Rahmen von korrektiver Wartung, dem Kunden in Rechnung gestellt wird, da er zuvor das System als Ganzes abgenommen hat.

Im Gegensatz zu Fehlern, werden Wünsche der Benutzer nur als Entwicklungsaufträge vergeben. Diese müssen in den Geschäftsjahresplan aufgenommen werden. Aktuell wurden keine neue Anforderungen genehmigt, es werden nur Fehler behoben. Anpassungen und Wünsche können somit nur durchgeführt werden, falls diese den Rahmen von fünf Personentagen nicht überschreiten.

4.1.4 Besonderheiten

Da die Entwicklung nicht Produkteigner ist, ergeben sich bei diesem System einige Besonderheiten:

- Die Entwicklung wird nur auf Anfrage des Kunden tätig.
Perfektive Wartung, um die Wartbarkeit zu erhöhen bzw. die Kosten für zukünftige Änderungen zu senken, kann nur im Auftrag des Kunden gemacht werden. Die Entwickler müssen den Kunden von den Vorteilen und der Notwendigkeit von perfekterer Wartung überzeugen. Dies hat, nach Aussage der Interviewpartner zur Folge, dass die Wartungskosten im Laufe der Zeit ansteigen. Erst wenn der Kunde bereit ist, Mittel für perfekte Wartung bereitzustellen, wird diese durchgeführt.
- Der Produkteigner ist für die Abnahme des implementierten Systems verantwortlich.
Er erstellt die Abnahmekriterien und überprüft diese.
- Kehrwoche
Die Entwickler leeren in regelmäßigen Abständen die Produktakte und gewährleisten so eine stetige Bearbeitung der Kundenanfragen.

4.2 System B

4.2.1 Vorstellung des Systems

Das nächste System ist eine Termingeldkontenverwaltung. Es wird seit Mitte der 1970er Jahre verwendet. Es handelt sich um eine Entwicklung im und für das eigenen Unternehmen. Während dieser langen Laufzeit hat sich inzwischen ein kompletter Wechsel der Entwickler vollzogen. Aufgrund dessen konnten in den Interviews keine Aussagen zur Erstentwicklung gemacht werden. Trotz dieser Umstände scheint die Dokumentation so ausführlich und die Komplexität des Systems handhabbar zu sein, dass die neuen Entwickler in der Lage waren sich in kurzer Zeit in das System einzuarbeiten.

Die Termingeldkontenverwaltung ist ein Batch-System. Die Benutzer geben während der Geschäftszeit ihre Daten ein und diese werden dann nachts verarbeitet. Daraus ergeben sich einige besondere Aspekte:

- Die Datenerhebung bzw. -eingabe muss sehr genau geprüft werden.
- Die Daten werden in Datenbanken gesammelt und nach Geschäftsschluss verarbeitet. Falls Eingaben nicht korrekt waren, wird dies erst bei der Verarbeitung nach Geschäftsschluss bekannt. Fehler können hier schnell zu Verzögerungen von einem Arbeitstag führen.
- Die zu verarbeitende Datenmengen machen es notwendig bei der Entwicklung auf eine möglichst hohe Verarbeitungsgeschwindigkeit zu achten.
- Ausfälle des Systems oder falsche Verarbeitung können sehr schnell zu hohen finanziellen Schäden führen.
- Die lange Laufzeit des Systems bringt es mit sich, dass die heutigen Entwickler nicht an der Erstentwicklung beteiligt waren und somit das System ausführlich dokumentiert sein muss.

4.2.2 Die Rollen

Der IT-Kundenmanager

Der IT-Kundenmanager ist neben dem Customer Care Center die zentrale Stelle für alle Anfragen der Benutzer an die Entwickler. Der Kundenmanager plant den Einsatz der ihm unterstellten Entwickler. Jeder Kundenmanager ist verantwortlich für mindestens ein Produkt. Jeder Entwickler ist wiederum an mindestens einem Produkt beteiligt.

Daraus kann sich folgende Konstellation ergeben: Ein Kundenmanager ist für mehrere Entwickler verantwortlich und Entwickler können mehrere Kundenmanager haben. Die Kundenmanager müssen sich in regelmäßigen Sitzungen über den Einsatz der Entwickler verständigen.

Das Customer Care Center

Das Customer Care Center ist im täglichen Betrieb die erste Anlaufstelle für die Benutzer der Termingeldkontenverwaltung, es bietet an zentraler Stelle Support an. Die Mitarbeiter des Customer Care Center haben eine ausführliche Fehlerdatenbank mit entsprechenden Lösungen zur Verfügung und können sehr schnell und umfassend bei der Fehlerbeseitigung helfen. Neue Fehlermeldungen oder Probleme werden an die Entwickler weitergeleitet.

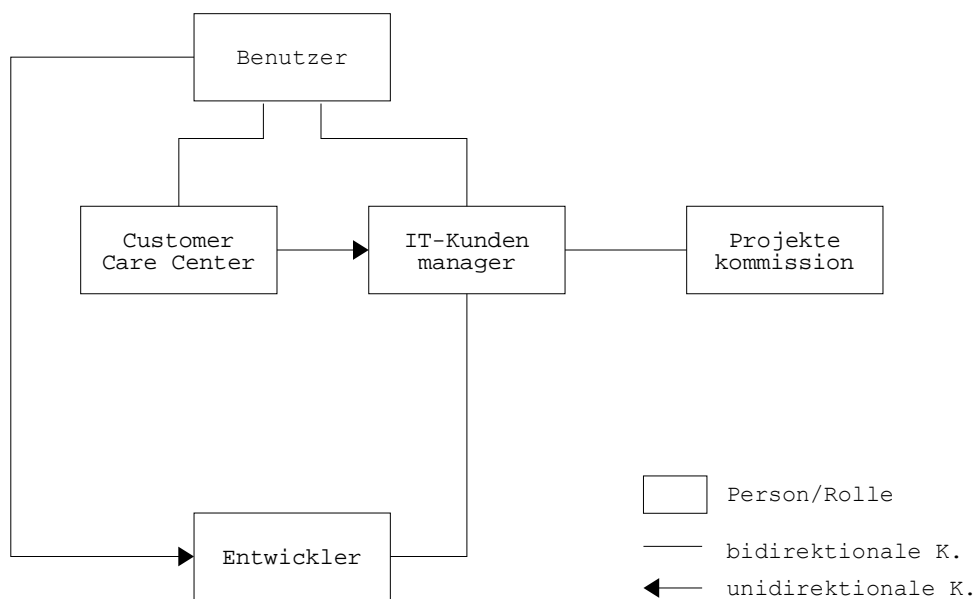


Abbildung 7 Kommunikationswege System B

In *Abbildung 7* sind die Kommunikationswege zwischen Benutzer und Entwickler dargestellt.

4.2.3 Der Wartungsprozess

Bei einer Fehler- oder Problemmeldung kontaktiert der Benutzer das Customer Care Center. Wird keine Lösung gefunden oder tritt ein neuer Fehler bzw. ein neues Problem auf, wird das Customer Care Center die Entwickler kontaktieren und eine Analyse der Meldung und mögliche Lösungen in Auftrag geben.

Da die Termingeldkontenverwaltung schon lange Jahre von den selben Entwicklern betreut wird, hat sich im Laufe der Zeit ein zusätzlicher Kommunikationsweg ergeben: Die Benutzer bitten die Entwickler auch direkt um Hilfe oder melden Fehler. Falls der Aufwand für die Beseitigung bzw. die Hilfe unter einem Arbeitstag eingeschätzt wird, helfen die Entwickler sofort.

Nach der Aufwandsschätzung durch die Entwickler gibt es abhängig von der Höhe folgendes Vorgehen:

- bis 5 Personentage

Der Benutzer, bzw. die Abteilung, entscheidet ob der Aufwand bezahlt wird und vergibt den Auftrag ohne weitere Prüfung. Der Entwickler informiert den zuständigen Kundenma-

nager, um den Aufwand in die weitere Planung aufzunehmen. In diese Kategorie fallen üblicherweise Fehlerbehebung und Kleinaufträge.

- bis 50 Personentage

Der Kundenmanager beginnt Verhandlungen mit der Kundenabteilung um den Zeitplan zu erstellen, gegebenenfalls werden auch bestehende Aufträge neu priorisiert und deren Zeitpläne verändert. Parallel zu diesen Verhandlungen muss auch mit den anderen Kundenmanagern die Planung abgestimmt und gegebenenfalls Prioritäten laufender Projekte verändert werden.

- ab 50 Personentage

Der Kundenmanager muss die Entscheidung an die Projektekommision übergeben, Dieses Gremium ist direkt dem Vorstand unterstellt und entscheidet bei Projekten dieser Größe.

- Ab 150 Personentage muss der Kundenmanager zusätzlich noch eine Vorstudie erstellen.

Nach Erteilung des Auftrages wird vom Entwickler ein Fachkonzept erstellt und danach folgt die Implementierung mit Test und Einsatz. Bei kleineren Fehlern bzw. Änderungen wird kein Fachkonzept erstellt sondern direkt implementiert. Falls notwendig wird die Dokumentation angepasst oder neu erstellt.

Im Gegensatz zu Fehlern oder Problemen werden Wünsche der Benutzer immer an den zuständigen Kundenmanager gerichtet. Diese führen auch selbständig Gespräche mit den Benutzern um neue Bedürfnisse frühzeitig zu erkennen und an die Entwickler weiterzuleiten.

Die Kundenmanager haben neben der Koordination der Entwickler auch folgende Aufgaben:

- Sie entscheiden in erster Instanz ob Wünsche der Kunden umgesetzt werden.
- Sie besprechen mit den Entwicklern den Inhalt von neuen Releases und bündeln die Wünsche und Anforderungen der Benutzer.

Die Entwickler erstellen bei Wünschen und anderen Anforderungen (wie gesetzliche Änderungen, oder wirtschaftliche Veränderungen beim Kunden) ebenfalls eine Aufwandschätzung. Nach dieser Aufwandschätzung und deren Einteilung in eine der drei oben genannten Stufen folgt das weitere Vorgehen dem der Fehler- oder Problemmeldung.

4.2.4 Besonderheiten

Bei diesem Projekt gibt es einige Besonderheiten:

- Trotz des Alters des Systems soll es eine vollständige Dokumentation geben. Mitte der 1980er Jahre wurde das System auf die Jahr-2000-Problematik hin untersucht und angepasst. Dabei entstand eine Dokumentation, die nach eigenen Angaben ungefähr 70% des Systems erfasste. Nach einem Wechsel in der Fachverantwortung des Systems, Anfang der 1990er Jahre, wurde bis zum heutigen Zeitpunkt fast das gesamte System (98%) dokumentiert. Änderungen und Erweiterungen wurden sukzessive dazu genutzt, undokumentierte Teile zu dokumentieren. Die Qualität der damals bestehenden Dokumentation scheint so gut zu sein, dass beim Wechsel der Entwickler diese sehr schnell in der Lage waren sich in das System einzuarbeiten und einen umfassenden Einblick in das System erhalten haben.
- Das System wird heute von den zwei verantwortlichen Entwicklern komplett überblickt und beherrscht. Dieser Umstand führt dazu, dass Analysen und Aufwandsabschätzungen für gesetzliche Anpassungen oder Wünsche der Benutzer sehr genau und schnell erstellt werden können.
- Es wird, aufgrund des übersichtlichen Systems und der geringen Anzahl von Personen, die mit der Wartung beschäftigt sind, kein Vorgehensmodell im Sinne dieser Untersuchung verwendet. Gleichwohl sind den zwei Entwicklern die Inhalte des im Hause bestehenden Vorgehensmodells bekannt. Sie werden durch die Aufgabenteilung in Fachverantwortung, mit Erstellung von Analysen und Konzepten, und die Implementierung mit Test simuliert.
- Beim Einsatz von Werkzeugen während der Entwicklung gibt es eine weitere Besonderheit: Es wird kein Konfigurationsmanagementwerkzeug eingesetzt. Die zwei Entwickler haben ihre Aufgabengebiete so weit getrennt, dass nur in Vertretungsfällen der eine das Gebiet des anderen bearbeitet.
- Die Entwickler haben die Möglichkeit, neben den eigenen Testfällen, Daten aus dem laufenden Betrieb in das Entwicklungssystem zu kopieren und somit über Jahre gewachsenen Datenbeständen für den Test einer neuen Version einzusetzen.

4.3 System C

4.3.1 Vorstellung des Systems

Aus dem Sachgebiet Saldierung stammt das nächste der betrachteten Systeme. Es dient für die Hauptbuchhaltung des Kunden und dessen Kundenbetreuung und enthält grob:

- Die Umsatzverarbeitung: Aufnahme, Plausibilisierung, Steuerung (Kontenzuordnung) der Buchungen (Geldeingang, interne Buchungen)
- Die eigentliche Saldierung (Buchung auf Kundenkonten und Sonderkonten und Zuordnung der Hauptbuchkontierung)
- Automatische Buchungen (im Wesentlichen Gebühren, Agio, Disagio)
- Automatische Korrekturen bei Rückevaluierung (im Wesentlichen Spar- und Darlehenszinsen, Gebühren und Zuteilungswerte)
- Aufbau der Schnittstelle zur Hauptbuchhaltung (Finanzbuchhaltung)
- Abstimmung Zugänge – Saldierung – Hauptbuchzugang

Es ist das älteste der betrachteten Systeme. Die Entwicklung wurde in den 1960er Jahren begonnen. Dieses Ursprungssystem war ein elektronisches 1:1 Abbild der damals bestehenden Buchhaltung auf Papier. In den 1980er Jahren wurden schrittweise Modifikationen durchgeführt. 1989 wurde die Umsatzverarbeitung, Hauptbuchzuordnung und Schnittstelle zum Hauptbuch von Grund auf neu entworfen.

Die Anpassungen an die Jahr-2000-Problematik (1999) und die EURO-Einführung (2001) waren weitere verhältnismäßig große Änderungen.

Von 2002 bis 2003 erfolgten Funktionserweiterungen und die Aufnahme (Integration und Migration) von Daten und Funktionen aus einem externen System.

Für die nähere Zukunft sind keine weiteren Änderungen der Software geplant. Die wesentlichen Änderungen bestehen derzeit in Anpassungen für neue Produkte und gesetzliche Maßnahmen. In einer Studie wird derzeit geklärt, welche Konsequenzen aus der Reorganisation der Arbeitsabläufe der Darlehensbearbeitung entstehen. Es wird auch geprüft, ob das System durch Standardsoftware ersetzt oder ergänzt werden kann.

Trotz des Alters des Systems und des beträchtlichen Aufwands hat das System nur eine Größe von ungefähr 30 kLOC. Der größte Teil des Systems wurde in PL/1 implementiert. Kleinere Teile wurden in SIRON programmiert.

4.3.2 Die Rollen

IT-Kontaktstelle

Die IT-Kontaktstelle ist die einzige Schnittstelle zwischen den Benutzern aus den Fachbereichen und den Entwicklern in der IT-Linie. Mitarbeiter werden durch Schulung mit erweiterten Kenntnissen ausgestattet und bilden dann die IT-Kontaktstelle. Sie haben, neben dieser Schnittstellenfunktion, keine zusätzlichen Aufgaben in den Fachbereichen. Neben Fehler- und Problemmeldungen ist die IT-Kontaktstelle auch in den Test des implementierten Systems eingebunden. Sie betreut die Einführung in ihrem Fachbereich und sie tritt gegenüber der

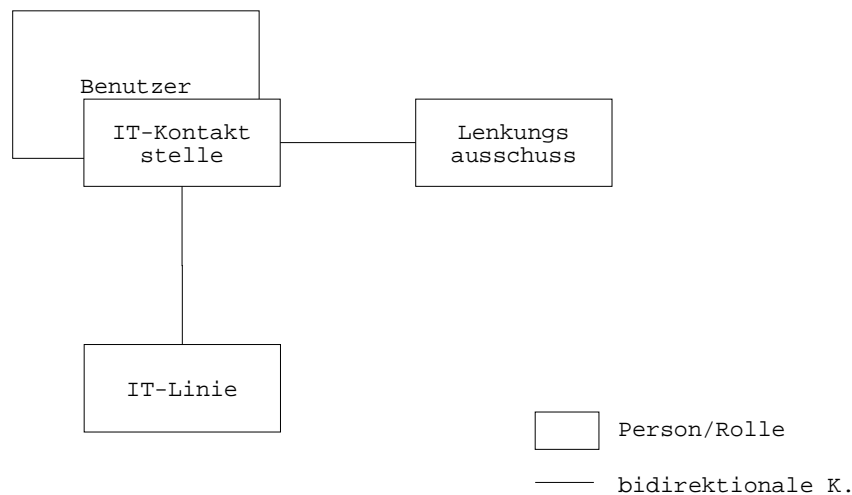


Abbildung 8 Kommunikationswege System C

Entwicklung als Auftraggeber auf. Die Kontaktstelle besetzt, als Auftraggeber, paritätisch mit der IT-Linie, als Auftragnehmer, das Change Control Board.

Lenkungsausschuss

Der Lenkungsausschuss tritt im täglichen Betrieb nur bei Projekten und Kleinaufträgen, die an ein Budget gebunden sind oder über die IT-Linie abgewickelt werden, in Erscheinung. Projekte sind Anforderungen, die zu Aufwänden mit mehr als 60 Personentagen führen. Der Lenkungsausschuss ersetzt die IT-Kontaktstelle bei globalen Steuerungs-/Entscheidungsfunktionen. Die Kommunikation und fachliche Steuerung (Vorgaben, Kontrolle, Test, usw.) läuft auch weiterhin über die Kontaktstelle.

IT-Linie

Die IT-Linie ist die Gesamtheit der Entwicklung. Abhängig von der Größe des Auftrages wird auf Ebene der Abteilung oder des Bereichs kommuniziert. Bei Auftrags-/Problemmeldung läuft die Hauptkommunikation über den Sachgebietsleiter zum Entwickler. Bei Projekten werden Projektleiter benannt, die dann mit den Sachgebietsleitern kommunizieren.

In *Abbildung 8* sind die Kommunikationswege zwischen Benutzer und IT-Linie (Entwickler) dargestellt.

4.3.3 Der Wartungsprozess

Falls der Kunde Wünsche hat, bei der Bedienung Fehler oder Programmabbrüche entstehen, so werden diese Einzelanforderungen bei der IT-Kontaktstelle als Problemmeldung registriert. Die Meldung wird, in Abhängigkeit des prognostizierten Aufwands, zu einem Kleinauftrag (bis 60 Personentage (PT)) oder zu einer Projektanforderung. Ab einer Größe von 20 PT muss zusätzlich eine Wirtschaftlichkeitsprüfung erfolgen.

Ein Projekt wird nur durchgeführt wenn es in der Jahresplanung genehmigt wurde. Kleinaufträge können durchgeführt werden wenn sie das im Jahresplan vorgesehene Budget nicht überschreiten. Für Problemmeldungen wird ebenfalls ein Budget in der Jahresplanung vorgesehen. Problemmeldungen bewegen sich bei Aufwänden $<5PT$.

Nach Erteilung des Auftrages wird von den Sachgebiets-/ Gruppenleitern eine Softwarebeschreibung oder Änderungsmeldung erstellt. Die Entwickler erstellen danach einen Entwurf und prüfen ob schon geplante oder zusätzlich notwendige Verbesserungen mit in den Auftrag integriert werden müssen. Diese werden dann mit in die Anforderungen aufgenommen.

Die Implementierung wird von den Entwicklern in Modultests überprüft. Danach folgt eine technische Kontrolle der Implementierung gegen die Anforderungen durch die Sachgebiets- und Gruppenleiter. Die Mitarbeiter der IT-Kontaktstelle machen in der Regel die fachliche Abnahme. In Ausnahmefällen (z.B. Großprojekte) werden Testgruppen auch mit Mitarbeitern außerhalb der Kontaktstelle besetzt. Bei diesen Tests unterstützt die IT dann die Kontaktstellen.

Die Qualitätssicherung ist eine eigenständige Abteilung, sie überprüft die technische Korrektheit und führt den Einsatz ins Produktivsystem durch.

Der Sachgebietsleiter trägt die Verantwortung, dass alle Änderungen oder Erweiterungen in die Dokumentation einfließen.

4.3.4 Besonderheiten

Bei diesem System bestehen einige Besonderheiten:

- Für die Wartung wurde ein wöchentlich wiederkehrendes Wartungsfenster eingerichtet, falls es notwendig wird das System kurzzeitig vom Netz zu nehmen. Ansonsten können Änderungen jederzeit eingepflegt werden.

Die gesamte IT (Entwicklung und Rechenzentrum) ist in eine eigene Gesellschaft ausgegliedert. Die Verrechnung von angeforderten Leistungen (Fehlerbehebung, adaptive und perfekte Wartung) erfolgt an die auftraggebenden Firmen, die auch jeweils Eigentümer der Software sind.

- Jeder Fachbereich erhält bzw. plant ein jährliches Budget für Fehlerbehebung ein.
- Adaptive Wartung kann nur über Projekte oder Kleinaufträge durchgeführt werden.
- Perfektive Wartung kann nur innerhalb von Projekten durchgeführt werden.

4.4 System D

4.4.1 Vorstellung des Systems

Das nächste betrachtete System ist eine Bankensoftware, die alle auftretenden Vorgänge in einer Bank widerspiegeln kann. Die Entwicklung wurde 1994 begonnen. Eine erste Version erschien 1997. Ein Unterschied zu anderen betrachteten Projekten ist, dass das Unternehmen Produkteigner ist, und das Produkt für den Markt und nicht für das eigene Unternehmen entwickelt wird.

Seit der Entwicklungsphase wird alle sechs Monate ein Update für die Kunden zur Verfügung gestellt. Die Kunden sind verpflichtet mindestens jedes zweite Update durchzuführen. Beinhaltet ein Update Anpassungen, welche auf gesetzlichen Anforderungen basieren, so muss dieses zusätzlich und ausnahmslos durchgeführt werden.

Die regelmäßige Weiterentwicklung des Produktes führte dazu, dass inzwischen für die Wartung annähernd so viel Aufwand entstand wie für die Erstentwicklung, jeweils 500 Mannjahre.

Neben der Entwicklung wird den Kunden auch eine Vollversorgung angeboten:

Sie beinhaltet neben der Software optional das Outsourcing im Rechenzentrum sowie den Support.

Mit dem Kunde werden Wartungsverträge über jeweils fünf Jahren Laufzeit abgeschlossen. In diesen werden Pauschalen für korrektive und adaptive Wartung vereinbart. Bei Kosten für perfektive Wartung wird individuell entschieden, wer diese trägt.

Für die nähere Zukunft sind einige größere Anpassungen der Software geplant, darunter fällt auch die sogenannte Neue Baseler Eigenkapitalvereinbarung, kurz Basel II genannt. Diese Anpassungen werden ungefähr 20% des bestehenden Systems betreffen.

4.4.2 Die Rollen

Poweruser/ Hotline

Der Poweruser ist ein besonders berechtigter Benutzer mit administrativen Rechten beim Kunden. Er ist die erste und wichtigste Anlaufstelle für die Benutzer. Er kann, abhängig vom Supportvertrag, auch die einzige Schnittstelle zwischen den Benutzern und den Entwicklern sein. Von der Größe des Kunden bzw. des Supportvertrags abhängig wird auch eine Hotline installiert. Beide Stellen informieren die Entwickler über bestehenden Probleme. Daneben gibt es aber auch Kunden, die direkten Kontakt zu den Entwicklern wünschen. Exemplarisch sind in *Abbildung 9* zwei Kunden dargestellt um den Unterschied bei der Ausprägung von Supportverträgen zu verdeutlichen.

Fach-Arbeitskreis

Der Fach-Arbeitskreis besteht aus Vertretern aller Kunden und der Entwicklung. Er trifft sich in regelmäßigen Abständen, fünf- bis sechsmal im Jahr. Dabei wird die Richtung für die Weiterentwicklung des Basissystems festgelegt.

Falls größere Änderungen anstehen, werden diese im Fach-Arbeitskreis aufeinander abgestimmt und die Frage der Kostenübernahme geklärt. Der Fach-Arbeitskreis diskutiert außerdem, welche bestehende Funktionalität in das Basissystem aufgenommen wird und welche nicht. Die nicht aufgenommene Funktionalität bleibt weiter nur in einer zusätzlich kostenpflichtigen Kundenversion verfügbar. Alle Vorschläge werden an das Produktplanungsgremium weitergeleitet.

Das Produktplanungsgremium besteht aus den IT-Vorständen der Kunden und den IT-Leitern des Unternehmens. Es trifft sich einmal im Jahr und beschließt die Vorschläge der Fach-Arbeitskreise.

4.4.3 Der Wartungsprozess

Vorgangsmeldung

Bei einem Fehler oder Programmabbruch, meldet sich der Benutzer beim Poweruser oder der Hotline und schildert sein Problem. Falls diese Ansprechpartner keine Lösung anbieten können, erstellen sie eine Vorgangsmeldung in einer Datenbank bei den Entwicklern. Bei Kunden, die direkten Kontakt mit den Entwicklern haben, wird diese Vorgangsmeldung vom angesprochenen Entwickler erstellt. Vorgangsmeldungen können auch von den Entwicklern, ohne Anregung vom Kunden, selbst erstellt werden. Dies soll aber nur in Ausnahmefällen ge-

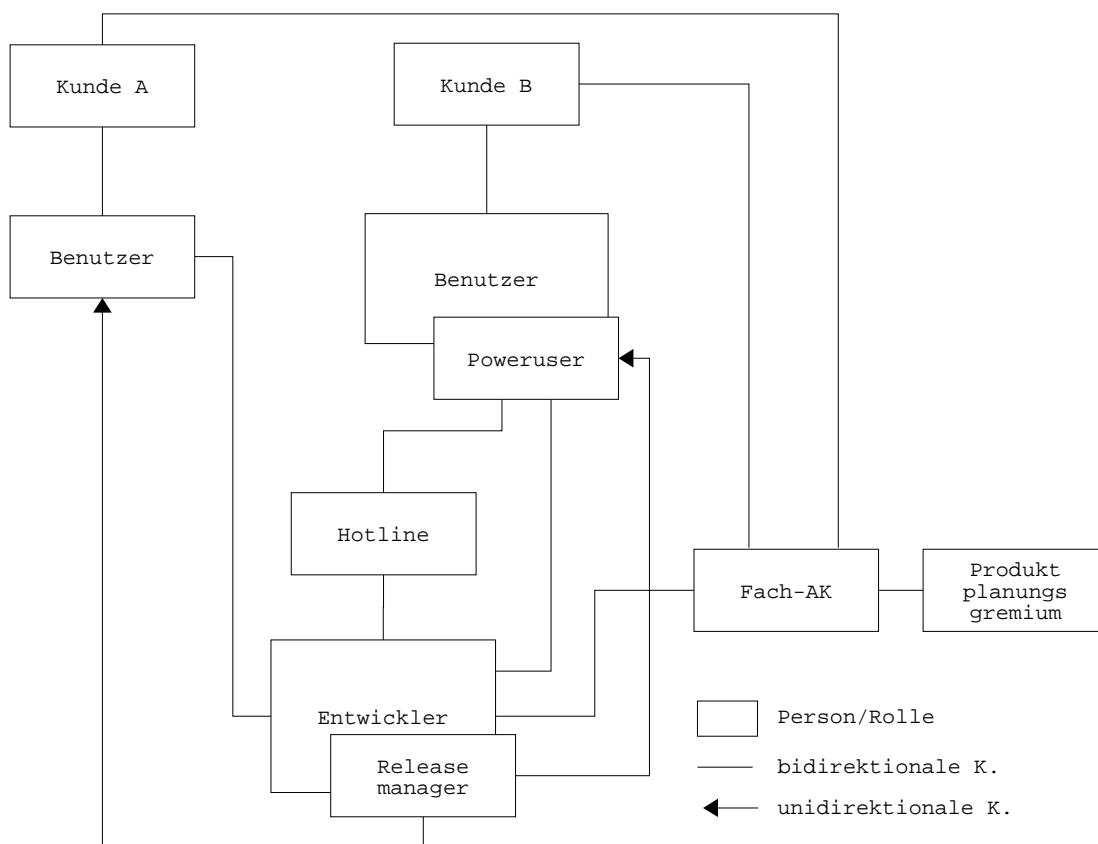


Abbildung 9 Kommunikationswege System D

schehen, da Schwierigkeiten mit der Abrechnung entstehen können. Zur Analyse des Fehlers besteht für die Entwickler die Möglichkeit direkt auf das System des Kunden zuzugreifen und dadurch schwierig zu beschreibende Problem leichter zu analysieren.

Die Vorgangsmeldung ist der zentrale Bestandteil der Wartungsaktivitäten, alle Tätigkeiten werden in dieser vermerkt. Sie besteht im Einzelnen aus:

- Beschreibung des Kunden bzw. Entwicklers
- Klassifizierung durch den Entwickler
- Release, in welchem die Lösung ggf. realisiert wird

Jede neu eingetroffene Vorgangsmeldung wird von einem fachlich verantwortlichen Mitarbeiter in eine der vier Klassen eingestuft. Diese Klassen gehen von critical bis minor. Als kritisch eingestufte Meldungen müssen innerhalb von wenigen Stunden behoben werden. Als minor eingestufte Vorgangsmeldung werden bei Gelegenheit in einem zukünftigen Release implementiert.

Nach Eingang einer Vorgangsmeldung, und deren Klassifizierung, wird durch ein Change Control Board, welches aus einem Vertreter der Datenbank, der Systementwicklung, dem Releasemanager und ggf. einem Spezialisten aus dem Fachbereich besteht, das weitere Vorgehen abgestimmt.

Nach Konzeption und Implementierung folgt ein mehrstufiger Test der Änderungen. Neben den Tests der Entwickler selbst, wird jedes Release vor Veröffentlichung durch eine Gruppe von Entwicklern, die nicht an den Änderungen beteiligt waren, abgenommen. Der Kunde testet das Update zusätzlich noch auf seinem eigenen System bevor er es in Betrieb nimmt.

4.4.4 Besonderheiten

Bei diesem System bestehen ebenfalls einige Besonderheiten:

- Abgesehen von kritischen Fehlern werden alle Änderungen am System gesammelt und in regelmäßigen Releases an die Kunden ausgeliefert. Für jedes Release wird ein Entwickler bestimmt, welcher die Rolle des Releasemanagers bekleidet. Der Releasemanager koordiniert alle Tätigkeiten in den einzelnen Fachbereichen. Er informiert die Kunden über geplante Releaseumfänge und dient den Kunden als Ansprechpartner für Anfragen, die das kommende Release betreffen.
- Die Releasetermine sind fest, ggf. muss der Umfang reduziert werden, falls Implementierungen nicht planmäßig fertig gestellt werden können. Als Beispiel die Größe des letzten Releases: Ungefähr 900 abgearbeitete Vorgangsmeldungen und 200 behobene Fehler.
- Für die Wartung wird kein Vorgehensmodell verwendet, sondern basierend auf dem ITIL Leitfaden entstandene Best Practice Beschreibungen verwendet. Der Unterschied besteht

darin, dass nicht das „Wie“, sondern das „Was getan werden muss“ beschrieben ist. Dementsprechend wurde auch nicht der Entwicklungs- oder Wartungsprozess im Allgemeinen zertifiziert, sondern das Verfahren „lückenlose Dokumentation über was, wann getan wurde“ zu erstellen. Neben den Kunden, vertreten durch den Bankenverband, tritt auch das Bundesamt für Finanzdienstleistungsaufsicht als Kontrollinstanz auf.

Die Wartungsverträge decken nur die Fehlerbehebung und Anpassung an gesetzliche Anforderungen ab. Die Kostenübernahme für weitere Anpassungen oder perfektive Wartung wird individuell entschieden.

4.5 System E

4.5.1 Vorstellung des Systems

Das zuletzt vorgestellte System ist eine Bankensoftware eines Rechenzentrums, die alle auftretenden Vorgänge in einer Bank widerspiegeln kann. Die Entwicklung wurde 1970 begonnen. Ein Unterschied zu den anderen betrachteten Projekten ist, dass es für dieses zum jetzigen Zeitpunkt ein definiertes Ende gibt. Die Software wird im 4. Quartal 2006 durch ein neues Produkt vollständig abgelöst. Trotz dieses Umstandes muss der Support, bis zum letzten Tag und dem letzten Kunden, in vollem Umfang und mit gleichbleibender Qualität gewährleistet werden.

Produkteigentümer ist das Rechenzentrum. Die Entwickler des Systems wurden in der Zwischenzeit, bei einer Reorganisation des Unternehmens, in eine eigene Gesellschaft ausgegliedert. Die Tochtergesellschaft ist nun Dienstleister und wartet das System im Auftrag der Muttergesellschaft. Das Produkt wurde für den Bankenmarkt und nicht für die Verwendung im eigenen Unternehmen entwickelt.

Das System wird von ungefähr 100 Mitarbeitern gewartet. Die regelmäßige Weiterentwicklung des Produktes führte dazu, dass inzwischen für die Wartung ein vielfaches des Aufwands der Erstentwicklung geleistet wurde.

Insgesamt wurden über 6000 Mannjahre aufgewendet. Dies führte zu einer Größe des System von ungefähr 25 MLOC. Bei der Erstentwicklung betrug die Größe des Kernsystem ungefähr 500 kLOC.

Der Releasezyklus beträgt nach Aussage des Interviewpartners einen Monat.

Für die nähere Zukunft sind einige Anpassungen der Software geplant, darunter fällt auch die sogenannte Neue Baseler Eigenkapitalvereinbarung, kurz Basel II.

Neben der Größe hat dieses System auch eine deutlich größere Benutzergruppe als alle anderen betrachteten Systeme. Insgesamt über 25000 Mitarbeiter von Banken nutzen dieses System.

Neben der Entwicklung, wird den Kunden eine Vollversorgung angeboten: Sie beinhaltet neben der Software, die mögliche Bereitstellung entsprechende Hardware für ein Rechenzentrum, oder die Nutzung eines bestehenden Rechenzentrum selbst, sowie den Support, realisiert durch ein Call Center.

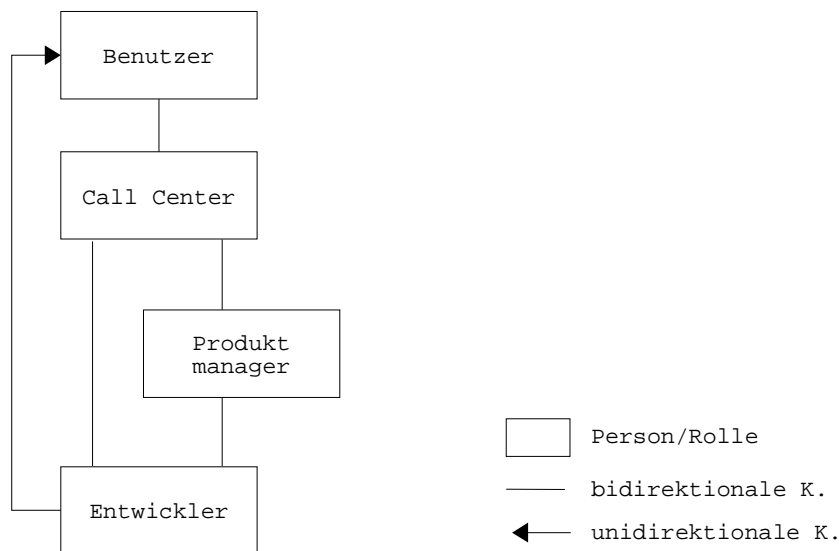


Abbildung 10 Kommunikationswege System E

4.5.2 Die Rollen

Call Center

Aufgrund der großen Zahl von Benutzern ist das Call Center die einzige Anlaufstelle für die Benutzer. Aufgetretene Fehler, Probleme und Wünsche müssen von den Benutzern immer an das Call Center gemeldet werden. Die Mitarbeiter im Call Center sind fachlich geschult und besitzen eine detaillierte Fehlerdatenbank. Falls eintreffende Anfragen, von den Mitarbeitern des Call Centers, nicht gelöst werden können, werden diese an die zuständigen Entwickler weitergeleitet.

Zwischen Call Center und Entwicklern ist auf Grund der Komplexität der Produkte manchmal ein Produktmanager notwendig. Er beurteilt ob es sich bei einer Meldung tatsächlich um einen Fehler handelt und erstellt eine Bewertung und Priorisierung der Anforderung.

In *Abbildung 10* sind die Kommunikationswege zwischen Benutzer und Entwickler dargestellt.

4.5.3 Der Wartungsprozess

Service Level

Alle Anfragen, die an die Entwickler weitergeleitet werden müssen, werden vom Call Center und dem entsprechenden Entwickler/Produktmanager gemeinsam in eines von drei Service Level eingestuft:

- Service Level 1:
Anfrage muss innerhalb von acht Stunden gelöst werden
- Service Level 2:
Anfrage muss innerhalb von drei Tagen gelöst werden
- Service Level 3:
Anfrage soll bis zum nächsten Release gelöst werden

Daneben sind in den Service Level festgelegt:

- Vertragsstrafen bei Nichteinhaltung
- Eskalationsprozesse
- Informationspflichten
- Dokumentation-Anforderungen
- Prozesse

In diesem Projekt sind während der Fehler- oder Problembhebung die Service Level entscheidend. Sie haben bei Konflikten mit Terminen der Weiterentwicklung Vorrang.

Nach einer Fehlermeldung besteht für die Entwickler die Möglichkeit auf das System des Kunden zuzugreifen und eine Analyse durchzuführen. In Ausnahmefällen wird auch direkt mit dem Urheber der Meldung Kontakt aufgenommen.

Die Wartung des Systems ist vollständig in den Entwicklungsprozess eingebunden. Jede Wartungsaktivität folgt demselben Ablauf, wie er auch für die Entwicklung vorgeschrieben ist:

- Analyse des Ist-Zustands
- Erfassung der Anforderungen/Bewertung und Priorisierung
- Erstellung des Entwurf (Erstellung des Pflichtenheftes, Aufwandsschätzung, ...)
- Implementierung
- Test mit Rollout und Pilotierung
- Einsatz

Während eines Wartungsfalls, z.B. des Service Level 1, können die einzelnen Schritte stark komprimiert und Aufgaben, wie die Erstellung von Dokumentation, im Einzelfall auf die Zeit nach der Lösung des Problems verschoben werden.

Nach Konzeption und Implementierung folgt ein mehrstufiger Test der Änderungen. Die Entwickler führen Modul- und Integrationstests durch. Danach folgen Regressionstest der Qualitätssicherung. In der QS sind ein Drittel aller Entwickler des Unternehmens beschäftigt. Die Qualitätssicherung testet jede Änderung und jedes Release und betreut die Pilotphase.

4.5.4 Besonderheiten

Bei diesem Projekt bestehen einige Besonderheiten im Kontext dieser Untersuchung:

- Service Level sind im Regelfall wichtiger als Termine
- Der Wartungsprozess ist in die Entwicklung integriert
- Das Wartungspersonal ist ausgegliedert
- Die Releasetermine sind fix, ggf. muss der Umfang reduziert werden, falls Implementierungen nicht planmäßig fertig gestellt werden können.
- Neben den Kunden, vertreten durch den Bankenverband, tritt auch das Bundesamt für Finanzdienstleistungsaufsicht als Kontrollinstanz auf.
- Es müssen die Grundsätze ordnungsmäßiger DV-gestützter Buchführungssysteme (GoBS), herausgegeben vom Bundesministerium für Finanzen, eingehalten werden.
- Die Wartung deckt neben Fehlerbehebung und Anpassung an gesetzliche Anforderungen, auch proaktive Wartung ab.

5 Auswertung

5.1 Vergleich der Standards

Nach der Vorstellung der drei Vorgehensmodelle im Kapitel 2.3 sollen nun die Gemeinsamkeiten und Unterschiede erörtert werden.

Der IEEE 1219 ist der ausführlichste der drei Standards, er hat auch die Wartung zum alleinigen Thema. Beim ISO/IEC 12207 ist Wartung nur ein Kapitel im Gesamtgebiet der Softwareentwicklung. CMMI beschäftigt sich nicht explizit mit der Software-Wartung, sondern beschreibt Prozessgebiete in der Softwareentwicklung. Diese können teilweise für die Wartung verwendet werden.

Im folgenden Vergleich ist zu jedem Schritt in der Wartung der entsprechende Schritt der verschiedenen Modelle aufgeführt. Allen Vorgehensmodellen gemeinsam ist die Erfassung von Metriken in allen Phasen.

	IEEE 1219	ISO/IEC 12207	CMMI
Initiierung	MR	MR	Request
Klassifizierung	4-stufig	4-stufig	-
Analyse	Analyse	Analyse von Problem und Änderung	Anforderungsmanagement und Anforderungsentwicklung
Entwurf	Entwurf	Implementierung der Änderung	Technische Umsetzung
Implementierung	Implementierung		Verifikation
Systemtest	Regressions-/ Systemtest		Validierung
Abnahmetest	Abnahmetest	Review und Abnahme der Wartung	-
Auslieferung	Auslieferung	Migration	-
Außerbetriebnahme	-	Außerbetriebnahme	-

Tabelle 4 Vergleich der Standards

Alle Zellen der *Tabelle 4* werden im Kapitel 2.3 ausführlich beschrieben.

5.2 Vergleich der Wartungs-Projekte

	Projekt A	Projekt B	Projekt C	Projekt D	Projekt E
Initiierung	Benutzer →Produktbetreuer → Produktakte →„Kehrwoche“	Benutzer →CCC oder Entwickler (Fehler, Problem) Benutzer →IT-Kundenma- nager (Wunsch)	Benutzer →IT-Kontakt- stelle	Benutzer A →Entwickler Benutzer B →Poweruser →Hotline → Entwickler	Benutzer →Call Center
Klassifizierung	Fehler (4 Kategorien) Wünsche nur als Entwicklungs- auftrag oder <5PT	3 Kategorien >150PT Vorstudie	Problemmeldung <5PT Klein-Auftrag <60PT Projekt >60PT	Vorgangsmeldung mit 4 Kategorien	3 Kategorien oder Wunsch
Change Control Board	Aufwand entscheidend; Projekt-Freigabe-Gremium ist letzte Instanz	Aufwand entscheidend; IT-Kundenmanager	IT-Kontaktstelle & Entwickler	Fach-Arbeitskreis (global) DB-, System-Entwickler & Release- manager (lokal)	Entwickler, evtl. mit Pro- duktmanager
Analyse	Produktbetreuer & Entwickler	Entwickler	Sachgebietsleiter (SGL) oder Gruppenleiter (GL)	Entwickler	Phasen werden wie wäh- rend der Entwicklung be- handelt
Entwurf	Entwickler (Fachkonzept)	Entwickler (Fachkonzept)	Entwickler	Entwickler	
Implementierung	Entwickler	Entwickler	Entwickler	Entwickler	
Systemtest	Entwickler (Modultest) Kunde &Betreuer (Systemtest)	Entwickler	Entwickler (Modultest) SGL/ GL (tech. Kon- trolle)	Entwickler (Modultest) unbeteiligte Entwickler (Systemtest) Kunde auf eigener HW (Systemtest)	Entwickler (Modul-/ Sy- temtest) QS (Regressionstest)
Abnahmetest	Produktbetreuer (& Kunde)	IT-Kundenmanager	IT-Kontaktstelle	Kunde	Kunde
Auslieferung	Produktbetreuer	Entwickler	Qualitätssicherung	Vertrieb	Qualitätssicherung
Außerbetrieb- nahme	Noch nicht absehbar	Studie für Nachfolgesystem beauftragt	Studie für Nachfolgesys- tem beauftragt	frühestens 2009 (Ende der Wartungs- verträge)	4. Quartal 2006

Tabelle 5 Vergleich aller Projekte

Alle Zellen der *Tabelle 5* werden im Kapitel 4 ausführlich beschrieben.

Wenn man nun die einzelnen Wartungs-Projekte näher betrachtet und miteinander vergleicht stellt man einige gemeinsame Merkmale fest:

- Zwischen der Benutzergruppe und den Entwicklern befindet sich immer eine Stelle zur Bündelung eingehender Anfragen. Mal ist es eine Hotline oder ein Kundenmanager, mal sind es zusätzlich ausgebildete Mitarbeiter der Kunden in den jeweiligen Abteilung vor Ort. Diese kanalisieren alle Anfragen und geben sie dann an die Entwickler weiter.
- Alle Industriepartner beurteilen eingehende Anfragen und habe dafür zwischen drei und vier Kategorien zur Verfügung.

Die Art der Kategorisierung ist zweigeteilt. Die Projekte A, B und C leiten ihre Kategorien aus dem Aufwand ab. Die Projekte D und E stufen die eingehenden Anfragen nach Wichtigkeit ein.

Alle Projekte haben ein Change Control Board (CCB) eingerichtet. Es existieren jedoch bei den einzelnen Change Control Boards erhebliche Unterschiede:

- Bei Projekt A ist das erste Kriterium für die Beurteilung der Wartungsanforderung der Aufwand der geleistet werden muss um eine Änderung durchzuführen. Schon ab einer Größe von fünf Personentagen wird die Entscheidung ob die Änderung durchgeführt wird in einem Gremium auf Vorstandsebene (PFG) entschieden.
- Bei Projekt B hat der Kundenmanager bis zu einem Aufwand von 50 PT, die Möglichkeit mit der auftraggebenden Abteilung die Modalitäten einer Änderung abzustimmen. Erst darüber muss die Entscheidung an eine Projektkommission im Vorstand weitergegeben werden.
- Bei Projekt C bilden die IT-Kontaktstelle und Vertreter der Entwicklung das Change Control Board, unabhängig von der Höhe des zu erwartenden Aufwands. Nur bei Projekten, dies sind Änderungen mit mehr als 60PT prognostiziertem Aufwand, muss die Steuerungs-/Entscheidungsfunktion an den Lenkungsausschuss abgegeben werden. Die technischen Entscheidungen verbleiben beim Change Control Board.
- Es gibt auch eine Lösung mit zwei Change Control Boards, dies ist in Projekt D zu sehen. Es gibt ein Board auf lokaler Ebene, das jede einzelne Anforderung beurteilt. Dies besteht aus einem Datenbank- und Systementwickler und dem Releasemanager, falls notwendig wird ein Spezialist aus dem entsprechende Bereich zusätzlich konsultiert. Das zweite Change Control Board, Fach-Arbeitskreis genannt, ist für globale Anforderungen an das

System verantwortlich. Es besteht aus Vertretern aller Kunden und den Entwicklern, es trifft sich fünf- bis sechsmal jährlich und erarbeitet Vorschläge für die Richtung der Weiterentwicklung des Basissystems. Diese Vorschläge werden dem einmal jährlich tagenden Projektplanungsgremium unterbreitet und dieses entscheidet über das weitere Vorgehen.

- Beim letzten Projekt E wird das Change Control Board aus den Reihen der Entwickler gebildet, eventuell mit Beteiligung des Produktmanagers.

Zur Frage des verwendeten Entwicklungs- bzw. Wartungsprozesses fällt beim Blick in die Übersicht in *Tabelle 3* auf Seite 25 auf, dass System 4 ohne besonderen Entwicklungs-/Wartungsprozess auskommt. Aufgrund der geringen Anzahl von beteiligten Entwicklern wurde darauf verzichtet einen Prozess zu definieren und zu benutzen. Sehr wohl gibt es ein koordiniertes Vorgehen bei der Wartung und Entwicklung, die Aufgaben wurden den Mitarbeitern fest zugeordnet und nur in Ausnahmefällen wird davon abgewichen.

5.3 Vergleich der Praxis mit der Literatur

Ich möchte nun die Literatur mit dem Vorgehen bei der Wartung in der Praxis vergleichen. In *Tabelle 5* stehen in der ersten Spalte die Ereignisse, Rollen oder Phasen wie man sie aus der Literatur ableiten kann. Es fällt dabei auf, dass alle von der Literatur geforderten Rollen und Aktivitäten in den betrachteten Projekten entsprechende Einträge enthalten. Damit wäre der erste Eindruck des Vergleichs als positiv zu bewerten.

Es gibt aber einen bedeutenden Unterschied zwischen Literatur und der Praxis in den untersuchten Wartungsprojekten. Dies ist die Kategorisierung von eingehenden Wartungsanforderungen. In der Literatur wird nach der Art der Wartung unterschieden und abhängig davon das weitere Vorgehen geplant. Die Industriepartner unterscheiden primär nach Aufwand der Wartung und danach wird die Art der der Wartung festgestellt.

Bei drei Projekten (A, B und C) ist der geschätzte Aufwand das wichtigste Kriterium ob die Änderung durchgeführt wird. In den Projekten D und E werden die Anforderungen nach der Wichtigkeit für den Benutzer klassifiziert. Der Grad der Wichtigkeit bestimmt den Zeitraum bis wann eine gemeldete Anforderung umgesetzt werden muss.

Somit erfüllt keines der Projekte die Forderung der Literatur eingehende Änderungsmeldungen nach Art der Wartung zu unterscheiden und daraus das weitere Vorgehen abzuleiten.

Sehr wohl unterscheiden die Unternehmen bei der Abrechnung nach der Art der Wartung. So werden einerseits Pauschalen für die adaptive und korrektive Wartung vereinbart, als Beispiel wäre dies Projekt D, andererseits verpflichtet sich der Kunde nach Abnahme des Systems jede Art der Wartung zu bezahlen, Projekt C wäre ein Beispiel dafür.

In der Literatur wird für die Initiierung des Wartungsprozesses ein formaler Anstoß gefordert. In den Projekten B und D besteht für die Benutzer die Möglichkeit direkt mit den Entwicklern in Kontakt zu treten. Dies wurde einerseits durch die geringe Anzahl der Entwickler, wie in Projekt B, andererseits durch Kundenwunsch in die Wartungsverträge aufgenommen. Dies führt dazu, dass die jeweiligen Entwickler nun in der Pflicht stehen eine formalisierte Wartungsanforderung erstellen zu müssen. Wenn die Benutzer dieses Dokument selbst erstellen haben sie die Möglichkeit besser über den Verlauf ihrer Anforderung informiert zu bleiben. Denn das Change Control Board ist verpflichtet jede Anforderung zu erfassen und den Urheber über den Status zu informieren. Bei mündlich formulierten Anforderungen können diese im Wartungsalltag vereinzelt verloren gehen.

Die Literatur fordert zu Beginn einer Wartungsaktivität eine umfassenden Analyse und Abschätzung der Folgen für das System. Es sollen die Auswirkungen bei Implementierung und bei Ablehnung ausführlich dokumentiert werden. Dies ist aus Zeitgründen in der Praxis nicht immer im geforderten Umfang möglich. Dies führt oft zu Problemen während der später folgenden Implementierung.

Der ISO/IEC 12207 fordert die Einrichtung eines Verfahrens zum Empfang von Problemberichten. In Projekt A gibt es dazu den Prozess der „Kehrwoche“, dieser wöchentlich wiederkehrende Vorgang beschreibt die Tätigkeit des Entwicklers die Produktakte zu leeren und enthaltenen Anforderungen zu bearbeiten. Dieser Aufgabe wird in regelmäßigen Abständen allen Entwicklern zugeteilt. Damit wird gewährleistet, dass keine Anforderung unbearbeitet bleibt.

Zu Beginn dieser Untersuchung wurde in *Tabelle 1* auf Seite 2 der Umfang bzw. Größe der Systeme verglichen. Einige Interviewpartner waren nicht in der Lage Aussagen über die Größe des System zu machen weil sie nie erfasst wurde. Es kann über die Aussagekraft von Metriken wie es die Kodezeilen sind diskutiert werden, aber wenn schon die Erhebung solch einer relativ einfach zu erfassenden Metrik nicht umgesetzt wird kann bezweifelt werden, dass

komplexere Metriken je Einzug in die Entwicklung bzw. Wartung halten werden. Die Interviewpartner konnten zwar Angaben zu der Anzahl von Modulen und Unterprogrammen machen, aber diese sind wegen fehlender Definition, wie groß den ein Unterprogramm oder Modul sein darf, nicht vergleichbar.

6 Zusammenfassung

In diesem Abschnitt werden die wesentlichen Ergebnisse der Untersuchung nochmals kurz zusammengefasst. Die Untersuchung hat gezeigt, dass bei der Wartung in der Regel am meisten Aufwand geleistet wird. Bis auf das jüngste System haben alle anderen Projekte in der Wartung schon die Aufwände der Erstentwicklung eingestellt oder schon deutlich überschritten. Dies bestätigen auch die Aussagen der Literatur. Alle untersuchten Projekte haben ein erfolgreiches Produkt entwickelt. Alle Produkte befinden sich bereits mehrere Jahre oder sogar Jahrzehnte im produktiven Einsatz. Das kontinuierliche Wachstum der Funktionalität zeigt den Bedarf der Weiterentwicklung der Software. Folglich nimmt der Wartungsanteil des Gesamtaufwands ständig zu.

Nur bei einem der untersuchten Projekte ist ein Ende der Weiterentwicklung der Software in Sicht, da die Software durch ein neues Produkt vollständig ersetzt wird. Bei zwei Systemen sind Studien in Arbeit um die Möglichkeiten von Erweiterung oder den Einsatz eines neuem System zu ergründen. Es gibt immer geplante neue Funktionalität oder noch nicht verwirklichte Kundenwünsche.

Bei der Mehrzahl der untersuchten Projekte werden keine Metriken erhoben. Aufgrund der Tatsache, dass Metriken nicht erhoben werden, können verschiedene Modelle aus der Literatur für die Berechnung des Wartungsaufwands nicht angewendet werden.

Ein Ergebnis dieser Arbeit ist, dass der derzeitige Zustand der Software bei drei Systemen besser ist, als zu Beginn des produktiven Einsatzes, d.h. unmittelbar nach der Erst-Entwicklung, und bei zwei Systeme durch das enorme Wachstum die aktuellen Versionen viel komplexer und schwieriger zu handhaben sind als deren ursprüngliche Systeme. Die Erweiterbarkeit und die Lesbarkeit des Quellcodes konnte auch nur in den drei kleineren System gehalten oder sogar gesteigert werden.

Die Befragung hat auch ergeben, dass die Fluktuation der Mitarbeiter gering ist und somit kein Problem darstellt. Die Mehrzahl der Personen, die bei der Erst-Entwicklung beteiligt waren, sind auch derzeit noch im Unternehmen. Nur bei einem System war keiner der Erst-Entwickler mehr in der Wartung beteiligt.

7 Fazit und Ausblick

Diese Arbeit hat gezeigt, dass sich der Zustand der Software während der Wartung in der Regel nicht verschlechtert. Die Wartung wird genutzt um die Dokumentation zu vervollständigen oder die Wartbarkeit zu erhöhen. Viele Aspekte sind derzeit besser als direkt nach der Erst-Entwicklung. Die Anzahl der Interviewpartner lässt es aber nicht zu, allgemeingültige Rückschlüsse zu ziehen. Auch wenn alle Industriepartner aus dem Finanzsektor sind, können selbst für diesen Bereich schwer allgemein gültige Aussagen gemacht werden.

Die Richtigkeit der Antworten wird bei dieser Untersuchung vorausgesetzt. Es war dem Bearbeiter aber nicht möglich die Korrektheit der Aussagen zu überprüfen.

Das Engagement der beteiligten Unternehmen hat gezeigt, dass in der Industrie Interesse für solche Untersuchungen besteht. Da es hier um die Bestandsaufnahme der Wartung in der Praxis geht, wurde bewusst auf eine Wertung der einzelnen Vorgehensweisen verzichtet. Es sollte der Stand der Wartung in den Unternehmen zum Zeitpunkt der Untersuchung aufgezeigt werden.

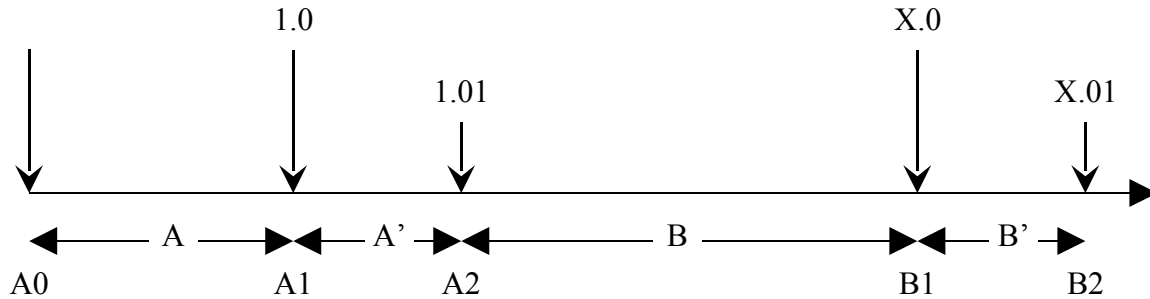
Für die Zukunft könnte sich für die Unternehmen neben der Problematik der Wartung ihrer Systeme auch das Problem der Arbeitskräfte stellen. Noch gibt es ausreichend Fachleute, die die verwendeten Programmiersprachen beherrschen und damit in der Lage wären diese Systeme auch zu warten. Da in der Ausbildung heute nicht mehr so viel Wert auf COBOL oder PL/1 gelegt wird, könnte in fünfzehn bis zwanzig Jahren das Problem entstehen, dass es kein geschultes Personal mehr gibt. Dieses müsste dann durch Schulung in den Unternehmen selbst nachgeholt werden. Ein Unternehmen engagiert sich schon heute in der Ausbildung von Fachkräften um ihre Systeme auch in Zukunft warten zu können.

Literaturverzeichnis

- Canning 1972: Canning R., *The maintenance iceberg*, 10.Oktober 1972 gefunden in [Sneed 1991]
- IEEE 1998: IEEE, *IEEE Standard for Software Maintenance*, 1998
- ISO/IEC 1995: ISO/IEC, *ISO /IEC 12207: Information Technology - Software life cycle processes*, 1995
- Kneuper 2003: Kneuper R.2003. *CMMI: Verbesserung von Softwareprozessen mit Capability Maturity Model Integration*. dpunkt.verlag, Heidelberg, D
- Konrad 2001: Konrad K.2001. *Mündliche und schriftliche Befragung - Ein Lehrbuch*. Verlag Empirische Pädagogik, Landau, D
- Swanson 1980: Lientz B., Swanson E.1980. *Software Maintenance Management*. Addison-Wesley Publishing, Reading Mass.
- Ludewig, Opferkuch 2004: Ludewig J., S. Opferkuch. *Software-Wartung - Eine Taxonomie. Softwaretechnik-Trends, Band 24 Heft 2. Gesellschaft für Informatik e.V. ISSN 0720-8928. Mai 2004. S.35-36. 2004*
- Martin, McClure 1983: Martin J., C. McClure1983. *Software Maintenance: The Problem and Its Solutions*. Prentice Hall, Englewood Cliffs
- Mayer 2004: Mayer H.O.2004. *Interview und schriftliche Befragung*. Oldenbourg, München, D
- Pigoski 1996: Pigoski T.M.1996. *Practical Software Maintenance*. John Wiley & Sons, Inc, New York, USA
- Pressman 2001: Pressman R.S.2001. *Software Engineering - A Practitioner's approach*. McGraw-Hill, Boston, USA
- Sneed 1991: Sneed H.1991. *Software Wartung*. Rudolf Müller, Köln
- ICSM 1996: Sneed H., *Evaluating the Maintenance Process at the Zurich Life Insurance*, 1996 gefunden im ICSM 2003, 22.-26- Sept. 2003 Amsterdam, NL
- Sommerville 1996: Sommerville I.1996. *Software Engineering*. Addison-Wesley Publishing, Harlow, England
- Mayrhauser 1990: von Mayshauser T.E.1990. *Software Engineering - Methods and Management*. Academic Press, Inc., San Diego

Anhang A – Fragebogen

Zeitachse der Softwareentwicklung und -Wartung



Der Zeitraum A ist die Zeit der Erst-Entwicklung. A1 ist der Zeitpunkt zu dem die erste Version (1.0) der Software an den Kunden ausgeliefert wurde. Ab diesem Zeitpunkt beginnt die Wartung. Auch die Weiterentwicklung des Produkts in dieser Phase wird als Wartung bezeichnet. A' ist die erste Wartungsphase nach der Auslieferung.

Während des Zeitraums B werden mehrere Versionen (Releases) der Software an den Kunden ausgeliefert. Die Anzahl der Releases ist nicht von Bedeutung. Der Zeitpunkt B1 stellt die Auslieferung des letzten großen Release dar.

B' entspricht der Phase A'. B' bezieht sich jedoch auf die gewartete Software und nicht wie A' auf die Erst-Entwicklung.

Ein Wartungszyklus entspricht dem Zeitraum zwischen zwei großen Releases, z.B. zwischen 2.0 und 3.0.

Produkt

Allgemeines

Um welches Produkt handelt es sich?

Wann wurde mit der Entwicklung des Produkts begonnen?

Wann wurde die Software zum ersten mal ausgeliefert?

Wann wurde die erste neue Version ausgeliefert?

Wann wurde/wird die letzte Version ausgeliefert?

Wer ist der Kunde?

Wie viel Aufwand wurde geleistet?

- Bei der Erst-Entwicklung
- Seit der Erst-Entwicklung

Wurde die Struktur des Produktes im Laufe der Zeit durch die Wartung übersichtlicher (bessere Werkzeuge, Programmierer)?

Wie viel Prozent der derzeitigen Anforderungen des Kunden sind zur Zeit umgesetzt?

Wo fand die Erst-Entwicklung statt?

- Bei den Kunden vor Ort
- Im eigenen Unternehmen

Sind derzeit größere funktionale Erweiterungen der Software geplant?

Quellcode

Nennen Sie den Umfang des Quellcodes:

- Nach der Erst-Entwicklung
- Derzeit

Wie beurteilen Sie die Stabilität des Quellcodes während der Wartung?

- Nach der Erst-Entwicklung
- Derzeit

Welche Programmiersprachen werden eingesetzt?

- Bei der Erst-Entwicklung
- Derzeit

Werden Metriken gemessen?

Dokumentation

Welche Dokumentation gibt es?

Wird die Codedokumentation automatisch generiert?

Welche Dokumente sind von Bedeutung ?

Sind die Verantwortlichkeiten für die jeweiligen Dokumente geregelt?

Produkteinsatz

Geben Sie die Anzahl der Benutzern an:

- Bei der Erst-Entwicklung
- Derzeit

Welche Verfügbarkeit muss gewährleistet werden?

Produktarten

Wie viele Varianten des Produkts gab/gibt es während der Wartung?

Wie viele Versionen des Produkts gab/gibt es während der Wartung?

Geben Sie die Anzahl der Installationsorte an.

Beurteilen Sie die Konsistenz der Varianten und Versionen untereinander (z.B. Oberflächengestaltung oder Benutzerhandbücher) während der Wartung

Leistungen

Welche Arten des Supports gibt es?

Wie lange wird der Support garantiert?

Kunden

Wird die Kundenzufriedenheit ermittelt (z.B. mit Umfragen)?

Wie wird mit den Kunden kommuniziert?

Wie häufig wird mit den Kunden kommuniziert?

Wie viele SPR werden eingereicht im Verlaufe der Wartung?

Auf welche Art melden die Kunden Fehler?

Wie beurteilen Sie die Genauigkeit der Fehlermeldung der Kunden?

Wartungsmodell

Allgemeines

- Gibt es Vorgaben des Kunden an den Entwicklungs-/Wartungsprozess?
- Wird ein Wartungsmodell verwendet?
- Ist der Entwicklungs-/Wartungsprozess definiert und dokumentiert?
- Gibt es Aktivitäten für die Prozessverbesserung?
- Wie viele Personen sind an der Erst-Entwicklung bzw. Wartung beteiligt?
- Wie sind die Zuständigkeiten geregelt?
- Sind oder waren externe Personen an der Wartung beteiligt?
- Wo findet die Wartung derzeit statt?
- Wird ein CM-System eingesetzt?
- Welche Phasen gibt es bei der Wartung?
- Wie beurteilen Sie das Aufwandsverhältnis zwischen Entwicklung und Wartung?
- Geben Sie an wie viel am System bei der Wartung pro Jahr geändert wird.
- Wer war für den Inhalt der Spezifikation bei der Erst-Entwicklung verantwortlich?
- Wer ist für die Anpassung der Dokumente verantwortlich?
- Welche Wartungsaktivitäten gibt es?
- Gibt es ein Change Control Board?
- Werden Wartungskosten erfasst?
- Werden die Wartungskosten abgerechnet?
- Wie beurteilen Sie die Verteilung des Wartungsaufwands?
 - _____ % adaptive Wartung
 - _____ % korrektive Wartung
 - _____ % perfektive Wartung

Adaptive Wartung

(Anpassung und Weiterentwicklung der Software)

- Wer trägt die Kosten für die adaptive Wartung?
- Wie viel Prozent des Aufwands sind derzeit bei der adaptiven Wartung für die Änderung der Spezifikation nötig?
- Wer ist derzeit für den Inhalt der Spezifikation bei adaptiven Wartung verantwortlich?
- Wie oft gibt es neue Kundenwünsche?
- Warum kommt es zu neuen Kundenwünschen?

Korrektive Wartung

(Behebung von Fehlern in der Software)

Wer trägt die Kosten für die korrektive Wartung?

Wie viele Fehler werden durchschnittlich gefunden?

Wer findet die Fehler?

Wo entstehen Fehler in der Wartung?

Wer ist für die Fehlerbehebung verantwortlich?

Wann werden die Fehler behoben?

Wie viel Aufwand wird durchschnittlich für die Behebung eines Fehlers benötigt?

Perfektive Wartung

(Verbesserung der Eigenschaften, wie z.B. Performance oder Wartbarkeit)

Wie viel % des Systems werden durchschnittlich bei der perfektiven Wartung pro Wartungszyklus geändert?

Wer trägt die Kosten für die perfektive Wartung?

Wer ist derzeit gegebenenfalls für den Inhalt der Spezifikation bei der perfektiven Wartung verantwortlich?

Projektleitung

Planung und Kontrolle

Wird die Planung und die Kontrolle dokumentiert?

Wie detailliert ist die Prozesserfassung

Wird der Fortschritt gemessen?

Gibt es Meilensteine?

Wie viele Versionen waren bereits während der Erst-Entwicklung geplant?

Wie viele Versionen sind derzeit geplant?

Wird der Aufwand für Wartung geschätzt?

Welche Maßnahmen werden bei Planungsabweichungen (Soll-Ist-Abweichungen) ergriffen?

Qualitätssicherung

Gibt es QS-Maßnahmen?

Gibt es eine unabhängige QS-Abteilung?

Wie viele Personen sind ggf. bei der Qualitätssicherung beteiligt?

Werden Tests durchgeführt?

Wartungszyklus

Wie groß ist durchschnittlich der Aufwand eines Wartungszyklus?

Wie lange dauert durchschnittlich ein Wartungszyklus?

Wie viele Wartungszyklen gab es bislang?

Tools

Welche Werkzeuge werden eingesetzt?

Personal

Allgemeines

Welche Form der Projektorganisation wird verwendet?

Wie viel Prozent der Arbeitszeit wenden die Mitarbeiter für Neuentwicklungen auf?

Wie viel Prozent der Arbeitszeit wenden die Mitarbeiter für Wartungsaktivitäten auf?

Wie groß ist der Anteil der Personen bei der Wartung, die bereits bei der Erst-Entwicklung beteiligt waren?

Wie beurteilen Sie den Anteil der Senior- und Junior-Entwickler?

Wie rekrutieren Sie Wartungsingenieure?

Gibt es Fortbildungsmöglichkeiten im Unternehmen?

Begriffslexikon

- Stabilität: Veränderungsrate des Systems, viele Änderungen ergeben instabilen Code
- SPR: Software Problem Report, Fehlermeldung des Kunden oder Benutzers
- Change Control Board: Gremium oder Person, die SPR bewertet
- Verteilung: Es wurde diese Gliederung der Wartung vorgenommen, falls diese Unterteilung nicht auf Ihre zutrifft, wird Ihre beschrieben
- Planung und Kontrolle: Werden Projektpläne dokumentiert und deren Einhaltung kontrolliert und ebenfalls dokumentiert?
- Fortbildungsmöglichkeiten: Wie wird aus einem Absolventen einer Hochschule oder einem Entwickler ein Wartungsfachmann?

Erklärung

Ich versichere, dass ich diese Arbeit selbständig verfasst und nur die angegebenen Hilfsmittel verwendet habe.

Stuttgart, den 27.07.2004

Dražen Pavlaković