

Institut für Softwaretechnologie
Abteilung Software Engineering

Universität Stuttgart
Universitätsstraße 38
D – 70569 Stuttgart

Diplomarbeit Nr. 2414

Modelle zur Wartbarkeit von Software

Christiane Taras

Studiengang:	Softwaretechnik
Prüfer:	Prof. Dr. rer. nat. Jochen Ludewig
Betreuer:	Dipl.-Ing. Rainer Schmidberger
begonnen am:	14.11.2005
beendet am:	16.05.2006
CR-Klassifikation:	D2.9, K6.3

Modelle zur Wartbarkeit von Software

Christiane Taras

Institut für Softwaretechnologie
Abteilung Software Engineering
Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart
christiane@familie-taras.de

Zusammenfassung

Schon seit einiger Zeit besteht Einvernehmen darüber, dass die Wartung existierender Software einen Großteil der Arbeit an Softwaresystemen ausmacht. Der Aufwand, der in Wartung von Software fließt, übersteigt den für die Entwicklung benötigten Aufwand bei weitem. Die Literatur ist sich darüber einig, dass der Wartungsaufwand von anderen Faktoren abhängt, als der Entwicklungsaufwand. Deshalb wurden bereits zahlreiche Modelle zur Abschätzung des Wartungsaufwandes von Software entwickelt. Auch zur Bestimmung der Wartbarkeit von Software, welche als wesentlicher Einflussfaktor in Bezug auf den Wartungsaufwand gesehen wird, sind in den letzten 20 Jahren einige Modelle entstanden. Alle Modelle wurden in kleineren bis mittleren Untersuchungen validiert. Dafür wurden sowohl reale Softwareprojekte aus Unternehmen herangezogen, als auch speziell für die Untersuchungen durchgeführte studentische Projekte. Die in den Modellen verwendeten Faktoren reichen von der weit verbreiteten „Lines Of Code“-Metrik bis hin zu Einschätzungen über die Fähigkeiten des Wartungspersonals und Eigenschaften des Wartungsprozesses. Die Modelle bieten unterschiedliche Möglichkeiten zur Anpassung an ein spezielles Unternehmen oder Projekt. Sie sind immer zumindest für eine der heute gebräuchlichsten Programmiersprachen (C, C++, Fortran, Ada, Java, Cobol) und Anwendungsdomänen validiert. Somit sollten im Grunde für jedes Projekt geeignete Modelle zur Schätzung des Wartungsaufwandes oder zur Bestimmung der Wartbarkeit zu finden sein.

Ich widme diese Arbeit
meinem Großvater Paul Butterich (08.11.1922 – 27.01.2006).

Inhalt

1 Einleitung.....	1
1.1 Ziel dieses Berichtes	1
1.2 Aufgabenstellung	1
1.3 Nutzen dieser Arbeit.....	2
1.4 Methodik der Diplomarbeit	6
1.5 Literaturgrundlage	8
2 Diskussion der für die Arbeit wesentlichen Definitionen.....	11
2.1 Modell	11
2.2 Software	11
2.3 Wartung und Wartbarkeit.....	12
2.4 Wartungsaufwand.....	20
3 Kurzbeschreibung der analysierten Modelle	21
3.1 Adaptive Maintenance Effort Model (AMEffMo)	22
3.2 Effort Estimation Model for Corrective Software Maintenance	23
3.3 FEAST Effort Model	24
3.4 Integrated Measure of Software Maintainability	26
3.5 Maintainability Index (MI).....	27
3.6 Maintainability Model of Object-Oriented Design.....	29
3.7 Maintainability Prediction Model (MainPredMo).....	31
3.8 Model for Estimation and Prediction of adaptive Maintenance Effort of Object-Oriented Systems	32
3.9 Predictive Effort Model for Software Maintenance Releases.....	35
3.10 Software Maintainability Index (SMI).....	36
3.11 Software Maintenance Project Effort Estimation Model (SMPEEM).....	37
4 Fragebogen zur Analyse von Modellen zur Wartbarkeit von Software.....	41
4.1 Analyse einer Veröffentlichung.....	41
4.2 Zugehörige Veröffentlichungen – Modell	45
4.3 Analyse eines Modells	46
4.4 Zugehörige Veröffentlichungen – Metrik	54
4.5 Analyse einer Metrik.....	54
4.6 In Modellen verwendete Metriken	55

1 Einleitung

5 Entwicklung eines Metamodells..... 57

6 Konzeption eines Systems zur Ermittlung geeigneter Modelle 63

6.1 Elektronische Erfassung der Analysedaten..... 63

6.2 Konzept des Abfragesystems..... 64

6.3 Nutzen des Gesamtsystems 66

7 Fazit und Ausblick..... 67

Anhang A - Fragebogen

Literatur

Erklärung

1 Einleitung

1.1 Ziel dieses Berichtes

Dieses Dokument berichtet über die Diplomarbeit zum Thema „Modelle zur Wartbarkeit von Software“, die vom 14.11.2005 bis 16.05.2006 am Institut für Software-technologie an der Universität Stuttgart durchgeführt wurde. Es soll über den Verlauf der Arbeit und die erzielten Ergebnisse informieren.

1.2 Aufgabenstellung

Diese Diplomarbeit wurde auf Grundlage der folgenden Aufgabenstellung durchgeführt:

In der Literatur findet man zahlreiche Untersuchungen zur Bewertung der Wartbarkeit von Softwaresystemen. Eine gewichtige Rolle bei diesen Betrachtungen spielt die Wartbarkeit des jeweiligen Programmcodes. Bekannte Vertreter sind der Maintainability Index (MI) von Welker et al. oder das Modell zur Wartungsaufwandsschätzung von Basili et al. Den Untersuchungen liegt in vielen Fällen ein gemeinsames Muster zugrunde: Es wird ein Bewertungsschema für eine bestimmte Fragestellung entwickelt und dieses Schema wird auf mehr oder weniger viele konkrete Softwaresysteme angewendet. Teilweise dienen diese Softwaresysteme aber auch nur zur Kalibrierung des Bewertungsschemas, die Bewertung selbst findet durch Experten-auffassung statt. Untersucht werden Open-Source-Systeme genauso wie kommerzielle Systeme unterschiedlicher Größe und Programmiersprache. Das Bewertungsschema besteht üblicherweise aus unterschiedlich zusammengestellten und unterschiedlich gewichteten Code- und Prozess-Metriken. Die zugrunde liegenden Fragestellungen sind – neben anderen – Aufwandsschätzungen oder Nutzenbetrachtung für Wiederverwendung.

In dieser Diplomarbeit sollen, im Sinne einer Metastudie, die in der Literatur vorhandenen Bewertungsschemen und Studien verglichen werden. Mindestens die folgenden Aspekte der einzelnen Publikationen sollen untersucht werden:

- zugrunde liegende Fragestellung
- angewendete Methodik (die Methodik soll auch bewertet werden)
- enthaltene Metriken mit Gewichtung
- gegebenenfalls Einschränkungen der Anwendbarkeit (z. B. Programmierparadigma, Programmgröße, usw.)

1 Einleitung

- Umfang der Evaluation oder Kalibrierung
- Alter und Zielsetzung der Untersuchung
- Gebrauchswert der Untersuchung (darin enthalten die Übertragbarkeit der Ergebnisse und eine Bewertung, wie praktikabel das Schema in der Praxis angewendet werden kann)

1.3 Nutzen dieser Arbeit

Die Aufgabenstellung gibt als Ziel dieser Arbeit die Durchführung einer Metastudie über Modelle zur Schätzung des Wartungsaufwandes und Modelle zur Bestimmung der Wartbarkeit vor. Die Durchführung einer solchen Studie ist natürlich nur dann sinnvoll, wenn sie auch einen Nutzen bringt. Dabei stellen sich drei wesentliche Fragen:

1. Warum sollte eine Metastudie durchgeführt werden?
2. Warum sollten Modelle zur Schätzung des Wartungsaufwandes von Software untersucht werden?
3. Warum sollten Modellen zur Bestimmung der Wartbarkeit von Software untersucht werden?

Diese drei Fragen sollen in den folgenden Abschnitten beantwortet werden, um den Nutzen dieser Arbeit deutlich zu machen.

1.3.1 Warum sollte eine Metastudie durchgeführt werden?

Ursprung der vorliegenden Aufgabenstellung war der Gedanke, dass es für die Durchführung von Softwareprojekten nützlich wäre, den Wartungsaufwand und die Wartbarkeit einer Software bestimmen zu können. Daraus entstand die Idee, eine Untersuchung der Abhängigkeiten des Wartungsaufwandes und der Wartbarkeit von Software anhand realer Softwareprojekte durchzuführen, um entsprechende Modelle aufzustellen. Eine kurze Literaturrecherche nach ähnlichen Untersuchungen zeigte jedoch, dass offensichtlich schon einige solcher Modelle entwickelt wurden. Deshalb wurde entschieden zunächst eine Metastudie über Modelle zur Wartbarkeit von Software durchzuführen, um einen Überblick über die schon durchgeführten Untersuchungen auf diesem Gebiet zu erhalten.

1.3.2 Nutzen von Modellen zur Schätzung des Wartungsaufwandes

Wartung spielt eine wesentliche Rolle im täglichen Geschäft der Software-Entwicklung. Immer mehr Software-Systeme sind immer länger im Einsatz und müssen fortlaufend den sich ändernden Anwendungsgebieten angepasst werden. Reine Neuentwicklungen begründen nur noch einen Bruchteil des in Software-Systeme investierten Aufwandes [PolPiaRui03]. Heutzutage ist der wesentliche Kostenfaktor für Software-Unternehmen der Aufwand, der in Korrekturen, Anpassungen und Erweiterungen der bestehenden Systeme fließt. Es ist also unerlässlich Methoden für dessen möglichst genaue Abschätzung zu entwickeln.

Der Nutzen eines konkreten Modells zur Abschätzung des Wartungsaufwandes drückt sich deutlich in seinem Schätzungsziel aus. Folgende Schätzungsziele sind denkbar:

- Ermittlung des Wartungsaufwandes für den restlichen Lebenszyklus
- Ermittlung des Wartungsaufwandes für einen bestimmten Zeitraum (z. B. für die kommende Geschäftsperiode, für ein Jahr, für einen Monat)
- Ermittlung des Aufwandes für eine konkrete Wartungsaufgabe

Alle diese Schätzungen dienen einerseits zur Planung der kommenden Wartungsarbeiten, können aber andererseits auch als Entscheidungshilfen für z. B. folgende Fragestellungen nützlich sein:

- Soll eine bestimmte Software gekauft werden?
- Soll die Wartung für eine bestimmte Software übernommen werden?
- Soll die Wartung für eine bestimmte Software fortgesetzt werden?
- Kann eine konkrete Wartungsaufgabe im vorgegebenen Zeitraum ausgeführt werden?

Verlässliche Modelle zur Schätzung des Wartungsaufwandes von Software können Software-Wartungs-Prozesse also in vielfältiger Weise verbessern.

1.3.3 Nutzen von Modellen zur Bestimmung der Wartbarkeit

Modelle zur Bestimmung der Wartbarkeit dienen im Wesentlichen drei verschiedenen Zwecken – der fortlaufenden Qualitätskontrolle, der Ursachenforschung bei Veränderungen des Wartungsaufwandes und der Einschätzung der Wirksamkeit von Reengineering-Maßnahmen.

Die fortlaufende Qualitätskontrolle ist der wohl am häufigsten genannte Nutzen von Modellen zur Bestimmung der Wartbarkeit. Aus der Definition der Wartbarkeit ist ableitbar, dass eine gut wartbare Software den Wartungsaufwand positiv beeinflusst. Somit bildet die Wartbarkeit einer Software einen Indikator für den späteren Wartungsaufwand. Es ist also erstrebenswert die Wartbarkeit einer Software auf einem gewissen Niveau zu halten oder gar zu verbessern. Dies geschieht aber nicht von allein. Es müssen Maßnahmen ergriffen werden, die dieses Ziel unterstützen. Eine dabei hilfreiche Maßnahme ist die fortlaufende Überprüfung der Wartbarkeit, so dass während der Entwicklung oder Änderung einer Software auf die Erhaltung der Wartbarkeit geachtet wird. Hierbei muss man natürlich bedenken, dass eine Verschlechterung der Wartbarkeit nicht immer vermeidbar ist, da z. B. umfangreiche Ergänzungen von komplexen Funktionalitäten fast zwangsweise zu einer Verkomplizierung der Software und somit zu einer Verschlechterung der Wartbarkeit führen. Bei der Überprüfung der Wartbarkeit müssen also sinnvolle Schwellwerte für gute und schlechte Wartbarkeit angesetzt werden.

Ein weiterer Nutzen von Modellen zur Bestimmung der Wartbarkeit ist die Unterstützung der Ursachenforschung bei Veränderungen des Wartungsaufwandes. Steigt der Wartungsaufwand, wird häufig gefolgert, dass die Software weniger wartbar ist als früher und dass eine Verbesserung der Wartbarkeit der Software den Wartungsaufwand wieder deutlich senken wird. Aufgrund der Vielzahl von Faktoren, die den Wartungsaufwand beeinflussen, erscheint es mir wenig logisch jegliche Veränderung des Wartungsaufwandes auf die Wartbarkeit der Software zu schieben. Zur erfolgreichen Durchführung passender Maßnahmen zur Aufwandssenkung ist es unbedingt notwendig eine objektive Analyse der möglichen Ursachen durchzuführen. Ein Teil davon ist natürlich die Betrachtung der Wartbarkeit. Aber damit ist es nicht getan. Stellt man bei der Analyse fest, dass sich die Wartbarkeit nicht oder nicht entscheidend verändert hat, so hat es auch keinen Sinn eine Reengineering-Maßnahme durchzuführen. Dies würde im besten Fall zu einer Linderung der Symptome führen. Durch eine weitere Verbesserung der Wartbarkeit könnte der Wartungsaufwand tatsächlich sinken. Es löst aber nicht das eigentliche Problem, das vielleicht in einer sinkenden Motivation des Wartungspersonals oder der Verschlechterung eines eingesetzten Werkzeuges besteht.

Um zu entscheiden, ob eine Reengineering-Maßnahme sinnvoll ist, müssen Wartbarkeit und Wartungsaufwand natürlich fortlaufend erfasst werden. Nur so kann man im Falle einer Veränderung des Wartungsaufwandes entscheiden, ob sich auch die Wartbarkeit der Software verändert hat und wenn ja, ob die Veränderung stark genug ist, um eine Ursache für die Veränderung des Wartungsaufwandes zu sein.

Hat man sich für die Durchführung einer Reengineering-Maßnahme entschieden, so zeigt sich der dritte Nutzen von Modellen zur Bestimmung der Wartbarkeit. Mit Hilfe der historischen Daten über die Wartbarkeit und den Wartungsaufwand kann man zunächst feststellen, wie stark die Wartbarkeit erhöht werden muss, um wieder zu einem akzeptablen Wartungsaufwand zu kommen. Nun kann man die während der Reengineering-Maßnahme durchführbaren Änderungen an der Software daraufhin untersuchen, ob sie mit vertretbarem Aufwand eine entsprechende Erhöhung der Wartbarkeit bringen können. Ist dies möglich, kann man nach Ausführung der Maßnahme deren Erfolg durch erneute Bestimmung der Wartbarkeit überprüfen. Erscheint es nicht möglich, die nötige Wartbarkeitserhöhung herbeizuführen, sollte man nach anderen möglichen Maßnahmen zur Senkung des Wartungsaufwandes suchen. Bei dieser Betrachtung sollte der für die Reengineering-Maßnahme benötigte Aufwand nie vergessen werden. Er muss in einem guten Verhältnis zur erwarteten Senkung des Wartungsaufwandes stehen.

Modelle zur Bestimmung der Wartbarkeit sind also ein gutes Mittel, um zu entscheiden, ob eine Reengineering-Maßnahme sinnvoll ist oder nicht. Durch sie wird auch deutlich, dass Reengineering nicht immer das Mittel der Wahl sein sollte, da die Wartbarkeit einer Software nicht endlos erhöht werden kann und auch nicht jede Verbesserung der Wartbarkeit tatsächlich Einfluss auf den Wartungsaufwand hat.

Um Modelle zur Bestimmung der Wartbarkeit erfolgreich einzusetzen ist es natürlich unumgänglich, sie fortlaufend zu verwenden. Nur so erhält man genügend historische Daten, um eine fundierte Entscheidung treffen zu können. Eine einmalige Erhebung der Wartbarkeit einer Software bringt praktisch keine Aussage, da so keine Beziehung zum Wartungsaufwand hergestellt werden kann. In solch einem Fall, kann auch nicht entschieden werden, ob die Wartbarkeit akzeptabel ist oder nicht, da man für diese Entscheidung im Grunde keine allgemeingültigen Grenzwerte festlegen kann. Eine Anwendung, die komplexe Funktionalitäten realisiert, wird auch eine entsprechend komplexe Implementierung aufweisen, da sich nicht jede Funktionalität mit beliebiger Einfachheit implementieren lässt. Von solch einer Anwendung sollte deshalb eine andere Wartbarkeit erwartet werden, als von einer Anwendung mit sehr einfachen Funktionalitäten. Die Grenzwerte für gute und schlechte Wartbarkeit müssen also unter Beachtung der Komplexität einer Anwendung festgelegt werden.

1.4 Methodik der Diplomarbeit

Das erste Drittel der Diplomarbeit war geprägt von einer umfangreichen Literaturrecherche. Es fanden sich schnell viele Veröffentlichungen zu Modellen zur Wartbarkeit von Software. Darunter waren sowohl Veröffentlichungen, die sich direkt mit Untersuchungen zur Schätzung des Wartungsaufwandes oder zur Bestimmung der Wartbarkeit beschäftigten, als auch solche, die Aussagen zu Richtlinien für empirische Untersuchungen im Bereich der Softwarewartung oder zu ähnlichen Dingen enthielten. Bei einer groben Betrachtung der Veröffentlichungen stellte sich schnell heraus, dass es große Unterschiede in der Durchführung der beschriebenen Studien zur Validierung der Modelle gibt. Vor allem die verschiedenen Rahmenbedingungen der einzelnen Untersuchungen machten schnell klar, dass es eher unwahrscheinlich ist ein Modell zu finden, das für jedes Projekt und jede Software anwendbar ist.

Aufgrund der Vielfältigkeit der gefundenen Veröffentlichungen setzte ich mir zunächst das Ziel, einen Weg zu finden, die einzelnen Modelle und die dazugehörigen Veröffentlichungen sinnvoll zu erfassen. Ziel dieser Erfassung sollte es sein, einen Überblick über die Eigenschaften, die nötigen Rahmenbedingungen und die mögliche Verwendbarkeit der Modelle zu geben. Außerdem sollten durch diese Erfassung die Qualität und der Inhalt der Veröffentlichung und der darin beschriebenen Untersuchungen genau überprüft werden können. Dies sollte einerseits sicherstellen, dass nur Ergebnisse von methodisch sinnvollen Untersuchungen weiter betrachtet werden. Andererseits sollte dies helfen zu erkennen, nach welchen Informationen über die Modelle oder Untersuchungen noch recherchiert werden muss. Eine Bewertung der Modelle selbst schloss ich als Ziel der Erfassung frühzeitig aus, da die bis dahin gelesenen Veröffentlichungen gezeigt hatten, dass der Wert der entwickelten Modelle stark davon abhängt, unter welchen Bedingungen sie eingesetzt werden sollen. Zunächst hatte ich die Idee, Modelle zur Bestimmung der Wartbarkeit anhand der erfassten Einflussfaktoren qualitativ zu bewerten. Diese verwarf ich allerdings nach der Beschäftigung mit den Definitionen zur Wartbarkeit von Software (siehe Kapitel 2.3.2). Dadurch ergab sich auch, dass ich die Analyse der Modelle zur Bestimmung der Wartbarkeit nicht grundsätzlich von der Analyse der Modelle zur Schätzung des Wartungsaufwandes trennte, da keine wesentlichen Unterschiede mehr erkennbar waren.

Bei der Arbeit mit den gefundenen Veröffentlichungen machte ich die Erfahrung, dass sich der Inhalt dieser Veröffentlichung am besten mit Hilfe gezielter Fragen erschließen ließ. Deshalb entschied ich mich, für die Erfassung der Modelle und Veröffentlichungen einen umfangreichen Fragebogen zu entwerfen. Dieser ist in Kapitel 4 näher erläutert. Er entstand auf Grundlage von Analyse Kriterien für die Modelle und Untersuchungen, welche ich aus den gelesenen Veröffentlichungen zusammenstellte. Zur Überprüfung der Brauchbarkeit und Verbesserung des Frage-

bogens habe ich exemplarisch elf Modelle und die dazu gefundenen Veröffentlichungen analysiert. Diese sind in Kapitel 3 kurz beschrieben. Anhand dieser elf Modelle habe ich weiterhin ein Metamodell erstellt, welches sehr gut dazu geeignet ist, alle Informationen zu Modellen zur Wartbarkeit von Software strukturiert darzustellen (siehe Kapitel 5).

Nach Analyse der elf Modelle zeigte sich, dass der Fragebogen zwar sehr hilfreich bei der Erfassung der Veröffentlichungen und Modelle ist, es aber recht aufwändig ist, allein anhand der ausgefüllten Fragebögen ein passendes Modell für eine bestimmte Situation zu finden. Da zu erwarten ist, dass außer den elf von mir analysierten Modellen noch viele weitere Modelle entwickelt wurden und werden, entschied ich mich ein System zu konzipieren, dass mit Hilfe einer kurzen Beschreibung der vorliegenden Situation ein passendes Modell zur Schätzung des Wartungsaufwandes oder zur Bestimmung der Wartbarkeit ermittelt. Grundlage für dieses System sollten natürlich die über den Fragebogen erfassten Daten und das Metamodell sein. Das von mir erstellte Konzept für ein solches System ist in Kapitel 6 beschrieben.

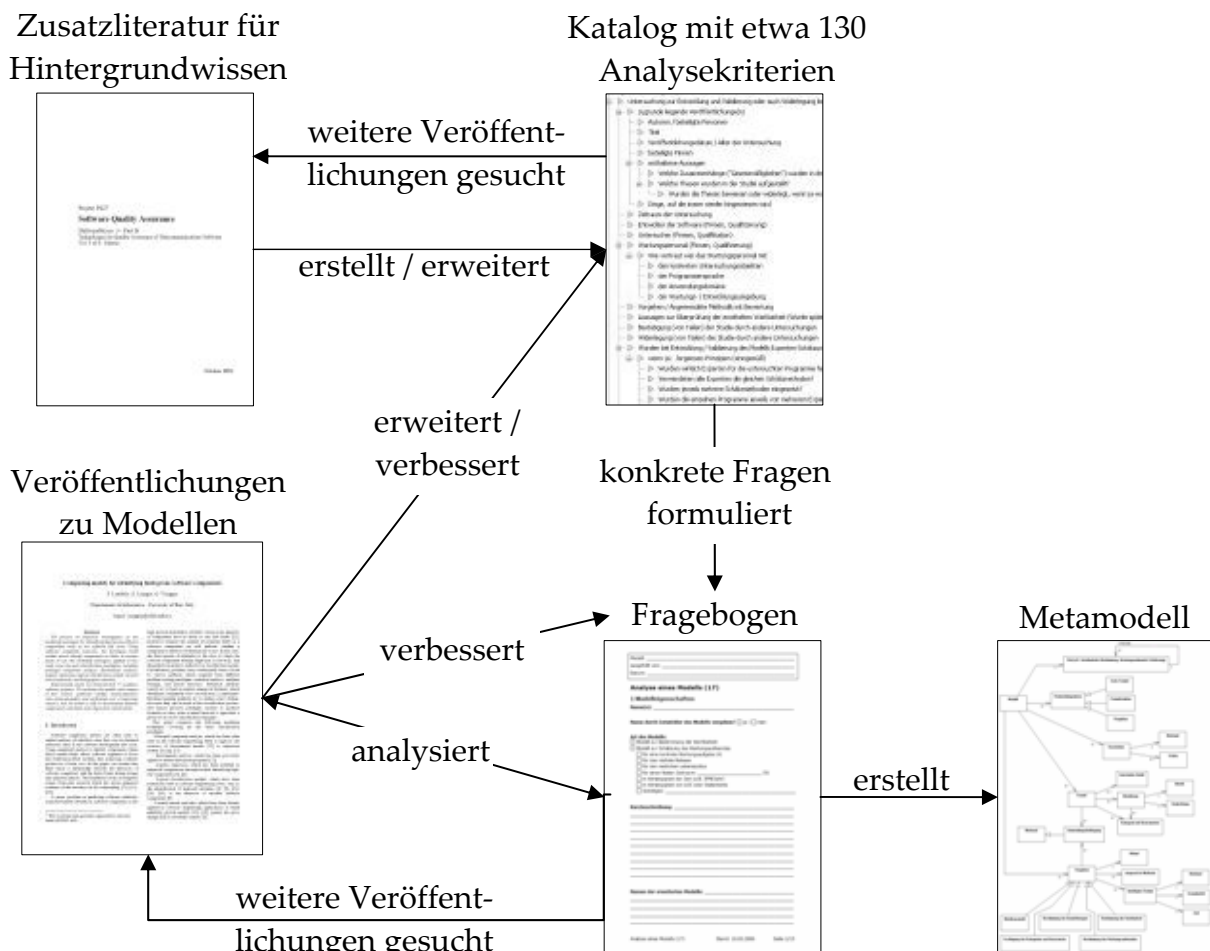


Abbildung 1: Darstellung der Methodik der Diplomarbeit

1.5 Literaturgrundlage

Während der Literaturrecherche zu dieser Arbeit fand sich keine Veröffentlichung, die über eine Metastudie, wie sie in dieser Arbeit durchgeführt werden sollte, berichtet. Da die Literaturrecherche recht umfangreich war, lässt dies vermuten, dass auch noch keine derartige Metastudie durchgeführt wurde oder zumindest keine Veröffentlichung über solch eine Metastudie existiert.

Veröffentlichungen zu Modellen selbst fanden sich schnell sehr viele. Nachdem ich mich entschieden hatte, zunächst einen Fragebogen zur Erfassung der verschiedenen Modelle zu entwickeln, brach ich die Suche nach weiteren Modellen ab und arbeitete mit einer Auswahl von elf Modellen weiter. Die ausgewählten Modelle repräsentieren recht gut den aktuellen Stand der Forschung zu Modellen zur Wartbarkeit von Software. Die Ergebnisse dieser Arbeit basieren im Wesentlichen auf den zu diesen Modellen gefundenen Veröffentlichungen. Leider war es mir bei einigen Modellen trotz intensiver Suche und Kontaktaufnahme mit den an der Validierung der Modelle beteiligten Personen nicht möglich mehr als eine zugehörige Veröffentlichung zu finden. Die folgende Tabelle zeigt eine Übersicht der untersuchten Modelle und der zugehörigen Veröffentlichungen.

Modell	zugehörige Veröffentlichungen
1. Adaptive Maintenance Effort Model (AMEffMo)	[HayPatZha04]
2. Effort Estimation Model for Corrective Software Maintenance	[LucPomSte02]
3. FEAST Effort Model	[RamLeh00], [Leh01]
4. Integrated Measure of Software Maintainability	[AggSinChh02], [AggSinCha05]
5. Maintainability Index (MI)	[ColAshLow94], [WelOma95], [WelOmaAtk97], [Lis01], [Wel01], [VanSciSpr02]
6. Maintainability Model of Object-Oriented Design	[KieJinMue04]
7. Maintainability Prediction Model (MainPredMo)	[HayZha05]

Modell	zugehörige Veröffentlichungen
8. Model for Estimation and Prediction of adaptive Maintenance Effort of Object-Oriented Systems	[FioNesPer98a], [FioNesPer98b], [FioNesSto99], [Fio01], [FioNes01]
9. Predictive Effort Model for Software Maintenance Releases	[BasRom88], [BasBriCon96]
10. Software Maintainability Index (SMI)	[TahGreKon99], [MutPonKon00], [TahKonMy103]
11. Software Maintenance Project Effort Estimation Model (SMPEEM)	[AhnSuhKim03]

Tabelle 1: analysierte Modelle mit zugehörigen Veröffentlichungen

Neben den in Tabelle 1 genannten Veröffentlichungen habe ich für meine Arbeit weitere Veröffentlichungen herangezogen, die sich allgemein mit den Themen Wartung, Wartbarkeit, empirische Untersuchungen oder Metriken beschäftigen. Auf diese ist an entsprechenden Stellen verwiesen.

2 Diskussion der für die Arbeit wesentlichen Definitionen

Um eine sinnvolle Literaturrecherche zu Modellen zur Wartbarkeit von Software durchführen zu können, habe ich mich zunächst mit den Definitionen der für diese Arbeit wesentlichen Begriffe auseinandergesetzt. Dies sind Modell, Software und Wartbarkeit (aus dem Arbeitstitel abgeleitet), sowie Wartung und Wartungsaufwand (aus der Aufgabenstellung abgeleitet). Das folgende Kapitel schildert die Ergebnisse meiner Auseinandersetzung mit diesen Begriffen.

2.1 Modell

Ein Modell ist ein vereinfachendes Abbild der Realität. In dieser Arbeit werden Modelle betrachtet, die Software auf eine Auswahl von Einflussfaktoren der Wartbarkeit oder des Wartungsaufwandes abbilden, um aus der Bestimmung der Werte dieser Einflussfaktoren die Wartbarkeit der Software oder den Wartungsaufwand für die Software ermitteln zu können.

2.2 Software

Dieser Arbeit liegt durchgängig die Definition von Software nach dem IEEE-Standard 610.12-1990 [IEEE90] zugrunde. Diese lautet übersetzt:

Software:

Programme, Abläufe, eventuell zugehörige Dokumentation und Daten, die sich auf den Betrieb eines Rechnersystems beziehen.

Diese Definition konnte ich problemlos auf alle von mir gelesenen Veröffentlichungen anwenden. In keiner der Veröffentlichungen wurde explizit eine andere Definition angegeben.

2.3 Wartung und Wartbarkeit

Bei der Suche nach den Definitionen von Wartung und Wartbarkeit, wurde schnell klar, dass darüber keine Einigkeit in der Literatur besteht. Besonders bei der Bestimmung der Faktoren, die Einfluss auf die Wartbarkeit haben, gehen die Meinungen auseinander. Im folgenden Abschnitt habe ich die Ergebnisse der Auseinandersetzung mit diesen Definitionen ausführlich festgehalten, da dieser Arbeitsabschnitt die Ergebnisse dieser Arbeit wesentlich beeinflusste.

Für das Verständnis der weiteren Arbeitsergebnisse sind im Grunde nur die Abschnitte 2.3.1.4 und 2.3.2.3 von Bedeutung, welche die Schlussfolgerungen aus den Diskussionen der Definitionen aufzeigen. Die Diskussionen selbst können übersprungen werden.

2.3.1 Wartung

2.3.1.1 Wartung nach IEEE

Die wohl bekannteste Definition der Software-Wartung stammt aus dem IEEE-Standard 610.12 von 1990 [IEEE90]. Sie lautet übersetzt:

Wartung:

Prozess der Veränderung eines Software-Systems oder einer Software-Komponente nach der Auslieferung zur Korrektur von Fehlern, Verbesserung der Leistungsfähigkeit oder anderer Merkmale, oder zur Anpassung an ein verändertes Umfeld.

Ein wesentlicher Punkt dieser Definition, der immer wieder kritisiert wird, ist die Beschränkung der Wartung auf Tätigkeiten „nach der Auslieferung“ der Software (siehe z. B. [Kaj03]). Betrachtet man die modernen Methoden der Software-Entwicklung, scheint die Abgrenzung der Wartung von der Entwicklung auf diese Weise nicht sinnvoll.

Software wird immer häufiger mit Hilfe von iterativen Modellen erstellt, um die sich mit dem System entwickelnden Kundenanforderungen befriedigen zu können. Dadurch findet die eigentliche Entwicklung des Systems häufig erst nach einer ersten, meist eher prototypischen Auslieferung eines Systems statt. Natürlich sind die wesentlichen Inhalte der einzelnen Iterationen im Entwicklungsplan bereits vorgesehen, aber es lässt sich kaum vermeiden, dass bereits im Anfangsstadium der Verwendung der Software Design- oder Implementierungs-Fehler auffallen, die so schnell wie möglich behoben werden müssen. Die Methoden, die dabei zum Einsatz kommen entsprechen denen, die auch nach Abschluss des Entwicklungsprojektes zur Fehlerbehebung eingesetzt werden. Entwicklung und Wartung des Systems finden somit

gleichzeitig statt. Zu Gunsten der obigen Definition könnte man behaupten, dass im Grunde alles, was nach der ersten Auslieferung geschieht, Wartung am bestehenden System ist. Allerdings würde man dann missachten, dass nach der ersten Auslieferung meist noch große Teile der Software vollständig neu entwickelt werden, ohne auf bestehenden Teilen aufzubauen.

Und auch ohne eine „erste Auslieferung“ zeigt sich die Problematik der obigen Definition. Bei der Verwendung eines einfachen Phasenmodells ist die Definition des Auslieferungszeitpunktes eindeutig. Es gibt genau eine Auslieferung und zwar nach Abschluss der Entwicklung. Aber natürlich gibt es vor der Auslieferung diverse Testphasen, in denen zum Beispiel Implementierungsfehler oder auch Ungenauigkeiten in der Spezifikation gefunden werden. In solchen Fällen muss man Korrekturmaßnahmen genauso planen und ausführen, wie bei Fehlern, die erst nach der Auslieferung erkannt werden. Es wäre nicht sinnvoll dabei andere Methoden anzuwenden. Im Grunde findet also, zumindest nach den Testphasen, schon Wartung am System statt, obwohl es offensichtlich noch nicht ausgeliefert wurde.

Die Anwendbarkeit der Wartungs-Definition nach dem IEEE Standard 610.12 ist somit höchst fragwürdig. Die Beschränkung der Wartung auf die Zeit „nach der Auslieferung“ passt mit Sicherheit nicht zu den heute angewendeten Entwicklungsmethoden. Dies ist aber im Grunde der einzige Punkt, durch den die Wartung in dieser Definition von der Entwicklung abgegrenzt wird. Der Versuch, die Wartung zusätzlich über die dabei ausgeführten Tätigkeiten abzugrenzen, scheitert an der Formulierung „Verbesserung ... anderer Merkmale“. Je nach Definition der Merkmale einer Software lässt dies Platz für Tätigkeiten jeglicher Art.

2.3.1.2 Wartung nach ISO/IEC

Bei der Analyse der Definitionen zur Wartung in den ISO/IEC-Standards 12207 [ISO95] und 14764 [ISO99] fällt schnell ein gewisser Widerspruch auf.

Im ISO/IEC-Standard 12207 wird die Software-Wartung als ein wesentlicher Prozess im Software-Lebenszyklus definiert, der Aktivitäten und Aufgaben des Wartungs-Personals enthält. Dem Wartungs-Personal werden explizit Aktivitäten und Aufgaben zugeschrieben, die in Phasen vor Auslieferung eines Produktes angesiedelt sind. Es wird explizit verlangt, dass das Wartungs-Personal in den Entwicklungsphasen miteinbezogen werden soll. Im Gegensatz dazu definiert der ISO/IEC-Standard 14764 vier Arten von Wartung – adaptiv, korrektiv, perfektionierend und präventiv – analog zum IEEE-Standard als Änderungen an einem Software-Produkt nach dessen Auslieferung. Weitere Arten der Wartung werden nicht definiert.

Die Definition der Wartung nach den ISO/IEC-Standards 12207 und 14764 ist also in sich nicht schlüssig und somit nicht wirklich anwendbar.

2.3.1.3 Wartung nach Ludewig und Opferkuch

Eine weitere Definition der Software-Wartung wurde 2004 von Jochen Ludewig und Stefan Opferkuch erarbeitet. In [LudOpf04] wird Software-Wartung wie folgt definiert:

Software-Wartung:

jede Arbeit, die nicht von Beginn der Entwicklung an geplant war oder hätte geplant werden können und die unmittelbare Auswirkungen auf den Benutzer der Software hat.

Die Wartung ist hier also nicht durch einen bestimmten Zeitpunkt oder durch die Durchführung bestimmter Tätigkeiten von der Entwicklung abgegrenzt, sondern durch die Planbarkeit der Tätigkeiten. Auffällig ist bei dieser Definition die Forderung nach unmittelbaren Auswirkungen der Wartungsarbeiten auf den Benutzer. Dies schließt jegliche präventive Arbeit von der Wartung aus. Dadurch erscheint diese Wartungsdefinition zunächst als unzureichend. Betrachtet man die Definition allerdings in dem in [LudOpf04] erläuterten Kontext, so wird die Abgrenzung sinnvoll. Wie in der folgenden Abbildung dargestellt ist, wird die Software-Wartung hier als ein Teil der „Software-Pflege“ definiert. Der zweite Teil der „Software-Pflege“ ist das „Software-Re-Engineering“

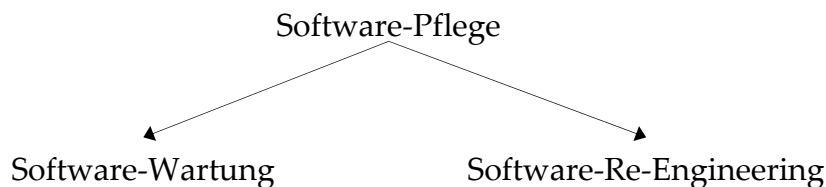


Abbildung 2: Software-Pflege nach Ludewig und Opferkuch

[LudOpf04] definiert weiter:

Software-Re-Engineering:

Änderungen an einer Software, die ausschließlich einer Verbesserung hinsichtlich von Wartungsqualitäten dienen.

Wartungsqualitäten:

Eigenschaften der Software, die direkt keinen Einfluss auf die Qualität aus Sicht eines Anwenders haben, sondern nur die Entwickler und Wartungsingenieure betreffen.

Die Trennung von Software-Wartung und Software-Re-Engineering ist durchaus sinnvoll, da Anlass und Methoden des Re-Engineerings sich deutlich von denen der Wartung unterscheiden.

Diese Definition der Software-Wartung ist meiner Meinung nach deutlich besser als die zuvor genannten Definitionen und sollte deshalb weiter verbreitet werden. Um eine breite Akzeptanz zu erreichen, ist es allerdings unerlässlich den definierten Kontext immer mitzuerläutern, da zum Verständnis der Definition bei vielen Menschen eine Neuordnung der Vorstellungen zu Software-Wartung nötig ist. Weiterhin sollte bei der Definition des Re-Engineerings darauf hingewiesen werden, dass „Änderungen an einer Software“ auch die Erstellung zusätzlicher Dokumente umfasst, um den Bereich des Reverse-Engineerings deutlich mit einzubeziehen.

2.3.1.4 Bedeutung der Definitionen für diese Arbeit

Wie die oben genannten Beispiele zeigen, besteht nicht unbedingt Einigkeit über die Definition von Software-Wartung. Dies ist wohl ein Grund dafür, dass in vielen von mir untersuchten Veröffentlichungen keine Definition für Wartung benannt wird, auf der die beschriebene Untersuchung beruht. Aus verschiedenen Formulierungen lässt sich häufig schließen, dass die Wartungs-Definition nach IEEE zugrunde gelegt wurde.

Für die Bewertung der Untersuchungen ist es allerdings auch nur von geringem Belang, welche Definition z. B. bei der Erfassung des Wartungsaufwandes verwendet wurde. Aufgrund der wenigen, undetaillierten Daten über den erfassten Aufwand, die in den Veröffentlichungen bereitgestellt werden, kann man meist nicht entscheiden, ob er tatsächlich zu der gewählten Definition passt oder nicht. Es ist auch nicht sinnvoll zu fordern, dass alle Aufwandsdaten, die für eine Untersuchung zur Verfügung stehen, detailliert wiedergegeben werden, da die Untersuchungen meist in Zusammenarbeit mit Unternehmen stattfinden und diese wahrscheinlich keine solchen Veröffentlichungen zulassen würden.

Man muss sich also darauf verlassen, dass die erfassten Aufwände tatsächlich zur gewählten Definition passen. Für eine Bewertung der Untersuchungen ist lediglich als positiv bzw. negativ zu werten, ob eine Definition explizit angegeben wurde oder nicht. Weiterhin sollten bei der Suche nach einem passenden Modell für ein bestimmtes Projekt solche Modelle bevorzugt werden, deren zugrunde liegende Wartungsdefinition zum Projekt passt, sofern eine Definition angegeben oder erkennbar ist.

2.3.2 Wartbarkeit

2.3.2.1 Definitionen der Wartbarkeit

Die Definition der Wartbarkeit erweist sich als äußerst schwierig. Zunächst existieren wie schon bei der Wartung verschiedene Definitionen:

Wartbarkeit nach IEEE 610.12-1990 (übersetzt):

Leichtigkeit, mit der ein Software-System oder eine Software-Komponente verändert werden kann, um Fehler zu korrigieren, die Leistungsfähigkeit oder andere Merkmale zu verbessern oder an ein verändertes Umfeld anzupassen. [IEEE90]

Wartbarkeit nach ISO/IEC-Standard 9126 (übersetzt):

Tauglichkeit des Software-Produktes, geändert zu werden. Änderungen können Korrekturen oder Verbesserungen der Software umfassen, sowie Anpassungen der Software an Änderungen der Umgebung, der Anforderungen oder der funktionalen Spezifikation. [ISO01]

Wartungsqualität nach Ludewig und Opferkuch:

Eigenschaften der Software, die direkt keinen Einfluss auf die Qualität aus Sicht eines Anwenders haben, sondern nur die Entwickler und Wartungsingenieure betreffen. [LudOpf04]

Das Wort „Wartungsqualität“ kann meiner Meinung nach mit dem Wort „Wartbarkeit“ gleichgesetzt werden.

Neben den unterschiedlichen Definitionen der Wartbarkeit existieren auch sehr unterschiedliche Auffassungen darüber, welche Faktoren Einfluss auf die Wartbarkeit haben.

2.3.2.2 Bestandteile der Wartbarkeit

Die Definitionen nach ISO/IEC-Standard 9126 und nach Ludewig und Opferkuch bezeichnen Wartbarkeit eindeutig als eine Eigenschaft der Software. Zur Bestimmung der Wartbarkeit einer Software dürfte demnach allein die Software und nicht der Entwicklungsprozess oder die Fähigkeiten der Mitarbeiter betrachtet werden.

Die Definition nach IEEE ist in dieser Hinsicht weniger konkret. Die „Leichtigkeit“ einer Software-Änderung kann im Grunde von sehr vielen Dingen abhängen – von einer tatsächlichen Eigenschaft der Software wie z. B. der Größe bis hin zur aktuellen Befindlichkeit eines Mitarbeiters. Die Wartbarkeit könnte somit an einem Tag sehr gut sein und am nächsten sehr schlecht, nur weil der mit der Wartung betraute Mitarbeiter in der Nacht dazwischen schlecht geschlafen hat. Es ist offensichtlich,

dass das Wort „Leichtigkeit“ an dieser Stelle nicht präzise genug ist. Die Abhängigkeit der Wartbarkeit muss deutlich eingeschränkt werden.

Die Begrenzung der Abhängigkeiten der Wartbarkeit auf die Software allein scheint allerdings auch nicht angemessen zu sein. Unter Beachtung der Definition für Software nach dem IEEE-Standard 610.12-1990 [IEEE90] dürfte die Wartbarkeit nach ISO/IEC-Standard 9126 und nach Ludewig und Opferkuch somit z. B. nicht von einem Werkzeug, mit dem die Software bearbeitet wird, abhängen. Es erscheint allerdings logisch, dass ein Werkzeug Einfluss auf die Wartbarkeit der Software hat. Deutlich wird dies z. B., wenn das einzige für die Wartung zur Verfügung stehende Werkzeug ein einfacher Texteditor ist und UML-Diagramme gewartet werden sollen. Unter diesen Umständen ist keines der Diagramme wartbar, egal welche Unterschiede bezüglich der Wartbarkeit der Diagramme bei der Betrachtung in einem UML-Editor festgestellt werden könnten. Die zur Verfügung stehenden Werkzeuge beeinflussen die Wartbarkeit einer Software also deutlich und müssen somit bei der Bestimmung der Wartbarkeit mit betrachtet werden.

Die Frage ist also, wo die Grenze zwischen Faktoren, die die Wartbarkeit beeinflussen und solchen, die keinen Einfluss haben, zu ziehen ist. Zur Beantwortung dieser Frage ist es sinnvoll, den relativ schwer zu erfassenden Begriff „Wartbarkeit“ in einfacher erfassbare Begriffe zu untergliedern. Gelingt dies, lassen sich leicht Abhängigkeiten erkennen und geeignete Messverfahren zur Bestimmung der Wartbarkeit finden.

Zur Aufgliederung der Wartbarkeit wurden bereits verschiedene Modelle entwickelt. Bei der Betrachtung dieser Modelle zeigt sich schnell, dass auch hier, wie bei den Definitionen zu Wartung und Wartbarkeit, keine Einigkeit besteht.

Im ISO/IEC-Standard 9126 [ISO01] wird die Wartbarkeit folgendermaßen zergliedert:

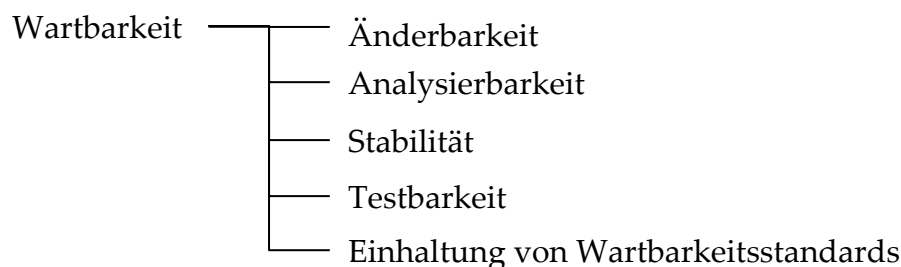


Abbildung 3: Aufgliederung der Wartbarkeit nach ISO/IEC-Standard 9126

Auffällig ist, dass hier die Portabilität kein Teil der Wartbarkeit ist, obwohl die Definition der Wartbarkeit nach dem ISO/IEC-Standard 9126 die Anpassung einer Software an Änderungen in der Umgebung mit einbezieht. Portabilität wird als eigenständige Qualität der Software definiert und wie in Abbildung 4 gezeigt zergliedert.

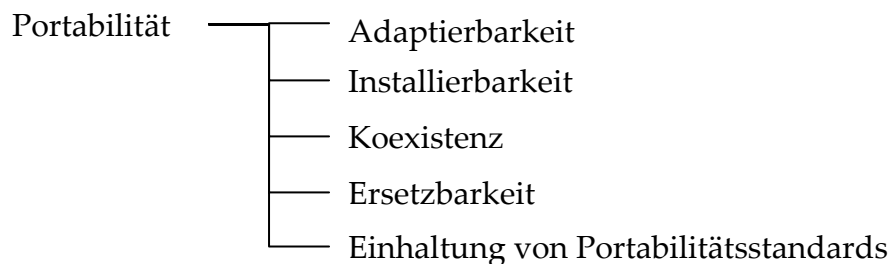


Abbildung 4: Aufgliederung der Portabilität nach ISO/IEC-Standard 9126

Im Sinne der Wartbarkeitsdefinition sollte die Portabilität der Wartbarkeit untergeordnet werden. Leider erfolgt im ISO/IEC-Standard keine weitere Aufgliederung, sodass die Erfassbarkeit der Wartbarkeit nicht wesentlich erhöht wird, da immer noch mit relativ abstrakten Begriffen gearbeitet wird.

Eine stärkere Aufgliederung erfolgt im Software-Qualitäts-Modell von Barry W. Boehm [BoeBroKas78]. Darin wird die Wartbarkeit folgendermaßen zergliedert:

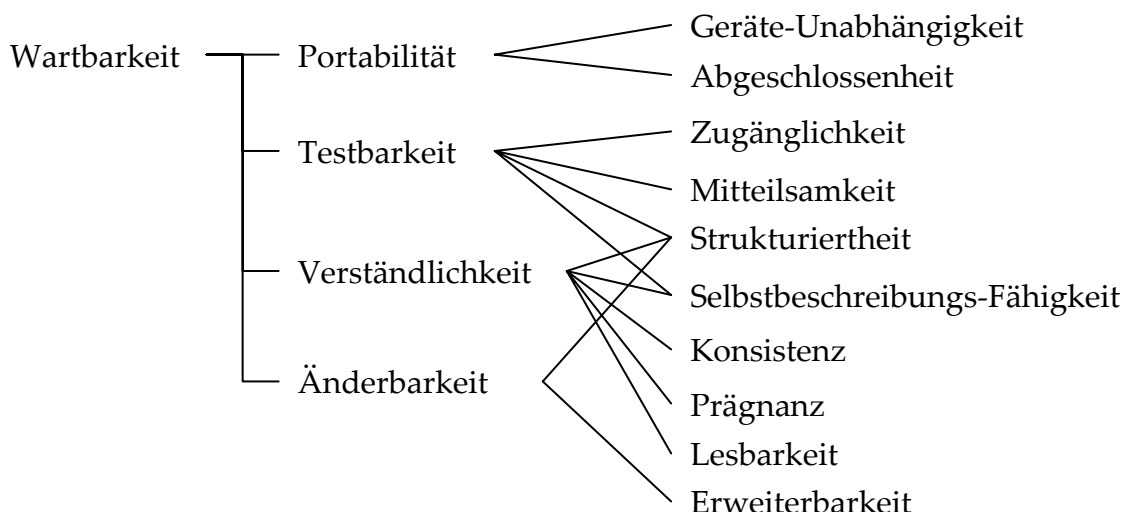


Abbildung 5: Aufgliederung der Wartbarkeit nach Barry W. Boehm

Hier ist deutlich zu erkennen, dass eine detailliertere Aufgliederung wesentlich zum bessern Verständnis des Begriffes Wartbarkeit beiträgt. Aus dem Aspekt „Selbstbeschreibungs-Fähigkeit“ lässt sich z. B. leicht ableiten, dass die Wartbarkeit auch von Kommentaren im Code und der Wahl von Bezeichnern abhängt. Eine detaillierte Aufgliederung hilft somit bei der Suche nach geeigneten Metriken zur Bestimmung der Wartbarkeit. Weiterhin kann sie als gemeinsame Gesprächs- und Forschungsgrundlage dienen. Je detaillierter ein Begriff beschrieben wird, desto weniger Platz wird für unterschiedliche Interpretationen gelassen. Es ist also aus verschiedenen Gründen erstrebenswert eine möglichst detaillierte Aufgliederung des Begriffes Wartbarkeit zu entwickeln.

Die Modelle nach Boehm und ISO/IEC-Standard 9126 sind natürlich nicht die einzigen dieser Art. Ein weiterer Vertreter ist z. B. die „Quality Measure Taxonomy“ des Software Engineering Institute (SEI) der Carnegie Mellon Universität [ForGroRos97]:

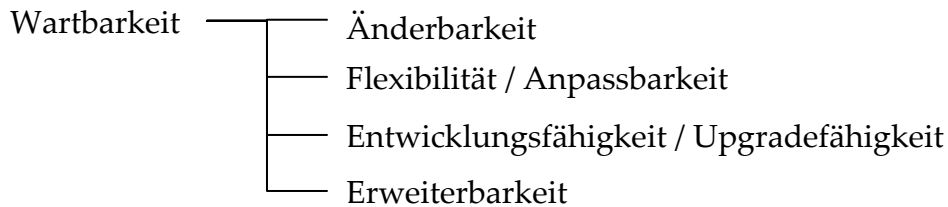


Abbildung 6: Aufgliederung der Wartbarkeit entsprechend SEI

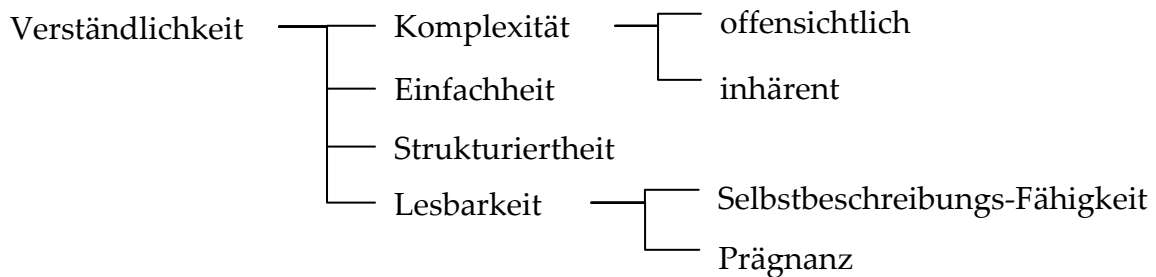


Abbildung 7: Aufgliederung der Verständlichkeit entsprechend SEI

Hier wird die Verständlichkeit von der Wartbarkeit getrennt. Dies erscheint mir allerdings nicht sinnvoll, da die Verständlichkeit auch nicht mit anderen Gebieten als der Wartung in Verbindung gebracht wird und neben der Verständlichkeit nur noch die Wartbarkeit der Wartung untergeordnet ist.

Bei der Aufgliederung der Wartbarkeit wird schnell deutlich, dass für unterschiedliche Bestandteile der Software unterschiedliche Aspekte betrachtet werden müssen. In [Lud98] wird z. B. die Testbarkeit in Boehms Software-Qualitäts-Modell durch den Punkt Prüfbarkeit ersetzt, der wiederum in Lokalität, Testbarkeit und Spezifikations-Vollständigkeit aufgegliedert wird. Diese Aufgliederung ist mit Sicherheit sinnvoll, wenn man die Wartbarkeit des Quellcodes betrachtet. Möchte man allerdings die Wartbarkeit der Spezifikation selbst ermitteln, so ist die Spezifikations-Vollständigkeit sicher nicht relevant.

2.3.2.3 Schlussfolgerung

Zu Beginn meiner Arbeit ging ich davon aus, dass bei den Modellen zur Bestimmung der Wartbarkeit eine qualitative Bewertung der Modelle selbst gut möglich wäre. Als Grundlage sollten einerseits die im Modell erfassten und ausgeschlossenen Einflussfaktoren der Wartbarkeit dienen. Andererseits wollte ich die zur Ermittlung der Werte der erfassten Einflussfaktoren herangezogenen Metriken auf ihre Eignung hin untersuchen.

Die Beschäftigung mit der Definition der Wartbarkeit und ihrer Einflussfaktoren zeigte allerdings, dass keine gefestigte Grundlage für eine solche Bewertung besteht. Da in der Literatur keine Einigkeit über die Definition der Wartbarkeit oder ihre Einflussfaktoren besteht, wäre eine Bewertung der Modelle anhand der Einflussfaktoren reine Willkür. Deshalb konnte ich im Rahmen dieser Arbeit keine solche Bewertung der Modelle zur Bestimmung der Wartbarkeit vornehmen.

2.4 Wartungsaufwand

Für den Wartungsaufwand wird zwar in keiner der von mir untersuchten Veröffentlichungen explizit eine Definition genannt, es scheint aber Einigkeit über die Bedeutung des Wortes zu bestehen. Folgende Definition kann problemlos auf alle von mir gelesenen Veröffentlichungen angewendet werden:

Wartungsaufwand:

Aufwand (in Personen-Zeit-Einheiten), der für die Wartung eines Software-Systems oder einer Software-Komponente benötigt wird.

Der Wartungsaufwand hängt von einer Vielzahl von Faktoren ab, die mehr oder weniger schwer erfassbar sind. Zu diesen Faktoren zählen z. B. Fähigkeiten des Personals, Eigenschaften des Wartungsprozesses, die Zusammensetzung der Wartungsteams, die Häufigkeit von Personalwechseln, die Motivation des Personals und die Wartbarkeit der zu wartenden Software. Dies muss bei der Verwendung von Modellen zur Schätzung des Wartungsaufwandes beachtet werden.

3 Kurzbeschreibung der analysierten Modelle

In diesem Kapitel sind kurz alle elf Modelle beschrieben, an denen ich den Fragebogen und das Metamodell zur Erfassung von Modelleigenschaften entwickelt habe.

Die folgende Tabelle zeigt noch einmal eine Übersicht über die analysierten Modelle und ihre zugehörigen Veröffentlichungen.

Modell	zugehörige Veröffentlichungen
1. Adaptive Maintenance Effort Model (AMEffMo)	[HayPatZha04]
2. Effort Estimation Model for Corrective Software Maintenance	[LucPomSte02]
3. FEAST Effort Model	[RamLeh00], [Leh01]
4. Integrated Measure of Software Maintainability	[AggSinChh02], [AggSinCha05]
5. Maintainability Index (MI)	[ColAshLow94], [WelOma95], [WelOmaAtk97], [Lis01], [Wel01], [VanSciSpr02]
6. Maintainability Model of Object-Oriented Design	[KieJinMue04]
7. Maintainability Prediction Model (MainPredMo)	[HayZha05]
8. Model for Estimation and Prediction of adaptive Maintenance Effort of Object-Oriented Systems	[FioNesPer98a], [FioNesPer98b], [FioNesSto99], [Fio01], [FioNes01]
9. Predictive Effort Model for Software Maintenance Releases	[BasRom88], [BasBriCon96]
10. Software Maintainability Index (SMI)	[TahGreKon99], [MutPonKon00], [TahKonMy103]

Modell	zugehörige Veröffentlichungen
11. Software Maintenance Project Effort Estimation Model (SMPEEM)	[AhnSuhKim03]

Tabelle 2: analysierte Modelle mit zugehörigen Veröffentlichungen

3.1 Adaptive Maintenance Effort Model (AMEffMo)

Das „Adaptive Maintenance Effort Model“ (AMEffMo) wurde zwischen 2001 und 2004 von Jane Huffman Hayes, Sandip C. Patel und Liming Zhao entwickelt. Zur Analyse dieses Modells wurde mir von Liming Zhao eine ungekürzte, unveröffentlichte Fassung der einzigen Veröffentlichung zu diesem Modell zur Verfügung gestellt. Diese enthält einige Informationen, die die Analyse des Modells unterstützen und in der gekürzten, veröffentlichten Fassung ([HayPatZha04]) nicht enthalten sind.

Ziel von AMEffMo ist die Bestimmung des Aufwandes für eine bestimmte Wartungsaufgabe. Das Modell baut darauf auf, dass eine Korrelation zwischen dem Wartungsaufwand und leicht messbaren Eigenschaften des zu wartenden Quellcodes oder der Wartungsaufgabe besteht. Als Ergebnis der in [HayPatZha04] geschilderten Untersuchung wurden zwei verschiedene Berechnungsformeln für AMEffMo aufgestellt:

1. $E = 63 + 0,1 \cdot DLOC$
2. $E = -124 + 7,5 \cdot DNoprtr$

Dabei ist E der Wartungsaufwand in Personen-Stunden. $DLOC$ ist die Anzahl der Codezeilen, die während der Wartung geändert, hinzugefügt oder gelöscht werden. Und $DNoprtr$ ist die Anzahl der Operatoren, die während der Wartung geändert, hinzugefügt oder gelöscht werden. Beide Formeln wurden mittels linearer Regression über den Daten von vier kleineren, anscheinend sehr verschiedenen Projekten ermittelt. Die Wahl zwischen den beiden Formeln wird dem Benutzer des Modells überlassen. Je nachdem, ob er eher die Anzahl der für die bevorstehende Wartungsaufgabe zu ändernden Codezeilen oder Operatoren abschätzen kann, kann er sich frei für eine der beiden Formeln entscheiden.

Die Validierung der beiden Formeln ist leider nicht sehr umfangreich. Die Autoren selbst räumen ein, dass noch eine größere Studie zur Validierung des Modells nötig ist. Allerdings gibt es zurzeit keine Anzeichen dafür, dass eine solche Studie stattfindet. Weiterhin macht auch die lange Fassung von [HayPatZha04] nur sehr wenige Aussagen über die Eigenschaften der untersuchten Programme, sowie der verwen-

deten Methoden und Werkzeuge, sodass nicht feststellbar ist, auf welche Art von Projekten die ermittelten Gewichtungen angewendet werden können. Der Zusammenhang zwischen den verwendeten Metriken (*DLOC* und *DNoprtr*) und dem Wartungsaufwand wurde im Rahmen einer Korrelationsanalyse mit 17 weiteren Metriken gezeigt und erscheint intuitiv auch als anwendbar. Die untersuchten Metriken zeigen allerdings deutlich, dass das Modell auf den Quellcode der Software beschränkt ist und keinerlei Dokumentation oder ähnliches beachtet. Eigenschaften des Wartungsprozesses, der Umgebung und des Personals werden vollständig in den Gewichtungen erfasst, wodurch eine ständige Überprüfung der Gewichtungen notwendig wird.

3.2 Effort Estimation Model for Corrective Software Maintenance

Das „Effort Estimation Model for Corrective Software Maintenance“ wurde 2002 von Andrea De Lucia, Eugenio Pompella und Silvio Stefanucci veröffentlicht ([LucPomSte02]). Es dient zur Bestimmung des Wartungsaufwandes für eine Wartungsphase auf Grundlage der Art und Anzahl der in dieser Wartungsphase getätigten Wartungsaufgaben und der Größe des zu wartenden Systems. Die Größe der Wartungsaufgaben wird dabei nicht beachtet, um die für die Schätzung des Aufwandes nötigen Daten möglichst einfach ermittelbar zu halten. Das Modell beschränkt sich auf korrektive Wartung entsprechend der Definition nach IEEE und unterscheidet darin drei Arten von Wartungsaufgaben, die unterschiedlichen Einfluss auf den Wartungsaufwand haben sollen:

- Typ A: Wartungsaufgaben, die Änderungen am Quellcode umfassen
- Typ B: Wartungsaufgaben, die Korrekturen an der Datenbasis erfordern
- Typ C: sonstige Wartungsaufgaben

Je nachdem wie gut die einzelnen Aufgaben der bevorstehenden Wartungsphase in diese drei Typen eingeteilt werden können, kann der Benutzer des Modells eine der drei Berechnungsformeln des Modells verwenden:

1. $Effort = a_1 \cdot N + a_2 \cdot Size$
2. $Effort = a_3 \cdot NA + a_4 \cdot NBC + a_5 \cdot Size$
3. $Effort = a_6 \cdot NA + a_7 \cdot NB + a_8 \cdot NC + a_9 \cdot Size$

Dabei ist *Effort* der Wartungsaufwand in Personen-Stunden. N ist die Anzahl aller Wartungsaufgaben. *Size* ist die Größe des zu wartenden Systems in kLOC (= Anzahl der Codezeilen / 1000). NA ist die Anzahl der Wartungsaufgaben vom Typ A. NBC ist die Anzahl der Wartungsaufgaben der Typen B und C zusammengefasst. NB ist die Anzahl der Wartungsaufgaben vom Typ B. Und NC ist die Anzahl der Wartungsaufgaben vom Typ C. Die Gewichtungen a_1 bis a_9 sollen projekt- oder unternehmensspezifisch mit Hilfe einer Regressionsanalyse über historische Daten des konkreten Projektes oder Unternehmens ermittelt werden. In [LucPomSte02] werden allerdings kaum Hinweise gegeben, wie die Ermittlung der Gewichtungen geschehen soll und wann eventuell eine Neuanpassung nötig ist. Die Anwendbarkeit der in der geschilderten Untersuchung ermittelten Gewichtungen für andere Projekte lässt sich nur schwer feststellen, da keine Angaben zu den verwendeten Programmiersprachen, den während der Wartung eingesetzten Methoden oder verwendeten Werkzeugen vorhanden sind.

Der Gedanke, die nötigen Wartungsaufgaben in verschiedene Typen einzuteilen, um sie mit unterschiedlichen Gewichtungen in die Aufwandsschätzung einfließen zu lassen, erscheint sinnvoll. Die in [LucPomSte02] geschilderte Untersuchung untermauert dies auch. Allerdings ist eine sinnvolle Einteilung der Wartungsaufgaben höchstwahrscheinlich stark vom Anwendungsbereich der Software abhängig. Die in [LucPomSte02] untersuchten Projekte stammten alle aus dem Feld der Business-Anwendungen. Für reine Maschinensteuerungen oder ähnliches müssten wohl andere Einteilungen gefunden werden.

3.3 FEAST Effort Model

Das „FEAST Effort Model“ wurde 2000 von Juan F. Ramil und Meir M. Lehman vorgestellt ([RamLeh00]). Es wurde im Rahmen des FEAST-Projektes entwickelt, dass sich mit der Untersuchung der Software-Evolution befasste (FEAST = Feedback, Evolution And Software Technology). Ziel dieses Modells ist es, den Aufwand für die kommende Evolutionsphase zu schätzen, wobei dabei die historischen Aufwandsdaten vergangener Phasen, sowie die geplanten Änderungen an der Software und Änderungen an den Kostenfaktoren zugrunde gelegt werden.

Das Modell wurde speziell darauf ausgelegt, auf Grundlage von ungeplant erfassten Aufwandsdaten arbeiten zu können, so dass für die Anwendung des Modells keine lange Vorbereitungsphase nötig ist. Deshalb baut das Modell auf der Analyse einfacher Changelogs auf. In diesen muss lediglich angegeben sein, welche Person eine Änderung an welchem Teil des Quellcodes durchgeführt hat oder welche Quellcode-Datei neu eingefügt hat. Das Löschen von Quellcode-Dateien muss nicht erfasst werden, da das Modell davon ausgeht, dass das Löschen von Dateien keinen

Aufwand verursacht. Zur Bestimmung des in einem bestimmten Zeitraum erbrachten Arbeitsaufwandes wird die Anzahl von verschiedenen Namen in den Changelogs dieses Zeitraums ermittelt und mit der möglichen Arbeitszeit pro Person in diesem Zeitraum multipliziert. Dabei wird davon ausgegangen, dass alle Entwickler in Vollzeit gearbeitet haben und nicht zwei Entwickler über einen längeren Zeitraum zusammen an einem Teil des Quellcodes gearbeitet haben.

Neben dem Aufwand wurden folgende einfache Metriken identifiziert, die leicht aus den Changelogs ermittelbar sind und in Korrelation zum Aufwand stehen:

- Anzahl der Änderungen an vorhandenen Modulen (*ModifHandlings*)
- Anzahl der geänderten Module (*ModulesChanged*)
- Anzahl der Module, die dem System hinzugefügt wurden (*ModulesCreated*)
- Anzahl der Hinzufügungen und Änderungen am System (*TotalHandlings*)
- Anzahl der hinzugefügten und geänderten Module (*ModulesHandled*)
- Anzahl von Subsystemen mit geänderten Modulen (*SubsysChanged*)
- Anzahl von Subsystemen mit hinzugefügten und/oder geänderten Modulen (*SubsysHandled*)
- Anzahl der Subsysteme mit hinzugefügten Modulen (*SubsysInclCreations*)

Mit diesen Metriken wurden sechs verschiedene Formeln zur Abschätzung des Aufwandes erstellt, deren Gewichtungen mittels linearer Regression über die Daten der Changelogs vergangener Evolutionsphasen unternehmens- oder projektspezifisch ermittelt werden müssen. Diese Formeln lauten folgendermaßen:

1. $Effort = A_1 \cdot ModulesHandled + B_1$
2. $Effort = A_2 \cdot SubsysHandled + B_2$
3. $Effort = A_3 \cdot TotalHandlings + B_3$
4. $Effort = A_{4_1} \cdot ModulesCreated + A_{4_2} \cdot (ModulesChanged - ModulesCreated) + B_4$
5. $Effort = A_{5_1} \cdot ModulesCreated + A_{5_2} \cdot (ModifHandlings - ModulesCreated) + B_5$
6. $Effort = A_{6_1} \cdot SubsysInclCreations + A_{6_2} \cdot SubsysChanged + B_6$

Welche der sechs Formeln für ein spezielles Projekt oder Unternehmen am besten geeignet ist, muss ebenfalls im Rahmen der Regressionsanalyse ermittelt werden. Es können natürlich auch mehrere Formeln gleichzeitig verwendet werden, um einen Mittelwert oder ähnliches zu ermitteln.

In [RamLeh00] wird darauf hingewiesen, dass die Gewichtungen der Berechnungsformeln bei starken Änderungen in der Software oder im Unternehmen neu ermittelt werden müssen. Zu diesen Änderungen zählen z. B. ausgedehnte Code-Refactorings, eine wesentliche Verbesserung der Dokumentation oder auch bedeutende Änderungen in der Prozessstruktur (vergleiche dazu [Leh01]). Solche Änderungen können natürlich auch dazu führen dass eine andere Berechnungsformel passender ist, als die bisher verwendete.

Das FEAST-Projekt wurde 2001 abgeschlossen ([Leh01]). Die Arbeit an diesem Modell scheint somit abgeschlossen zu sein. Leider gibt es außer [RamLeh00], worin angeblich nur anfängliche Ergebnisse zu dem Modell geschildert sind, keine weitere Veröffentlichung zu diesem Modell.

3.4 Integrated Measure of Software Maintainability

Krishan K. Aggarwal, Yogesh Singh und Jitender Kumar veröffentlichten 2002 ein Modell zur Bestimmung der Wartbarkeit von Software, das mit Fuzzy-Logik arbeitet ([AggSinChh02]). Dieses Modell versucht die Subjektivität der Wartbarkeit mitzuerfassen. In der 2002 veröffentlichten Version des Modells wird die Wartbarkeit in die drei Teile Lesbarkeit des Quellcodes, Qualität der Dokumentation und Verständlichkeit der Software aufgegliedert. Diese drei Teile werden separat mit Metriken gemessen.

Der Wert der Lesbarkeit wird über das Verhältnis von Kommentarzeilen zur Gesamtzahl der Zeilen im Quellcode bestimmt. Die Qualität der Dokumentation wird über den Fog-Index bestimmt, der wie folgt definiert ist:

$$FOG = 0,4 \cdot \left(\frac{\text{Wortzahl}}{\text{Anzahl der Sätze}} + \text{Anteil der Worte mit 3 oder mehr Silben} \right)$$

Da dieser Index den Anteil der Worte mit drei oder mehr Silben erfasst, ist er so wahrscheinlich nicht auf deutschsprachige Dokumente anwendbar. Die Verständlichkeit der Software wird mit Hilfe eines Werkzeugs von Kari Laitnen ermittelt, das die Dokumentation der Software auf Vorkommen von Symbolen aus dem Quellcode untersucht (siehe [Lai96]). Die Werte der drei Teile werden auf Skalen eingetragen, die jeweils in drei Bereiche (gut, mittel und schlecht) eingeteilt sind. So wird für jeden Teil der Wartbarkeit ein Bereich ermittelt. Die drei ermittelten Bereiche werden schließlich mit Hilfe von 27 logischen Regeln zum Wert der Wartbarkeit kombiniert. Dieser Wert liegt auf einer Skala von 0 bis 10, wobei 0 eine sehr gute und 10 eine sehr schlechte Wartbarkeit bedeutet.

Leider gibt [AggSinChh02] keinen Aufschluss darüber, welchen Nutzen das relativ komplizierte Berechnungsverfahren dieses Modell gegenüber anderen Verfahren haben soll. Das Berechnungsverfahren geht von eindeutigen Zahlwerten aus und endet auch wieder mit solchen. Es wäre somit vermutlich sinnvoller die drei definierten Metriken in einer Berechnungsformel mit Gewichtungen zu einem Wartbarkeitswert zu verknüpfen. Weiterhin begründet [AggSinChh02] nicht, warum gerade die drei gewählten Metriken ausgesucht wurden. Der Zusammenhang dieser Metriken mit der Wartbarkeit wird nicht gezeigt. Positiv anzumerken ist bei diesem Modell, dass versucht wurde, auch die Dokumentation in die Wartbarkeitsbetrachtung mit einzu beziehen.

In [AggSinCha05] stellen Krishan K. Aggarwal, Yogesh Singh, Pravin Chandra und Manimala Puri eine weitere Version dieses Modells vor. Diese beruht auf dem gleichen Berechnungsverfahren, verwendet aber andere Metriken. Leider wird mit den neu gewählten Metriken nur noch der Quellcode der Software betrachtet. In dieser Version des Modells werden die folgenden vier Metriken kombiniert:

- durchschnittliche Anzahl aktiver Variablen pro Anweisung im Quellcode
- durchschnittliche Lebensdauer von Werten
- Verhältnis von Kommentarzeilen zur Gesamtzahl der Zeilen im Quellcode
- durchschnittliche zyklomatische Komplexität nach McCabe

3.5 Maintainability Index (MI)

Der „Maintainability Index“ wurde bereits in den 1990er Jahren entwickelt ([ColAshLow94], [WelOma95]). Er dient zur Bestimmung der Wartbarkeit von Quellcode. Zur Entwicklung dieses Indexes wurden verschiedene, sehr große Industrieprojekte mit Quellcode von bis zu 240000 Zeilen Code untersucht ([WelOmaAtk97]). Ziel war es, eine Berechnungsformel für die Wartbarkeit eines Softwaremoduls zu finden, die auf möglichst einfach zu messenden Codemetriken aufbaut.

Dafür wurde zunächst die Wartbarkeit der einzelnen Module des untersuchten Quellcodes von Experten eingeschätzt. Als Schätzhilfe diente ein Fragebogen, der 25 Fragen enthielt, die jeweils mit ein bis fünf Punkten bewertet werden konnten. Die Gesamtzahl der für ein Modul vergebenen Punkte ergab schließlich die Wartbarkeit des Moduls. Wobei die Minimalpunktzahl 25 eine schlechte Wartbarkeit implizierte und die Maximalpunktzahl von 125 eine sehr gute Wartbarkeit (vergleiche hierzu [VanSciSpr02]). Nachdem die Wartbarkeit der einzelnen Module mehr oder weniger subjektiv eingeschätzt wurde, wurden verschiedene Metrikwerte für die einzelnen Module ermittelt und auf Korrelation mit der geschätzten Wartbarkeit untersucht. Dabei wurden besonders starke Korrelationen bei den Metriken über die zyklomatische Komplexität festgestellt.

3 Kurzbeschreibung der analysierten Modelle

matische Komplexität nach McCabe ($V(g)$) und Myers ($V(g')$) festgestellt. Auch die Metriken über Aufwand (E) und Volumen (V) nach Halstead, die Anzahl der Codezeilen (LOC) und die Anzahl der Kommentarzeilen (CMT) zeigten starke Korrelation mit den eingeschätzten Wartbarkeiten. Aufgrund dieser Korrelationen wurden zwei Berechnungsformeln für den MI aufgestellt:

1. $MI = 171 - 5,2 \cdot \ln(aveV) - 0,23 \cdot aveV(g') - 16,2 \cdot \ln(aveLOC)$
2. $MI = 171 - 5,2 \cdot \ln(aveV) - 0,23 \cdot aveV(g') - 16,2 \cdot \ln(aveLOC) + 50 \cdot \sin(\sqrt{2,4 \cdot perCM})$

Dabei bedeutet *ave* jeweils, dass der Durchschnittswert der Metrik über alle betrachteten Module gebildet werden soll. Die zweite Formel bezieht im Gegensatz zur ersten den durchschnittlichen Kommentaranteil (*perCM*) der betrachteten Module mit ein. Sie soll nur angewendet werden, wenn die Kommentare in den Modulen auch sinnvolle, ergänzende Informationen zum Quellcode enthalten. Bei Anwendung der zweiten Formel ist also eine Code-Inspektion nötig.

Das Modell definiert für beide Berechnungsformeln je eine Skala, die die ermittelte Wartbarkeit in verschiedene Stufen einteilt. Für die erste Berechnungsformel gilt folgende Einteilung:

- gute Wartbarkeit, falls $MI \geq 50$
- schlechte Wartbarkeit, falls $MI < 50$

Für die zweite Formel gilt folgende Einteilung:

- gute Wartbarkeit, falls $MI \geq 85$
- akzeptable Wartbarkeit, falls $65 \leq MI < 85$
- schlechte Wartbarkeit, falls $MI < 65$

Aufgrund seiner einfachen Berechnungsformel ist der MI ein Modell zur Bestimmung der Wartbarkeit, mit dem sich die Literatur häufig beschäftigt. In [Lis01] stellt Aldo Liso z. B. eine Verbesserung der zweiten Formel vor, durch die die Gewichtung des Kommentaranteils auf verschiedene Sprachen anpassbar wird. In vielen anderen in diesem Bericht genannten Veröffentlichungen wird der MI als Vergleichsmodell in den Untersuchungen von neuen Modellen herangezogen. Dadurch ist er das am besten validierte Modell in dieser Auflistung.

3.6 Maintainability Model of Object-Oriented Design

Das „Maintainability Model of Object-Oriented Design“ wurde 2004 von Matinee Kiewkanya, Nongyao Jindasawat und Pornsiri Muenchaisri veröffentlicht ([KieJinMue04]). Dieses Modell dient zur Ermittlung der Wartbarkeit eines objekt-orientierten Software-Entwurfs. Zur Bestimmung der Wartbarkeit werden UML-Diagramme auf einfach zu messende Eigenschaften untersucht. Dabei werden sowohl Klassen-Diagramme, als auch Sequenz-Diagramme betrachtet. Zur Validierung des Modells wurden 40 kleinere Software-Entwürfe verwendet. Diese wurden von insgesamt 60 Studenten mit Hilfe von Fragebögen mit 30 Fragen nach Verständlichkeit und Änderbarkeit bewertet. Diese beiden Qualitäten werden in diesem Modell als die hauptsächlichen Bestandteile der Wartbarkeit definiert. Die Bewertung der Wartbarkeit ergab sich in dieser Untersuchung aus der Summe der Bewertungen von Verständlichkeit und Änderbarkeit. Weiterhin wurden jeweils 18 Metrikwerte für die Entwürfe erhoben. Die folgenden Tabellen zeigen die erhobenen Metriken.

Metriken für Klassen-Diagramme		
Klassen		Anzahl der Klassen (NC)
Attribute		durchschnittl. Anzahl von Attributen – ungewichtet ($ANAUW$)
		durchschnittl. Anzahl von Attributen – gewichtet ($ANAW$)
Methoden		durchschnittl. Anzahl von Methoden – ungewichtet ($ANMUW$)
		durchschnittl. Anzahl von Methoden – gewichtet ($ANMW$)
Beziehungen	Assoziation	durchschnittl. Anzahl von Assoziationsbeziehungen ($ANAsso$)
	Aggregation	durchschnittl. Anzahl von Aggregationsbeziehungen ($ANAgg$)
		Anzahl der Aggregationshierarchien ($NAggH$)
		maximale Anzahl der Aggregationshierarchien ($MaxHAgg$)
	Generalisierung	durchschnittl. Anzahl von Generalisierungsbeziehungen ($ANGen$)
		Anzahl der Generalisierungshierarchien ($NGenH$)
maximale Anzahl der Generalisierungshierarchien ($MaxDIT$)		

Tabelle 3: Metriken für Klassen-Diagramme des Maintainability Model of Object-Oriented Design

Metriken für Sequenz-Diagramme	
Szenarien	Anzahl von Szenarien (<i>NOS</i>)
Nachrichten	gewichtete Nachrichten zwischen Objekten (<i>WMBO</i>)
	durchschnittl. Anzahl von Return Messages (<i>ANRM</i>)
	durchschnittl. Anzahl direkt versendeter Nachrichten (<i>ANDM</i>)
	durchschnittl. Anzahl von Elementen im transitiven Abschluss der direkt versendeter Nachrichten (<i>ANET</i>)
Bedingungen	durchschnittliche Anzahl von Condition Messages (<i>ANCM</i>)

Tabelle 4: Metriken für Sequenz-Diagramme des Maintainability Model of Object-Oriented Design

Aus den subjektiven Einschätzungen der Studenten und den erhobenen Metrikerwerten wurden für das Modell jeweils drei Diskriminanzfunktionen für Wartbarkeit, Verständlichkeit und Änderbarkeit erstellt, die die Einstufung eines Software-Entwurfs in jeweils eine der drei Stufen schwer, mittel oder leicht zulässt. Dazu müssen nur die Werte der drei zusammengehörigen Diskriminanzfunktionen ermittelt werden. Die Diskriminanzfunktion, für die der höchste Wert ermittelt wird, bestimmt die Einstufung des Entwurfs. Zur Bestimmung der Wartbarkeitsstufe eines Entwurfs müssen also zum Beispiel die Werte der drei folgenden Diskriminanzfunktionen für leichte (*E_Main*), mittlere (*M_Main*) und schwere (*D_Main*) Wartbarkeit ermittelt und verglichen werden.

$$\begin{aligned}
 E_Main = & 0,985 \cdot NC + 11,574 \cdot ANAUW - 9,219 \cdot ANMUW + 7,224 \cdot ANAsso \\
 & + 12,094 \cdot NAggH + 0,1142 \cdot MaxHAgg + 3,039 \cdot NGenH - 0,195 \cdot MaxDIT \\
 & + 6,71 \cdot NOS + 0,002251 \cdot WMBO + 5,601 \cdot ANRM - 0,0205 \cdot ANDM \\
 & + 5,072 \cdot ANET - 3,499 \cdot ANCM - 49,816
 \end{aligned}$$

$$\begin{aligned}
 M_Main = & 0,8 \cdot NC + 9,27 \cdot ANAUW + 0,212 \cdot ANMUW + 3,78 \cdot ANAsso \\
 & + 16,933 \cdot NAggH + 1,274 \cdot MaxHAgg + 1,493 \cdot NGenH + 1,689 \cdot MaxDIT \\
 & + 6,889 \cdot NOS + 1,928 \cdot WMBO + 6,001 \cdot ANRM + 2,033 \cdot ANDM \\
 & + 3,937 \cdot ANET - 3,81 \cdot ANCM - 44,758
 \end{aligned}$$

$$\begin{aligned}
 D_Main = & 0,938 \cdot NC + 9,787 \cdot ANAUW - 3,94 \cdot ANMUW + 5,439 \cdot ANAsso \\
 & + 17,173 \cdot NAggH - 0,93 \cdot MaxHAgg + 4,941 \cdot NGenH + 2,219 \cdot MaxDIT \\
 & + 5,935 \cdot NOS + 0,198 \cdot WMBO + 5,935 \cdot ANRM + 0,911 \cdot ANDM \\
 & + 3,45 \cdot ANET - 3,969 \cdot ANCM - 44,446
 \end{aligned}$$

Aufgrund der vielen Metrikwerte, die erhoben werden müssen, und der komplizierten Berechnungen zur Ermittlung der Einstufung, wäre das Modell ohne eine Automatisierung nur schlecht verwendbar. Deshalb wurde im Rahmen der in [KieJinMue04] geschilderten Untersuchung ein Werkzeug entwickelt, das die einzelnen Metrikwerte ermitteln und die Einstufung vornehmen kann. Die Autoren geben allerdings an, dass das Werkzeug nur für UML-Diagramme verwendbar ist, die mit Rational Rose erstellt wurden.

Der Ansatz des „Maintainability Model of Object-Oriented Design“ eignet sich gut zur Vorhersage, ob ein bestimmter Software-Entwurf wartbar ist oder nicht. Da die Einstufung jedoch mit Diskriminanzfunktionen geschieht, die nur eine sehr grobe Einstufung ermöglichen, kann dieses Modell nur schlecht zur Überwachung der Wartbarkeit einer Software-Architektur verwendet werden. Die in den Diskriminanzfunktionen verwendeten Gewichtungen stammen aus einer Untersuchung mit eher kleinen Software-Entwürfen. Bei Verwendung großer Entwürfe müssten sie höchstwahrscheinlich angepasst werden.

3.7 Maintainability Prediction Model (MainPredMo)

Das „Maintainability Prediction Model“ (MainPredMo) wurde zwischen 2001 und 2005 von Jane Huffman Hayes und Liming Zhao entwickelt. Zur Analyse dieses Modells wurde mir von Liming Zhao eine ungekürzte, unveröffentlichte Fassung von [HayZha05] zur Verfügung gestellt, die weitere nützliche Informationen enthielt.

MainPredMo ist ein Modell zur Vorhersage der Wartbarkeit einer Software aufgrund der Aufwände für Anforderungsanalyse, Entwurf und Implementierung, die in die Software eingeflossen sind. Dem Modell liegt die Behauptung zu Grunde, dass die Wartbarkeit einer Software mit dem Verhältnis dieser Aufwände korreliert, welches als *RDCRatio* bezeichnet und folgendermaßen definiert wurde:

$$RDCRatio = \frac{ReqEffort + DesignEffort}{CodingEffort}$$

Wobei *ReqEffort* der Aufwand für die Anforderungsanalyse, *DesignEffort* der Aufwand für den Entwurf und *CodingEffort* der Aufwand für die Implementierung des Systems ist. Zur Bestätigung der Korrelation zwischen dem *RDCRatio* und der Wartbarkeit der Software wurde der *RDCRatio* verschiedener, kleinerer und meist studentischer Programme mit Wartbarkeitsschätzungen der Entwickler der Programme verglichen. Dabei konnte eine gute Korrelation zwischen dem *RDCRatio* und der geschätzten Wartbarkeit festgestellt werden. Die Veröffentlichung gibt allerdings keine Hinweise darauf, wie viele Personen jeweils Schätzungen für die verschiedenen

Programme abgegeben haben, wie die Personen auf die Schätzung vorbereitet wurden und ob die Schätzungen wirklich ehrlich waren. Als Anhaltspunkt für die Schätzungen diente die Vorgabe einer Skala von 1 bis 10. Wobei 10 bedeutete, dass der Quellcode sehr leicht änderbar ist. Wartbarkeit 1 bedeutete entsprechend, dass eine Code-Änderung nur sehr schwer durchzuführen ist. Die Vorgabe dieser Skala zeigt deutlich, dass sich dieses Modell ausschließlich mit der Wartbarkeit des Quellcodes befasst.

Für den *RDCRatio* schlägt das Modell zwei mögliche Verwendungen vor. Eine Verwendungsmöglichkeit besteht in der Erstellung einer Rangfolge der verschiedenen Module des Quellcodes entsprechend ihres *RDCRatio*, um die Entscheidung zu unterstützen, bei welchen Modulen die Wartbarkeit verbessert werden soll. Allerdings überlässt das Modell die Entscheidung, was ein guter und was ein schlechter *RDCRatio* ist, dem Benutzer. Die zweite Verwendungsmöglichkeit besteht in der Berechnung der Wartbarkeit über folgende Formel:

$$\text{MainPredMo} = 3,795 + 1,652 \cdot \text{RDCRatio}$$

Diese Formel entstand aus einer Regression über die ermittelten *RDCRatios* und die geschätzten Wartbarkeiten und dient dazu den *RDCRatio* auf eine der Schätzung entsprechende Skala zu transformieren. Die Verwendbarkeit dieser Formel ist allerdings zweifelhaft, da niemals ein Wert unter 3,795 erreicht werden kann, obwohl die ursprüngliche Skala bei 1 beginnt.

3.8 Model for Estimation and Prediction of adaptive Maintenance Effort of Object-Oriented Systems

Das „Model for Estimation and Prediction of adaptive Maintenance Effort of Object-Oriented Systems“ wurde von Fabrizio Fioravanti und Paolo Nesi entwickelt. Es soll dazu dienen den Wartungsaufwand für das nächste Release einer Software abzuschätzen. Das Modell geht davon aus, dass der bei der Wartung entstehende Aufwand mit Metrikwerten der Software folgendermaßen korreliert:

$$\text{Eff}_{am} \approx M_{am} = M_a - w_{M_b} \cdot M_b$$

Dabei ist Eff_{am} der Aufwand für die adaptive Wartung, M_{am} ist der Metrikwert für die adaptive Wartung, M_a ist der Metrikwert vor der adaptiven Wartung, M_b ist der (geschätzte) Metrikwert nach der adaptiven Wartung und w_{M_b} ist der Gewichtungsfaktor für M_b . Für M kann im Grunde eine beliebige Metrik eingesetzt werden, sofern sie mit dem Aufwand für die adaptive Wartung von objekt-orientierten Sys-

temen korreliert. Die Aussagekraft des Modells hängt entscheidend von der gewählten Metrik ab.

Im Rahmen der in [FioNes01] beschriebenen Untersuchung wurden verschiedene Codemetriken für dieses Modell vorgeschlagen, die die Eigenschaften objekt-orientierter Systeme besser erfassen, als allgemein bekannte Codemetriken wie z. B. LOC (= Anzahl der Codezeilen) und die zyklomatische Komplexität nach McCabe. Hierzu gehören insbesondere die CC-Metrik (siehe [FioNesPer98a]) und die NAM-Metrik (= Anzahl von Attributen und Metriken). Beide Metriken zeigten in der in [FioNes01] beschriebenen Untersuchung eine starke Korrelation mit dem Wartungsaufwand. Die Untersuchung fand allerdings nur an verschiedenen Releaseständen eines Systems statt.

Gerade bei der CC-Metrik ist die Bestimmung des Metrikwertes relativ aufwendig, da er aus vielen verschiedenen Werten zusammengesetzt wird. Deshalb wurde ein Werkzeug zur Erfassung der einzelnen Werte entwickelt, welches in [FioNesPer98b] vorgestellt wird. Die Gewichtungen der einzelnen Terme der CC-Metrik müssen allerdings vor Verwendung des Werkzeugs durch eine Regressionsanalyse projekt- oder unternehmensspezifisch ermittelt werden. Dazu schlägt das Modell die Analyse von Changelog-Einträgen vor, da diese in den meisten Unternehmen vorhanden sind. Aus den daraus ermittelten Aufwandsdaten und den Eigenschaften der Software müssen dann die Gewichtungen (w) für folgende Berechnungsformel ermittelt werden:

$$CC_{am} = CC_a - w_{cc_b} \cdot CC_b$$

Dabei ist die CC-Metrik wie folgt definiert:

$$\begin{aligned} CC &= ICI + ECD \\ &= w_{CL} \cdot CL + w_{CI} \cdot CI + ECDL + ECDI \\ &= w_{CL} \cdot CL + w_{CI} \cdot CI + w_{CACL} \cdot CACL + w_{CMICL} \cdot CMICL \\ &\quad + w_{CACI} \cdot CACI + w_{CMICI} \cdot CMICI \end{aligned}$$

3 Kurzbeschreibung der analysierten Modelle

Die einzelnen Terme der Metrik erfassen folgendes:

Term	Bedeutung
CC	Komplexität der Klassen (Class Complexity)
ICI	Komplexität der Implementierungen von Methoden (Internal Class Implementation)
ECD	Komplexität der Schnittstellen-Beschreibungen von Klassen (External Class Description)
CL	Komplexität der Implementierungen lokaler Methoden (Class complexity Local)
CI	Komplexität der Implementierungen ererbter Methoden (Class complexity Inherited)
ECDL	Komplexität lokaler Schnittstellen-Beschreibungen von Klassen (External Class Description Local)
ECDI	Komplexität ererbter Schnittstellen-Beschreibungen von Klassen (External Class Description Inherited)
CACL	Komplexität lokal definierter Attribute (Class Attribute Complexity Local)
CMICL	Komplexität von Parametern lokal definierter Methoden (Class Method Interface Complexity Local)
CACI	Komplexität ererbter Attribute (Class Attribute Complexity Inherited)
CMICI	Komplexität von Parametern ererbter Methoden (Class Method Interface Complexity Inherited)

Tabelle 5: Bedeutung der Bestandteile der CC-Metrik

3.9 Predictive Effort Model for Software Maintenance Releases

Das „Predictive Effort Model for Software Maintenance Releases“ beschäftigt sich stärker als andere Modelle mit dem individuellen Wartungsprozess. Ziel dieses Modells ist es, den Wartungsprozess eines konkreten Unternehmens oder Projektes so genau wie möglich zu verstehen, um so eine möglichst genaue Vorhersage des Wartungsaufwandes für das nächste Release eines Projektes machen zu können. Den Wartungsprozess verstehen bedeutet dabei, zu verstehen, von welchen Faktoren die Wartung und somit der Wartungsaufwand beeinflusst wird und wie sich diese Faktoren im Laufe der Zeit verändern. Der Schwerpunkt dieses Modells liegt deshalb auf dem Vorgehen zur Ermittlung einer Berechnungsformel für den zukünftigen Wartungsaufwand. Die ermittelte Formel selbst ist für das Modell weniger wichtig, da sie nur für ein Unternehmen oder gar nur für ein Projekt verwendbar ist.

Das diesem Modell zugrunde liegende Vorgehen basiert auf dem GQM-Paradigma (GQM = Goal / Question / Metric, siehe [BasRom88]). Es wird in [BasBriCon96] relativ detailliert beschrieben.

Um den Wartungsprozess genau zu verstehen, werden zunächst konkrete Analyseziele (Goals) aufgestellt. Diese werden jeweils über die Beantwortung folgender Fragen definiert:

1. Was soll analysiert werden?
2. Zu welchem Zweck soll diese Analyse durchgeführt werden?
3. Auf welche Eigenschaften des Prozesses soll bei der Analyse geachtet werden?
4. Aus der Sicht welcher Person(en) soll die Analyse stattfinden?

In [BasBriCon96] sind drei solcher Analyseziele angegeben, die gut als Ausgangspunkt für eine unternehmens- oder projektspezifische Analyse genutzt werden können. Mit steigendem Wissen über den eigenen Prozess sollten aber möglichst individuelle Ziele definiert werden.

Nach der Definition der Analyseziele wird nach konkreten Fragen (Questions) gesucht, deren Beantwortung dazu führt, dass die Analyseziele erreicht werden. Für jedes Analyseziel werden eigene Fragen aufgestellt. Je mehr Ziele und Fragen man definiert, desto genauer wird die Analyse natürlich. Allerdings steigt dabei auch der Analyseaufwand stark an.

Die Analyseziele und die zugehörigen Fragen müssen natürlich dazu dienen, den Einfluss verschiedener Faktoren auf den Wartungsaufwand zu erkennen. Sie führen also immer dazu, dass die Entwicklung verschiedener Faktoren und des Wartungsaufwandes über einen längeren Zeitraum erfasst werden muss. Um diese Entwick-

lung erfassen zu können müssen passende Metriken (Metrics) ausgewählt und erhoben werden. Aus den so erfassten Werten wird schließlich mit Hilfe einer Regressionsanalyse eine Berechnungsformel zur Vorhersage des Wartungsaufwandes ermittelt. Das Modell legt nicht fest, welche Struktur diese Formel haben soll.

3.10 Software Maintainability Index (SMI)

Der „Software Maintainability Index“ wurde 1998 von Sridhar Muthanna vorgestellt. Ziel dieses Modells ist die Ermittlung der Wartbarkeit einer Software. Dabei wurde besonderer Wert auf eine möglichst einfache Berechnung gelegt. Die Berechnungsformeln des Modells zeigen große Ähnlichkeit mit dem Maintainability Index (siehe Kapitel 3.5). Allerdings wird hier größerer Wert auf die Verwendung von Design-Metriken gelegt, um eine möglichst frühzeitige Abschätzung der Wartbarkeit zu ermöglichen. Weiterhin wird durch [MutPonKon00] deutlich, dass beim SMI auch wesentlich mehr Wert auf das Vorgehen zur Ermittlung einer passenden Berechnungsformel gelegt wird, als beim MI.

Das in [MutPonKon00] beschriebene Vorgehen zur Ermittlung einer Berechnungsformel für den SMI besteht aus vier Schritten:

1. Auswahl geeigneter Metriken
2. Sammlung von Vergleichsdaten
3. Korrelationsanalyse
4. Regressionsanalyse und Aufstellung der Formel

Im Schritt zur Auswahl geeigneter Metriken wird zunächst eine möglichst umfangreiche Liste verschiedenster Design-Metriken zusammengestellt, die auf die zu analysierende Software anwendbar sind. Aus dieser Liste werden alle Metriken gestrichen, deren Werte sehr schwer zu ermitteln sind, um die spätere Berechnungsformel möglichst einfach zu halten. Aus den verbleibenden Metriken wird mit Hilfe einer Korrelations-Analyse nach Spearman-Pearson eine minimale Menge von Metriken ermittelt, die nicht miteinander korrelieren. Alle anderen Metriken werden aus der Liste gestrichen.

Im zweiten Schritt wird die Wartbarkeit verschiedener Softwaresysteme subjektiv ermittelt, um Vergleichsdaten für die Korrelationsanalyse im dritten Schritt und die Regressionsanalyse im vierten Schritt zu erhalten. Hierfür wird wie beim MI ein Fragebogen verwendet, bei dem für verschiedene Fragen Punkte von 1 bis 5 vergeben werden können. Der hier verwendete Fragebogen enthält allerdings 32 statt 25 Fragen.

Im dritten Schritt werden mittels einer Korrelationsanalyse die Metriken für die Berechnungsformel aus der im ersten Schritt erstellten minimalen Liste ausgewählt.

Im vierten Schritt werden schließlich die Gewichtungen für die im dritten Schritt ausgewählten Metriken ermittelt. Dies geschieht mit Hilfe einer linearen Regressionsanalyse. Aus den ausgewählten Metriken und deren Gewichtungen wird schließlich die Berechnungsformel erstellt.

Nach diesem Verfahren wurde in [MutPonKon00] folgende Berechnungsformel ermittelt:

$$SMI = 125 - 3,989 \cdot FAN_{avg} - 0,954 \cdot DF - 1,123 \cdot MC_{avg}$$

Dabei ist FAN_{avg} die durchschnittliche Anzahl externer Aufrufe aus einem Modul heraus, DF ist die Gesamtzahl ausgehender und eingehender Datenflüsse eines Moduls und MC_{avg} ist die durchschnittliche zyklomatische Komplexität der Module nach McCabe.

3.11 Software Maintenance Project Effort Estimation Model (SMPEEM)

Das „Software Maintenance Project Effort Estimation Model“ wurde 2003 von Yunsik Ahn, Jungseok Suh, Seungryeol Kim und Hyunsoo Kim vorgestellt ([AhnSuhKim03]). Ziel dieses Modells ist die Abschätzung des Aufwandes für eine bestimmte Wartungsaufgabe oder ein Wartungsprojekt. Dabei wurde versucht, Eigenschaften des Umfelds, in dem die Wartung stattfindet mit in die Schätzung einfließen zu lassen.

Als Grundlage der Schätzung dienen die Function Points der Wartungsaufgabe bzw. des Wartungsprojektes. Diese werden entsprechend dem „Function Point Counting Practices Manual“ der International Function Point Users Group ermittelt (siehe [IFPUG99]). Diese im Modell als „unadjusted function count“ (FC) bezeichneten Function Points werden dann mit verschiedenen Faktoren in den „adjusted function count“ (FP) überführt. Dies geschieht mit Hilfe folgender Berechnungsformel:

$$FP = FC \cdot \left(1 - \frac{\delta}{100} + \frac{2\delta}{5000} \cdot VAF \right)$$

In dieser Formel kann zunächst über den Parameter δ bestimmt werden, wie stark die Aufwandsabschätzung von Veränderungen im Wartungsumfeld beeinflusst werden soll. Für ein δ von 20% variiert FP z. B. zwischen $80\% \cdot FC$ für $VAF = 0$ (Minimalwert) und $120\% \cdot FC$ für $VAF = 50$ (Maximalwert). Ein δ von 20% erzeugt also

3 Kurzbeschreibung der analysierten Modelle

eine Varianz von $\pm 20\%$. Der *VAF*-Wert (*VAF* = value adjustment factors) wiederum ergibt sich aus Eigenschaften des Umfeldes, in dem die Wartung stattfindet. Diese Eigenschaften sind im Modell in drei Gruppen eingeteilt:

1. Fähigkeiten des Personals
2. technische Eigenschaften der Software
3. Wartungsumgebung

Diesen drei Gruppen sind verschiedene Faktoren zugeordnet:

Gruppe	Faktoren
Fähigkeiten des Personals	(1) Wissen über den Anwendungsbereich (2) Vertrautheit mit der Programmiersprache (3) Erfahrung mit der Systemsoftware (OS, DBMS)
technische Eigenschaften der Software	(4) Strukturiertheit der Softwaremodule (5) Unabhängigkeit zwischen den Softwaremodulen (6) Änderbarkeit/Lesbarkeit der Programmiersprache (7) Wiederverwendbarkeit alter Softwaremodule
Wartungs- umgebung	(8) Aktualität der Dokumentation (9) Konformität mit Software-Engineering-Standards (10) Testbarkeit

Tabelle 6: Faktoren für den *VAF*-Wert von SMPEEM mit zugehörigen Gruppen

Jedem der 10 Faktoren wird ein Wert zwischen 0 und 5 zugeordnet. Wobei 0 die schlechteste Bewertung ist und 5 die beste. Außerdem erfolgt eine Gewichtung des Einflusses der einzelnen Gruppen und der Faktoren innerhalb einer Gruppe. Die Bewertungen der einzelnen Faktoren müssen natürlich unternehmens- oder projekt-spezifisch ermittelt und fortlaufend angepasst werden. Auch die Gewichtungen der einzelnen Gruppen und Faktoren sollten in regelmäßigen Abständen überprüft und gegebenenfalls korrigiert werden. Aus den Werten der Faktoren und den Gewichtungen errechnet sich schließlich der *VAF*-Wert wie folgt:

$$VAF = \sum_{gruppe=1}^3 \left(\sum_{faktor=1}^k \frac{wert \cdot Wf}{100} \cdot \frac{Wg}{100} \right)$$

Dabei ist k die Anzahl der Faktoren in der aktuellen Gruppe, $wert$ ist der für den aktuellen Faktor vergebene Wert zwischen 0 und 5, W_f ist die Gewichtung des aktuellen Faktors innerhalb der aktuellen Gruppe in Prozent und W_g ist die Gewichtung der aktuellen Gruppe in Prozent. Die Summen der Gewichtungen aller Faktoren innerhalb einer Gruppe und der Gruppen selbst müssen jeweils 100% ergeben.

Aus den mit Hilfe der Faktoren VAF und δ ermittelten angepassten Function Points (FP) kann schließlich der Wartungsaufwand über folgende Formel abgeschätzt werden:

$$Aufwand = a \cdot FP^b$$

Die Faktoren a und b müssen dabei wiederum projekt- oder unternehmensspezifisch mit Hilfe von Regression aus historischen Daten ermittelt werden.

4 Fragebogen zur Analyse von Modellen zur Wartbarkeit von Software

Im Anhang A dieses Berichtes befindet sich der von mir entworfene Fragebogen zur Analyse von Modellen zur Wartbarkeit von Software. Er besteht im Wesentlichen aus drei weitgehend unabhängigen Teilen, deren Seiten wegen ihrer Unabhängigkeit auch getrennt nummeriert sind. Der erste Teil umfasst Fragen zur Analyse einer einzelnen Veröffentlichung. Im zweiten Teil befinden sich Fragen zur Analyse eines Modells. Der dritte Teil enthält Fragen zur Analyse einer einzelnen Metrik. Das folgende Kapitel soll verdeutlichen, warum für den Fragebogen gerade die darin enthaltenen Fragen ausgewählt wurden. Zum besseren Verständnis sollten Sie beim Lesen dieses Kapitels den Fragebogen zur Hand nehmen.

4.1 Analyse einer Veröffentlichung

Dieser Teil des Fragebogens dient zur Analyse von Veröffentlichungen, die über Untersuchungen zu Modellen zur Wartbarkeit von Software berichten. Dabei kann es sich um Untersuchungen zur Validierung, Widerlegung oder Automatisierung von Modellen oder Teilen davon handeln. Als Teil eines Modells ist hier z. B. die Annahme eines Zusammenhangs zwischen einer im Modell verwendeten Metrik und einem bestimmten Merkmal von Software oder Prozessen oder die Anwendbarkeit einer Berechnungsformel auf eine bestimmte Programmiersprache zu verstehen.

Die Fragen zu Veröffentlichungen sind in sechs Abschnitte gegliedert. Zu Beginn erfolgt die Erfassung allgemeiner Daten der Veröffentlichung. Danach wird die Qualität der Veröffentlichung analysiert. In den nächsten Abschnitten folgen Fragen zum Ziel, der Ausführung und den Ergebnissen der in der Veröffentlichung beschriebenen Untersuchung. Im letzten Abschnitt erfolgt eine kurze, abschließende Bewertung der Veröffentlichung.

4.1.1 Allgemeine Daten

Die von mir als „allgemeine Daten“ einer Veröffentlichung bezeichneten Informationen sollen auf einen Blick zeigen, was für eine Veröffentlichung überhaupt analysiert wurde. Zur genauen Bestimmung der Herkunft der Veröffentlichung werden hier der oder die Titel, die Autoren, der Publikationsort und das Datum der Veröffentlichung erfasst. Zur kurzen Beschreibung des Inhalts erfolgt eine Nennung der an der beschriebenen Untersuchung beteiligten Firmen und Personen, sowie der betrachteten Modelle und Metriken und eine kurze Zusammenfassung der Veröffentlichung.

4.1.2 Qualität der Veröffentlichung

Bei der Analyse verschiedener Veröffentlichungen ist mir aufgefallen, dass es schnell Vorkommen kann, dass ein Leser durch einen eleganten Ausdrucksstil einer Veröffentlichung davon abgelenkt wird, die damit vermittelten Inhalte auch auf ihre Plausibilität zu überprüfen. Aus diesem Grund habe ich den Abschnitt „Qualität der Veröffentlichung“ in den Fragebogen eingefügt. Er soll vor allem dabei helfen, eine kritische Haltung gegenüber einer Veröffentlichung einzunehmen.

Die ersten Fragen dieses Abschnitts sollen den Leser dazu anregen, Aussagen zu überprüfen, die aus anderen Veröffentlichungen stammen und der zu analysierenden Veröffentlichung zugrunde liegen. Danach folgen Fragen zu wichtigen Aussagen und Schlussfolgerungen der Veröffentlichung selbst. Alle grundlegenden Aussagen und Schlussfolgerungen, die gemacht werden und in irgendeiner Weise Einfluss auf die Ergebnisse der beschriebenen Untersuchung haben, sollten natürlich nachvollziehbar begründet sein. Werden Tabellen oder Diagramme verwendet, die die Aussagen unterstützen sollen, so sollte genau geprüft werden, ob die Diagramme auch wirklich das zeigen, was behauptet wird. Auch angegebene Beispiele sollten auf ihre Glaubhaftigkeit überprüft werden. Da in vielen Veröffentlichungen verschiedene Berechnungsformeln aufgestellt und verglichen werden, um die beste zu ermitteln oder Zusammenhänge zwischen verschiedenen Merkmalen zu zeigen, sollte geprüft werden, ob die verschiedenen Formeln tatsächlich verschieden sind. Sollte dies nicht der Fall sein, so sollte geprüft werden, ob gleiche Formeln auch gleiche Ergebnisse liefern.

Fallen bei der Beantwortung der Fragen in diesem Abschnitt sehr viele Unstimmigkeiten in der zu analysierenden Veröffentlichung auf, so sollte man in Betracht ziehen, die Analyse der Veröffentlichung abzubrechen, da die Aussagekraft der Veröffentlichung stark anzuzweifeln ist.

4.1.3 Analyse der Untersuchung, über die die Veröffentlichung berichtet

Wesentlicher Inhalt der zu analysierenden Veröffentlichung ist die Beschreibung einer Untersuchung von Modellen zur Wartbarkeit von Software. Aus dieser Beschreibung müssen die Informationen zu den einzelnen Modellen entnommen werden können.

Eine wesentliche Eigenschaft dieser Untersuchung ist natürlich ihr Ziel. Da der Fragebogen ausschließlich zur Analyse von Untersuchungen dienen soll, die sich mit Modellen zur Wartbarkeit von Software beschäftigen, habe ich die möglichen Ziele von Untersuchungen, die mit diesem Fragebogen analysiert werden können, eingeschränkt. Selbstverständlich dient der Fragebogen zur Analyse von Untersuchungen

zur Validierung oder Widerlegung von Modellen oder Teilen davon. Weiterhin sollen aber auch reine Untersuchungen von Metriken auf Zusammenhänge mit der Wartbarkeit, dem Wartungsaufwand oder Einflussfaktoren davon mit Hilfe dieses Fragebogens erfasst werden, weil diese für die Auswahl passender Metriken für ein Modell sehr wichtig sind. Da die Anwendbarkeit eines Modells entscheidend durch den Aufwand zur Ermittlung der nötigen Daten beeinflusst wird, sollten außerdem Untersuchungen zur Automatisierung von Modellen oder Teilen davon analysiert werden. Die Analyse von Untersuchungen mit anderen Zielen halte ich für das Thema dieser Diplomarbeit nicht für sinnvoll.

Zur Bestimmung der Verwendbarkeit der Ergebnisse einer Untersuchung muss deren Ausführung genau analysiert werden. In dem dazu gehörigen Abschnitt des Fragebogens werden zunächst das Alter und die Dauer der Untersuchung erfasst. Gerade bei Untersuchungen zum Wartungsaufwand ist dies interessant. Ergebnisse einer sehr alten Untersuchung könnten überholt sein, da sich die in den Projekten angewendeten Methoden grundlegend geändert haben könnten. Ein sehr kurzer Untersuchungszeitraum kann auf mangelnde Qualität einer Validierung hinweisen, da eventuell nur wenige Vergleichsdaten gesammelt werden konnten.

Die darauf folgenden Fragen nach den Untersuchungsobjekten dienen zur genauen Erfassung der Eigenschaften der untersuchten Programme, Dokumente, Projekte usw. Diese definieren einerseits die Rahmenbedingungen, unter denen die ermittelten Ergebnisse anwendbar sind. Andererseits geben sie Aufschluss darüber, ob das Untersuchungsziel überhaupt erreicht werden konnte. Sollte in einer Untersuchung z. B. ein Zusammenhang zwischen dem Alter von Programmen und der Wartbarkeit gezeigt oder ausgeschlossen werden, so müssen die untersuchten Programme auch ein breites Altersspektrum aufweisen.

Als nächstes wird die Erfassung der Vergleichsdaten für die Untersuchung überprüft, also der Daten, die z. B. verwendet wurden, um die Korrektheit der durch eine Berechnungsformel ermittelten Werte für Wartbarkeit oder Wartungsaufwand zu zeigen. Die Veröffentlichung muss einerseits die Daten an sich so genau wie möglich beschreiben, um überprüfen zu können, ob die erfassten Daten überhaupt für die beschriebene Untersuchung geeignet waren. Andererseits muss das Vorgehen zur Erfassung der Daten detailliert beschrieben sein, um dieses bei Verwendung der Modelle als Vorgehensmuster wiederholen zu können. Bei Modellen zur Ermittlung der Wartbarkeit werden in den meisten Fällen Expertenschätzungen als Vergleich herangezogen. Solche Schätzungen müssen mit hoher Sorgfalt durchgeführt werden, um als gute Grundlage für die Validierung oder Widerlegung von Modellen dienen zu können (vergleiche [Jor04]). Die Schätzungen sollten auf jeden Fall unabhängig von mehreren Personen durchgeführt werden, die auch tatsächlich Experten für die Untersuchungsobjekte sind. Weiterhin sollten die Schätzungen durch geeignete Schätzgrundlagen wie z. B. Fragebögen oder Vergleichsobjekte unterstützt und die

Experten gut auf ihre Aufgabe vorbereitet werden. Eine reine Schätzung nach Bauchgefühl von wenigen eventuell auch noch unerfahrenen Personen ist mit Sicherheit keine gute Grundlage für eine Untersuchung. Vor allem das Schätzungsziel sollte von Anfang an klar sein, da z. B. eine Schätzung die gemacht wurde um gegenüber einem Vorgesetzten gut dazustehen, kaum die Realität widerspiegeln wird. Weiterhin sollten im Rahmen einer Untersuchung, die auf Grundlage von Schätzungen geführt wurde, diese Schätzungen auch an der Realität geprüft werden. Dies kann z. B. bei Schätzungen der Wartbarkeit geschehen, in dem die an untersuchten Programmen getätigten Wartungsarbeiten mit ihrem Aufwand über einige Zeit erfasst und verglichen werden. Programme, für die ähnliche Wartbarkeit geschätzt wurde, sollten bei ähnlichen Wartungsaufgaben auch einen ähnlich hohen Aufwand produzieren. Programme mit verschiedener Wartbarkeit sollten bei ähnlichen Wartungsaufgaben auch verschieden hohe Aufwände produzieren. Bei Modellen zur Schätzung des Wartungsaufwandes werden meist realistische historische Daten zum Vergleich herangezogen. Hierbei ist darauf zu achten, dass die Daten aus zuverlässigen Quellen stammen und über einen genügend langen Zeitraum erfasst wurden. Außerdem müssen die Daten in passender Granularität erhoben worden sein. Wurden z. B. nur Aufwandsdaten für ein ganzes Softwaresystem erhoben, können daraus nicht einfach so Modelle zur Bestimmung des Wartungsaufwandes für einzelne Module entwickelt werden.

Nach der Analyse der Erfassung der Vergleichsdaten folgen Fragen zu den bei der Untersuchung verwendeten Metriken. Ziel der meisten Modelle zur Bestimmung der Wartbarkeit oder Schätzung des Wartungsaufwandes ist die Bereitstellung einer Berechnungsformel. Diese muss natürlich auf messbaren Größen basieren, die durch Prozess-, Code- und Dokument-Metriken erhoben werden können. Zur Ermittlung einer solchen Berechnungsformel werden meist mehrere Metriken untersucht und verglichen. Diese Metriken sollten natürlich zum Untersuchungsziel passen. Ist dies nicht der Fall, kann das ein Hinweis darauf sein, dass die Untersuchung künstlich groß gemacht wurde, um wichtig zu wirken. Aus diesem Grund sollte auch darauf geachtet werden, dass nicht größtenteils redundante Metriken untersucht wurden. Zur Aufstellung der Berechnungsformel wird meist eine Auswahl der untersuchten Metriken zusammengestellt, um diese weiter zu untersuchen. Die Gründe, aus denen die nicht ausgewählten Metriken verworfen wurden, sollten genau erfasst werden, um bei Verwendung der Modelle in einem anderen Projekt feststellen zu können, ob diese Gründe auch dort sinnvoll sind. Ebenso sollten Metriken erfasst werden, die in ähnlichen, vorher entwickelten Modellen häufig verwendet werden, in dieser Untersuchung aber nicht betrachtet wurden. Hierfür sollten nachvollziehbare Gründe angegeben sein, um zu zeigen, dass die Auswahl der zu untersuchenden Metriken nicht einfach spontan geschah.

Die letzten Fragen dieses Abschnittes dienen dazu, das in der Untersuchung ange-

wendete Vorgehen noch einmal Schritt für Schritt zu erfassen, um es später nachvollziehen zu können und mögliche Lücken aufzudecken. Hierbei sollte auch darauf geachtet werden, dass für die Untersuchung geeigneten Methoden verwendet wurden (vergleiche [BriLanPfl98] und [IEEE98]).

Im letzten Abschnitt zur Analyse von Untersuchungen sollen die Ergebnisse der Untersuchung zusammengefasst und mit den Untersuchungszielen verglichen werden. Hier soll auch erfasst werden, ob das Untersuchungsteam selbst Zweifel an den Ergebnissen äußert, die Gültigkeit der Ergebnisse stark einschränkt oder ähnliches.

4.1.4 Bewertung der Veröffentlichung

Der letzte Abschnitt zur Analyse von Veröffentlichungen dient zur abschließenden Bewertung einer Veröffentlichung. Hier soll zusammenfassend notiert werden, ob sich die Veröffentlichung inhaltlich mit anderen deckt, ob sie nützliche und verständlich dargestellte Informationen enthält und auf welche Ergebnisse der geschilderten Untersuchung aufgebaut werden kann oder besser nicht aufgebaut werden sollte. Außerdem ist hier Platz um weitere Auffälligkeiten der Veröffentlichung oder Bemerkungen zur Veröffentlichung zu notieren, die sonst nirgends erfasst wurden. Ich habe diesen Platz gelassen, um Bearbeiter des Fragebogens darauf hinzuweisen, dass alle Dinge, die zu einer Veröffentlichung auffallen auch notiert werden sollten, damit einmal erlangte Erkenntnisse nicht verloren gehen.

4.2 Zugehörige Veröffentlichungen – Modell

In der Tabelle zur Erfassung von Veröffentlichungen zu einem Modell sollen alle Veröffentlichungen erfasst werden, die in Zusammenhang mit dem im Titel anzugebenden Modell stehen. In der Tabelle sollen nicht nur Veröffentlichungen stehen, die schon analysiert wurden, sondern auch solche, die wahrscheinlich weitere Informationen zu dem Modell liefern können. Bei diesen Veröffentlichungen sollte auch angegeben werden, welche Informationen darin erwartet werden, um bei noch fehlenden Informationen zu Modellen schnell die richtige Veröffentlichung zur Analyse auswählen zu können. Weiterhin sollen die Veröffentlichungen nummeriert werden, um die im Fragebogen zur Analyse des Modells erfassten Informationen leicht mit den Veröffentlichungen verknüpfen zu können, aus denen sie entnommen wurden. Die zwei Spalten „eingetragen von / am“ und „ausgewertet von / am“ sollen den Informationsaustausch zwischen den Personen fördern, die an der Analyse eines Modells beteiligt sind. Um die Tabelle leicht fortführen zu können, habe ich die Seitenzahlen offen gelassen. Sie sollten auf jeden Fall von Hand eingetragen werden, um nachvollziehen zu können, ob die Tabelle vollständig vorliegt. Außerdem sollte der Name des Modells in der Fußzeile jedes Tabellenblattes eingetragen werden.

4.3 Analyse eines Modells

Dieser Teil des Fragebogens dient zur Analyse von Modellen zur Bestimmung der Wartbarkeit einer Software oder zur Schätzung des Wartungsaufwandes. Die Informationen, die hier erfasst werden, müssen natürlich aus den analysierten Veröffentlichungen entnommen werden. Ich habe versucht den gesamten Fragebogen so zu gestalten, dass möglichst wenige Informationen doppelt erfasst werden, um ihn möglichst klein zu halten. Sehr umfangreiche Fragebögen schrecken meiner Meinung nach davon ab, diese auch vollständig auszufüllen. Außerdem besteht so ein höherer Anreiz, die zu einem Modell ermittelten Informationen auch direkt in den Fragebogen zu diesem Modell einzutragen, da sie sonst nirgends erfasst werden können. Auf diese Weise sind alle wichtigen Informationen zu einem Modell auch im zugehörigen Fragebogen zu finden.

Die Fragen zu Modellen sind in vier Abschnitte gegliedert. Zu Beginn erfolgt die Analyse der Eigenschaften des Modells. Danach werden die Rahmenbedingungen erfasst, unter denen das Modell entwickelt, validiert, widerlegt oder automatisiert wurde. Zum Schluss erfolgen die Bestimmung der möglichen Verwendbarkeit des Modells und eine Zusammenfassung der Informationen zur schnellen Übersicht über das Modell.

4.3.1 Modelleigenschaften

Im Abschnitt zu den Modelleigenschaften wird zunächst der Name des Modells erfasst zusammen mit der Information, ob dieser Name von den Entwicklern vergeben wurde. Diese Information soll einen Hinweis darauf geben, dass es eventuell weitere ausgefüllte Fragebögen zu diesem Modell geben könnte, bei denen der Name etwas anders gewählt wurde. Weiterhin kann hier ein Kurzname für das Modell angegeben werden, wie z. B. „MI“ für „Maintainability Index“. Die folgenden Fragen erfassen die Art des Modells, eine Kurzbeschreibung und die Namen der erweiterten Modelle, sofern es solche gibt. Die Auswahl zur Art des Modells habe ich auf Grundlage der elf von mir analysierten Modelle erstellt (siehe Kapitel 3).

Danach soll angegeben werden, ob das Modell die Wartung von Code und/oder verschiedenen Dokumenten erfasst und für welche Projektphase das Modell anwendbar ist. Diese beiden Fragen sollen bei der Analyse der Modelle darauf aufmerksam machen, dass Wartung nicht nur für Code und auch nicht nur nach Abschluss der Entwicklung stattfindet und deshalb auch Modelle existieren müssen, die die Wartung von Dokumenten erfassen und auch während der Entwicklungsphase einsetzbar sind.

Bei der Frage nach der Granularität, in der das Modell verwendbar ist, soll erfasst werden, ob das Modell z. B. gleichermaßen für die Bestimmung der Wartbarkeit des

Gesamtsystems wie für die Bestimmung der Wartbarkeit eines einzelnen Moduls geeignet ist oder ob der Wartungsaufwand z. B. genauso gut für die Analyse von Wartungsaufgaben im nächsten Monat wie für die gesamte Wartung im nächsten Jahr geschätzt werden kann.

Der nächste große Bereich dieses Abschnitts befasst sich mit Hinweisen zur Verwendung des Modells, die in den Veröffentlichungen gegeben sind. Diese haben großen Einfluss auf die tatsächlichen Verwendungsmöglichkeiten der Modelle. Diese werden im letzten Abschnitt dieses Fragebogenteils anhand der hier gemachten Angaben und der im nächsten Abschnitt erfassten Informationen zu den Rahmenbedingungen ermittelt. Die Fragen in diesem Bereich beschäftigen sich vor allem mit Anleitungen zur Anpassung von angegebenen Berechnungsformeln an ein bestimmtes Unternehmen bzw. Projekt.

Die nächsten drei Fragen beschäftigen sich mit dem Aufwand, der bei Anwendung des Modells entsteht. Dieser ist entscheidend für eine Abschätzung, wie schnell das Modell in den Entwicklungs- und/oder Wartungsprozess integriert werden kann und in welcher Weise diese Integration sinnvoll ist. Erfordert z. B. die Abschätzung des zukünftigen Wartungsaufwandes mehrere Stunden oder gar Tage, da eventuell viele Details beachtet werden, so sollte der Wartungsaufwand nicht wöchentlich geschätzt werden, da dies mehr kostet, als es nützt. Der Berechnungsaufwand für die Wartbarkeit oder den Wartungsaufwand kann wesentlich durch die Verwendung eines Werkzeugs beeinflusst werden. Gerade für die Erhebung der Wartbarkeit kann es sinnvoll sein, die Berechnung durch ein Werkzeug in die Entwicklungsumgebung einzubinden, so dass praktisch eine ständige Verfolgung der Wartbarkeit möglich ist und die Entwickler frühzeitig auf Probleme hingewiesen werden können.

Die letzten Fragen dieses Abschnitts erfassen die Angaben zu Skalen bei Modellen zur Bestimmung der Wartbarkeit. Bei Einsatz eines solchen Modells in einem konkreten Projekt, muss geprüft werden, ob die Einteilung der Skala für das Projekt passend ist. Ist eine Software aufgrund ihrer Anforderungen sehr komplex, so ist sie von Natur aus schlechter wartbar als z. B. ein „Hello-World-Programm“. Bei einer sehr groben Skaleneinteilung ist zu erwarten, dass in solch einem Fall einfach immer eine schlechte Wartbarkeit berechnet wird. Dies ist weder für eine Verfolgung der Wartbarkeitsentwicklung sinnvoll noch für eine Entscheidung, an welchem Teil der Software ein Refactoring am ehesten nötig ist. Sind keine Grenzwerte für gute und schlechte Wartbarkeit angegeben, so müssen sie in einer Untersuchung vor Verwendung des Modells ermittelt werden. Bei angegebenen Grenzwerten ist aber eventuell auch eine Anpassung sinnvoll. In jedem Fall sollte die verwendete Skala innerhalb eines Projektes eindeutig sein. Es sollte z. B. keine Überschneidungen der Bereiche für gute, akzeptable und schlechte Wartbarkeit geben, da sonst Veränderungen der Wartbarkeit verschleiert werden könnten. Existieren solche Überschneidungen, sollte versucht werden diese sinnvoll aufzulösen.

4.3.2 Rahmenbedingungen für das Modell

In diesem Abschnitt werden die Rahmenbedingungen erfasst, unter denen das Modell validiert, widerlegt oder automatisiert wurde. Diese sind meiner Meinung nach besonders für Modelle zur Schätzung des Wartungsaufwandes wichtig, da der Wartungsaufwand von sehr vielen Faktoren abhängt, die nur schwer in einer Formel erfasst werden können. Hierzu zählen vor allem Dinge wie die Fähigkeiten des Personals, die in einem Softwareprozess angewendeten Methoden und die Komplexität des Anwendungsgebietes der Software. Für Modelle zur Bestimmung der Wartbarkeit ist die Erfassung der Rahmenbedingungen aber auch wichtig, da z. B. Modelle zur Bestimmung der Wartbarkeit von UML-Modellen schlecht auf Java-Quellcode anwendbar sind. Bei der Suche nach einem passenden Modell für ein konkretes Projekt sollten bevorzugt Modelle ermittelt werden, bei denen die für die Validierung erfassten Rahmenbedingungen möglichst genau mit den Bedingungen des Projektes übereinstimmen. Außerdem sollten solche Modelle, die für Bedingungen widerlegt wurden, die denen des Projektes sehr ähnlich sind, möglichst ausgeschlossen werden. Die Wahrscheinlichkeit, dass ein Modell richtige Ergebnisse für ein konkretes Projekt ermittelt, ist meiner Meinung nach umso höher, je besser die Rahmenbedingungen des Projektes mit denen der Modellvalidierungen übereinstimmen.

Zu den Rahmenbedingungen eines Modells gehören Eigenschaften der Software, des Personals und des Prozesses. Bei den Softwareeigenschaften sollen die Größen der Programme und Dokumente, die Anwendungsbereiche, die Programmiersprachen und Formate und das Alter der Programme und Dokumente bei Modellvalidierung, -widerlegung und -automatisierung erfasst werden. Zu den Eigenschaften des Personals gehören die Vertrautheit des Wartungspersonals mit den konkreten Programmen und Dokumenten, die gewartet werden, ihren Programmiersprachen, Formaten und Anwendungsbereichen, sowie mit der Wartungs- oder Entwicklungsumgebung. Die Vertrautheit mit diesen Dingen hat höchstwahrscheinlich großen Einfluss auf den Wartungsaufwand. Weiterhin sollten alle Kenntnisse und Fähigkeiten erfasst werden, die das Personal in den Untersuchungen hatte. Für die Verwendbarkeit eines Modells kann es entscheidend sein, ob das Personal eines konkreten Projektes ähnliche Kenntnisse und Fähigkeiten aufweist. Werden z. B. bei der Aufstellung einer projektspezifischen Berechnungsformel Kenntnisse über statistische Verfahren oder Expertenschätzungen benötigt, so müssen die für die Ermittlung der Formel zuständigen Personen entsprechend geschult sein oder werden. Bei den Prozesseigenschaften wird zunächst die für das Modell verwendete Definition von Wartung erfragt. Unterscheidet sich diese grundlegend von der in einem Projekt verwendeten, mindert dies die Verwendbarkeit des Modells für dieses Projekt, da z. B. nicht die für das Modell benötigten Daten erfasst werden können oder das Modell nicht das errechnet, von was der Anwender ausgeht. Danach sollen die Anforderungen an den

Entwicklungs- und Wartungsprozess erfasst werden, also welche Methoden und Werkzeuge bei den Untersuchungen in den Prozessen eingesetzt wurden und welche Art von Wartung betrachtet wurde. Die darauf folgende Frage dient zur genauen Beschreibung der für das Modell zu erfassenden Daten. Dies muss für den erfolgreichen Einsatz eines Modells geklärt sein, um sicherzustellen, dass der für das Projekt definierte Prozess die Erfassung dieser Daten überhaupt zulässt. Die letzte Frage zu den Prozesseigenschaften soll dazu dienen, alle weiteren Anforderungen an den Prozess zu erfassen, von denen in den Untersuchungen ausgegangen wurde. Viele Modelle gehen z. B. direkt oder indirekt von Vollzeit-Mitarbeitern aus. Wird ein Projekt größtenteils mit Teilzeit-Arbeitern bestritten, kann es vorkommen, dass die in diesen Modellen verwendeten Methoden zu Datenerfassung nicht angewendet werden können oder angepasst werden müssen.

Bei der Wahl eines passenden Modells ist natürlich auch dessen Genauigkeit entscheidend. Da ein Modell unter verschiedenen Rahmenbedingungen und bei Verwendung verschiedener Formeln meist unterschiedlich genau ist, soll mit der vorletzten Frage zu den Rahmenbedingungen die Genauigkeit der verschiedenen Formeln des Modells unter verschiedenen Bedingungen erfasst werden. Als Rahmenbedingungen sollen hier kurz alle Bedingungen aufgezählt werden, die in den vorigen Fragen abgefragt wurden und für die Modellgenauigkeit entscheidend sind.

Zum Abschluss dieses Abschnitts sollen alle Rahmenbedingungen erfasst werden, die nicht eindeutig geklärt scheinen.

Wie schon erwähnt ist die Erfassung der Rahmenbedingungen wichtig für die Auswahl eines passenden Modells. Sie sollte aber auch genutzt werden, um die Validierung von Modellen gezielt voranzutreiben. Die erfassten Rahmenbedingungen sollten Grundlage für die Auswahl passender Projekte für weitere Untersuchungen sein, um z. B. zu verhindern, dass ein Modell immer nur an sehr ähnlichen Projekten validiert wird.

4.3.3 Verwendbarkeit des Modells

In diesem Abschnitt soll geklärt werden, in welcher Weise das Modell aufgrund der durch die Veröffentlichungen gegebenen Informationen tatsächlich verwendet werden kann. Dabei sind für jedes Modell zur Bestimmung der Wartbarkeit oder zur Schätzung des Wartungsaufwandes grundsätzlich drei Verwendungsweisen denkbar. Diese sind:

1. direkte Verwendung der in einer Untersuchung entwickelten Berechnungsformel ohne Änderungen

4 Fragebogen zur Analyse von Modellen zur Wartbarkeit von Software

2. Verwendung der Struktur der entwickelten Berechnungsformel mit Anpassung von Gewichtungen und Skaleneinteilungen, und evtl. Austausch der angegebenen Metriken nach bestimmten Kriterien
3. Verwendung des im Modell angewendeten Vorgehens zur Ermittlung einer unternehmens- oder projektspezifischen Berechnungsformel

Die drei unterschiedlichen Verwendungsweisen stellen unterschiedliche Anforderungen an die durch die Veröffentlichung bereitgestellten Informationen, wie die folgende Tabelle zeigt.

Anforderung	Verwendung 1 (Formel)	Verwendung 2 (Formelstruktur)	Verwendung 3 (Vorgehen)
Formel vorgegeben	vollständige Gleichung mit Metriken und Gewichtungen, bei Wartbarkeitsmodellen zusätzlich Skala mit Grenzwerten	Formelstruktur mit möglichen Metriken	nicht nötig
Begründung zur Auswahl der Metriken	so detailliert wie möglich	so detailliert wie möglich	nicht nötig
Validierung der Zusammenhänge zwischen den Metriken und der Wartbarkeit / dem Wartungsaufwand	sehr umfangreich	sehr umfangreich	nicht nötig (Bereitstellung einer Liste mit möglichen Metriken positiv)
Validierung der Gewichtungen	sehr umfangreich	nicht nötig	nicht nötig

Anforderung	Verwendung 1 (Formel)	Verwendung 2 (Formelstruktur)	Verwendung 3 (Vorgehen)
Beschreibung des Vorgehens zum Erkennen von Zusammenhängen zwischen Metriken und der Wartbarkeit / dem Wartungsaufwand und zur Auswahl der richtigen Metriken	nicht nötig (zum Nachvollziehen der Untersuchung wünschenswert)	nicht nötig (zum Nachvollziehen der Untersuchung wünschenswert)	so detailliert wie möglich
Beschreibung des Vorgehens zum Finden geeigneter Gewichtungen (bei Modellen zur Bestimmung der Wartbarkeit zusätzlich auch Finden geeigneter Grenzwerte, sofern nicht fest vorgegeben)	nicht nötig (zum Nachvollziehen der Untersuchung wünschenswert)	so detailliert wie möglich	so detailliert wie möglich
Beschreibung der, in der Untersuchung verwendeten Vergleichsdaten	so detailliert wie möglich	so detailliert wie möglich	nicht nötig (als Beispiel evtl. hilfreich)
Beschreibung eines Vorgehens zum richtigen Erfassen der historischen Daten	nicht nötig	so detailliert wie möglich (zur Anpassung der Gewichtungen und Grenzwerte)	so detailliert wie möglich (zur Anpassung der Gewichtungen, Grenzwerte und Metriken)
Beschreibung von Umständen, die eine Anpassung der Gewichtungen (oder Grenzwerte) erfordern	nicht nötig	so detailliert wie möglich	so detailliert wie möglich

Anforderung	Verwendung 1 (Formel)	Verwendung 2 (Formelstruktur)	Verwendung 3 (Vorgehen)
Beschreibung von Umständen, die eine Neuauswahl der Metriken erfordern	nicht nötig	nicht nötig	so detailliert wie möglich
Angabe der Rahmenbedingungen, die durch Änderung der Gewichtungen (oder Grenzwerte) nicht ausgeglichen werden können (also in der Auswahl der Metriken repräsentiert sind)	nicht nötig (alle Rahmenbedingungen fest vorgegeben)	nötig	nicht nötig (keine festen Rahmenbedingungen)
Beschreibung der Untersuchungsobjekte (Programme, Dokumente, Projekte, Personal, Prozesse, Art der Wartung usw.)	so detailliert wie möglich	so detailliert wie möglich (für nicht ausgleichbare Rahmenbedingungen)	nicht nötig

Tabelle 7: Anforderungen verschiedener Verwendungsweisen an die Veröffentlichungen

Aus dieser Tabelle wurden die in diesem Abschnitt des Fragebogens zu findenden Tabellen ermittelt. Dort soll für jedes Kriterium angekreuzt werden, ob es erfüllt ist, oder nicht. Die Tabellen besitzen mit Absicht je eine Spalte für „erfüllt“ und „nicht erfüllt“, um zu vermeiden, dass aus einem eventuell einfach vergessenen Kreuz falsche Schlussfolgerungen gezogen werden. Sind viele oder gar alle Kriterien für eine Verwendungsweise erfüllt, so ist das Modell potentiell für diese Verwendungsweise geeignet. Sind nur wenige Kriterien für eine Verwendungsweise erfüllt, ist die Eignung des Modells für diese Verwendungsweise auszuschließen. Ist die direkte Verwendung der Berechnungsformel oder der Formelstruktur eines Modells möglich, so muss bei Verwendung des Modells in einem konkreten Projekt natürlich darauf geachtet werden, dass die für diese Verwendungsweisen erforderlichen Beschreibungen von Rahmenbedingungen auch zu den Bedingungen des Projektes passen.

Wird ein Modell für ein Projekt nach seiner potentiellen Verwendungsweise ausgewählt, so sollte beachtet werden, dass mit den unterschiedlichen Verwendungsweisen auch unterschiedliche Eigenschaften der Modelle und Anforderungen an den Prozess verbunden sind. Dies ist in der folgenden Tabelle dargestellt.

	Verwendung 1 (Formel)	Verwendung 2 (Formelstruktur)	Verwendung 3 (Vorgehen)
projektspezifische Anpassung möglich	nein	teilweise	vollständig
ständige Überwachung und Korrektur nötig	nur für die Rahmenbedingungen	für Rahmenbedingungen, Gewichtungen und Grenzwerte	für Rahmenbedingungen, Gewichtungen, Grenzwerte und Metriken
Aufwand zur Ermittlung einer Berechnungsformel	keiner	hoch	sehr hoch
Aufwand zur Auswahl eines passenden Modells (Prüfung der Rahmenbedingungen)	sehr hoch	hoch	gering
Zuverlässigkeit der ermittelten Werte	sehr gering bis sehr hoch (stark abhängig von der Übereinstimmung zwischen Rahmenbedingungen des Modells und Projektbedingungen)	hoch bis sehr hoch (leicht abhängig von der Übereinstimmung zwischen Rahmenbedingungen des Modells und Projektbedingungen)	sehr hoch

Tabelle 8: Eigenschaften der Modelle und Anforderungen an den Prozess aufgrund verschiedener Verwendungsweisen

Diese Tabelle zeigt deutlich, dass alle drei Verwendungsweisen Vor- und Nachteile haben. Aufgrund dieser Vor- und Nachteile ist es denkbar, dass z. B. ein Modell, das für alle drei Verwendungsweisen geeignet ist zunächst in Verwendungsweise 1 eingesetzt wird, um es schnell einführen zu können und projektspezifische historische Daten mit diesem Modell sammeln zu können. Später kann dieses Modell dann mit Hilfe der gesammelten Daten schrittweise in die Verwendungsweise 3 überführt werden, um die Genauigkeit der ermittelten Werte zu erhöhen.

4.3.4 Zusammenfassung

Im letzten Abschnitt zur Analyse von Modellen sollen zusammenfassend alle Einflussfaktoren der Wartbarkeit oder des Wartungsaufwandes, die das Modell explizit erfasst, in einer Hierarchie dargestellt werden, sofern dies möglich ist. Dies soll eine schnelle Übersicht über das Modell liefern und aufzeigen, ob das Modell mit aktuellen Untersuchungen zu Einflussfaktoren von Wartbarkeit und Wartungsaufwand vereinbar ist. Weiterhin soll hier notiert werden, ob weitere Untersuchungen für nötig gehalten werden und, wenn ja, welche Ziele diese haben sollten. Der letzte Teil dieses Abschnitts bietet Platz für offene Fragen oder sonstige Bemerkungen zum Modell.

4.4 Zugehörige Veröffentlichungen – Metrik

Die Tabelle zur Erfassung von Veröffentlichungen zu einer Metrik erfüllt die gleichen Funktionen, wie die in Kapitel 4.2 beschriebene Tabelle zur Erfassung von Veröffentlichungen zu einem Modell.

4.5 Analyse einer Metrik

Dieser Teil des Fragebogens dient zur Analyse von Metriken, die mit der Wartbarkeit oder dem Wartungsaufwand korrelieren. Ich habe diesen Teil in den Fragebogen eingefügt, da in vielen Veröffentlichungen zu Modellen darauf hingewiesen wird, dass für das Modell auch andere Metriken verwendet werden können, als die in der Veröffentlichung genannten. Um andere als die vorgeschlagenen Metriken auswählen zu können, muss man jedoch zunächst weitere Metriken kennen, für die eine für die Modelle passende Korrelation festgestellt wurde. Außerdem sollte man bei der Verwendung von Metriken, für die solche Korrelationen festgestellt wurden, auch wissen, unter welchen Rahmenbedingungen die Korrelationen festgestellt wurden, da diese natürlich genauso wie die zuvor erfassten Rahmenbedingungen des Modells zu dem Projekt passen müssen, in dem das Modell angewendet werden soll.

Die Veröffentlichungen, aus denen die in diesem Teil erfassten Informationen stammen, sollten zuvor mit dem ersten Teil des Fragebogens auf ihre Qualität hin untersucht werden. Natürlich kann es sich bei den analysierten Veröffentlichungen auch um solche handeln, die sich nicht mit einem konkreten Modell, sondern ausschließlich mit verschiedenen Metriken beschäftigen, wie z. B. [BinSch98] und [YiWuGan04].

Der Teil zur Analyse von Metriken besteht nur aus zwei Abschnitten. Der erste Abschnitt beschäftigt sich kurz mit den Eigenschaften der analysierten Metrik, der zweite mit deren Korrelationen mit Merkmalen, die im Zusammenhang mit der Wartbarkeit oder dem Wartungsaufwand stehen. Bei den Eigenschaften werden der Name der Metrik und ihre genaue Definition erfasst und überprüft, ob die Metrik objektiv und zuverlässig ist (vergleiche [IEEE98]). Bei den Korrelationen wird erfasst, für welche Merkmale von Software, Prozessen oder Personal eine Korrelation zwischen der Metrik und einem Merkmal validiert oder widerlegt wurde und unter welchen Rahmenbedingungen dies geschah.

4.6 In Modellen verwendete Metriken

Der letzte Teil des Fragebogens soll im Wesentlichen zum Vergleich der analysierten Modelle dienen. Die Modell-Metrik-Matrix in diesem Teil soll übersichtlich darstellen, welche Metriken meist in Modellen zur Bestimmung der Wartbarkeit oder zur Schätzung des Wartungsaufwandes verwendet werden. Dies soll einerseits Hinweise darauf geben, welche Metriken bevorzugt weiter validiert werden sollten, und andererseits dabei helfen passende Metriken für eine projektspezifische Anpassung eines Modells auszuwählen.

Bei diesem Teil habe ich wieder auf eine automatische Seitennummerierung verzichtet, um eine einfache Erweiterbarkeit der Matrix zu ermöglichen. Um sicherzustellen, dass immer mit der vollständigen Matrix gearbeitet wird, soll in der vorgelagerten Tabelle nach jeder Änderung die aktuelle Seitenzahl festgehalten werden. Außerdem kann dort das Datum und der Bearbeiter der letzten Änderung notiert werden, um Änderungen leicht nachverfolgen zu können.

5 Entwicklung eines Metamodells

Anhand der elf analysiert Modelle und unter Beachtung der durch den Fragebogen erfassten Informationen habe ich ein Metamodell für Modelle zur Wartbarkeit von Software entwickelt, mit dessen Hilfe alle wichtigen Informationen zu diesen Modellen übersichtlich dargestellt werden können.

Die folgenden drei Abbildungen zeigen das Metamodell in UML-Notation. Zur Verbesserung der Übersichtlichkeit habe ich alle wichtigen Informationsarten in Form eigener UML-Klassen dargestellt. Sollte das Metamodell in Datenstrukturen umgesetzt werden, reicht in einigen Fällen auch die Verwendung einfacher Attribute mit Standardtypen statt eigenständiger Klassen. Weiterhin tauchen manche Klassen in einzelnen Diagrammen mehr als einmal auf. Auch dies dient der Verbesserung der Übersichtlichkeit. Klassen mit gleichen Namen in einem Diagramm sind auch als gleiche Klassen zu betrachten.

Abbildung 8 zeigt die zur Erfassung von Untersuchungen nötigen Informationsarten und die Beziehungen zwischen Untersuchungen, Veröffentlichungen und Modellen. Eine Veröffentlichung, die mit diesem Metamodell beschrieben werden kann, berichtet immer von einer oder mehrerer Untersuchungen. Diese wiederum befassen sich mit einem oder mehreren Modellen. Die Untersuchungen werden an verschiedenen Software-Projekten, den Untersuchungsobjekten, durchgeführt und finden unter bestimmten Untersuchungsbedingungen statt. Die Untersuchungsbedingungen können in verschiedene Kategorien eingeteilt werden, die deutlich machen, wie wichtig die Einhaltung der Bedingung für die Anwendung der untersuchten Modelle ist. Die beschriebenen Untersuchungen liefern im Idealfall ein oder mehrere Untersuchungsergebnisse. Dies können Validierungen, Widerlegungen oder Automatisierungen von Modellen oder Teilen davon sein. Wurden Modelle automatisiert, so können Informationen zu den dazu entwickelten Werkzeugen erfasst werden. Zu den Veröffentlichungen, die über die Untersuchungen berichten, können natürlich alle Informationen, die mit Hilfe der Fragebögen ermittelt werden, beschrieben werden. Hierzu gehören z. B. die Autoren der Veröffentlichungen.

5 Entwicklung eines Metamodells

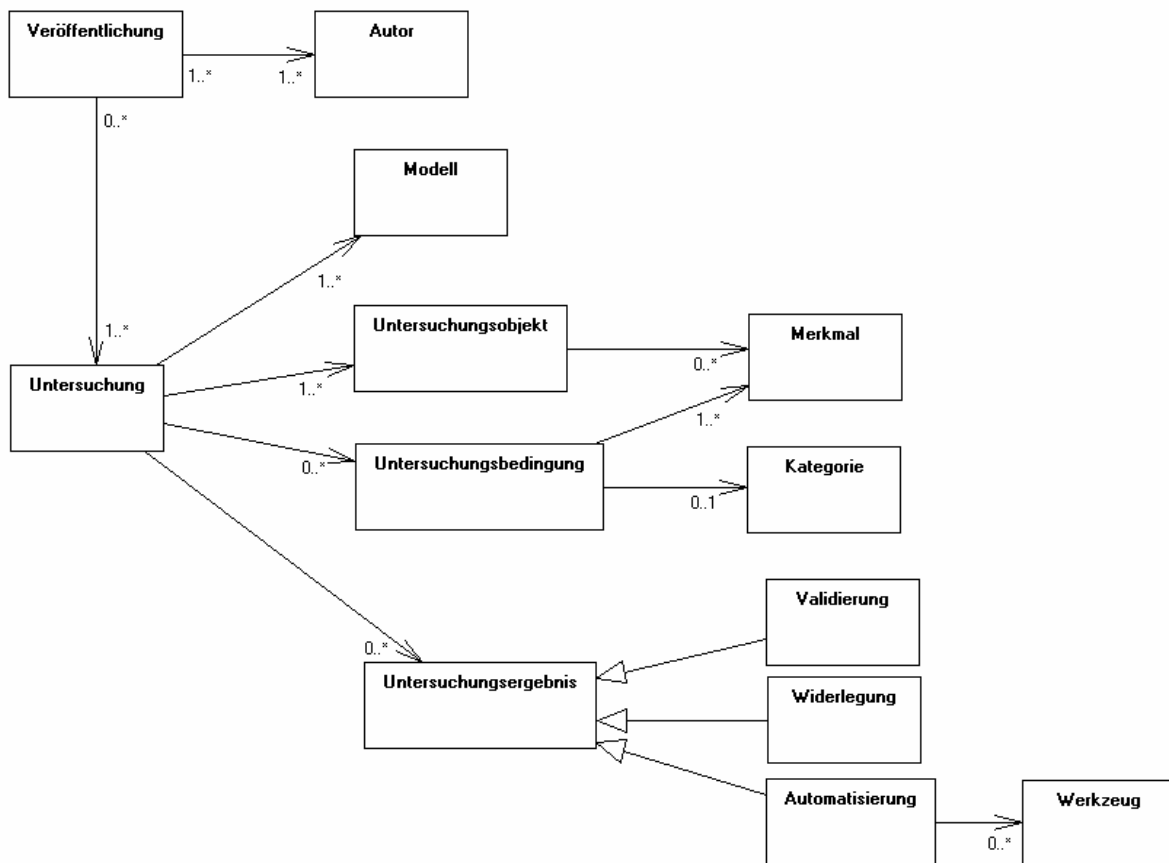


Abbildung 8: UML-Modell zur Erfassung von Untersuchungen

Das UML-Diagramm in Abbildung 9 enthält alle nötigen Informationsarten, die zur Beschreibung von Modellen wichtig sind. Die Analyse der elf in Kapitel 3 vorgestellten Modelle zeigte, dass sich Modelle zur Wartbarkeit von Software im Wesentlichen durch vier Bestandteile beschreiben lassen.

Die wichtigste Information zu einem Modell zur Wartbarkeit von Software ist das Ziel, das für dieses Modell definiert wurde. Das erstellte Metamodell dient nur zur Beschreibung von Modellen, die in irgendeiner Weise der Bestimmung der Wartbarkeit einer Software oder der Schätzung des Wartungsaufwandes für eine Software dienen. Die Ziele der Modelle sollten möglichst in eine Hierarchie gebracht werden, um die Verwaltung der Modelle zu erleichtern. Die oberste Ebene sollte nur die zwei Ziele „Bestimmung der Wartbarkeit“ und „Schätzung des Wartungsaufwandes“ unterscheiden. Danach sollten Aufgliederungen erfolgen, wie z. B. „Schätzung des Wartungsaufwandes für das nächste Release“ und „Schätzung des Wartungsaufwandes für eine bestimmte Wartungsaufgabe“.

Der zweite wichtige Bestandteil der meisten Modelle sind Annahmen über Korrelationen zwischen verschiedenen Merkmalen von Software, Prozessen und Personal mit bestimmten Stärken. Viele Modelle gehen z. B. von einer Korrelation zwischen der Wartbarkeit und der Anzahl von Codezeilen in der zu wartenden Software aus.

Der dritte wichtige Teil sind Berechnungsformeln, um z. B. die Wartbarkeit oder den Wartungsaufwand zu bestimmen. Diese Formeln bestehen meist aus verschiedenen Metriken mit Gewichtungsfaktoren und einem konstanten Anteil. Besonders bei der Berechnung der Wartbarkeit sind häufig Grenzwerte angegeben, über die die zu wartende Software verschiedenen Kategorien wie z. B. „gut wartbar“ und „schlecht wartbar“ zugeordnet werden kann. Jedes Modell kann mehrere Berechnungsformeln besitzen. Dabei können für jede Formel verschiedene Bedingungen existieren, unter denen diese Formel validiert wurde bzw. angewendet werden soll. Diese Bedingungen lassen sich über verschiedene Merkmale wie z. B. die Größe der Software beschreiben.

Weiterhin gehören Beschreibungen zu wesentlichen Vorgehen zu den Modellen. Hierzu gehören die Auswahl der passenden Metriken für ein Modell, die Bestimmung der zugehörigen Gewichtungen, die Festlegung von Kategorien mit entsprechenden Grenzwerten und schließlich die Bestimmung der Wartbarkeit oder des Wartungsaufwandes. Jedes Vorgehen wiederum lässt sich beschreiben durch den schrittweisen Ablauf, die eingesetzten Methoden und die benötigten Daten. Diese können z. B. der Verlauf des Wartungsaufwandes für einzelne Module über die letzten 12 Monate oder einfach die Größe der zu wartenden Software sein. Wurden für ein Modell in verschiedenen Untersuchungen unterschiedliche Vorgehen angewendet, so können die entsprechenden Untersuchungsbedingungen als Bedingungen für die Anwendbarkeit der Vorgehen erfasst werden.

Wurde ein Modell vollständig analysiert, so kann anhand verschiedener Kriterien festgestellt werden, in welcher Weise ein Modell verwendbar ist (siehe Kapitel 4.3.3). Auch diese Information ist durch das Metamodell darstellbar.

Das letzte UML-Diagramm (Abbildung 10) zeigt eine Übersicht über Merkmale, die zur Beschreibung der Untersuchungsobjekte, Untersuchungsbedingungen, Korrelationen, Anwendungsbedingungen und benötigten Daten aus den anderen beiden Diagrammen nötig sind.

5 Entwicklung eines Metamodells

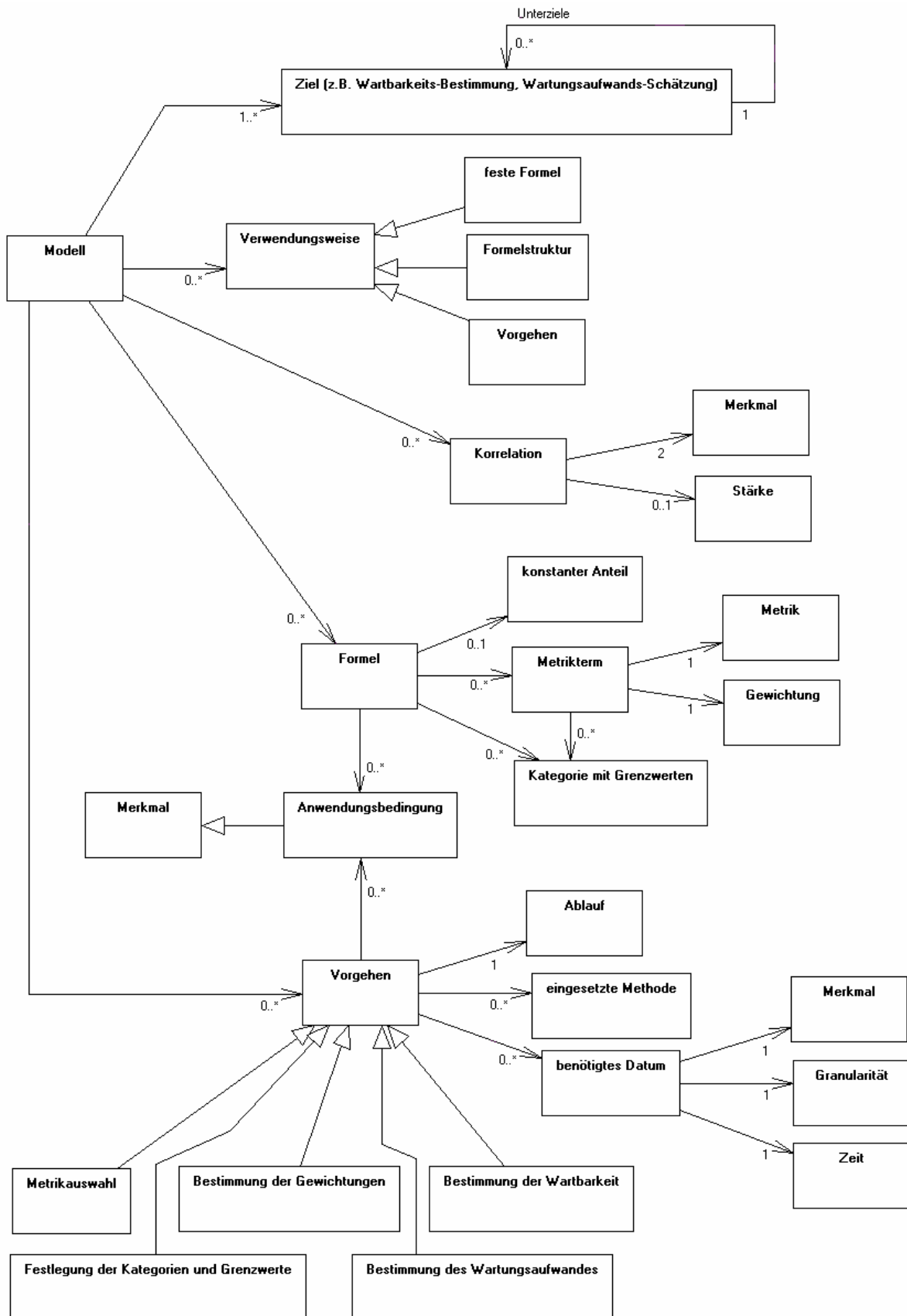


Abbildung 9: UML-Modell zur Darstellung von Modelleigenschaften

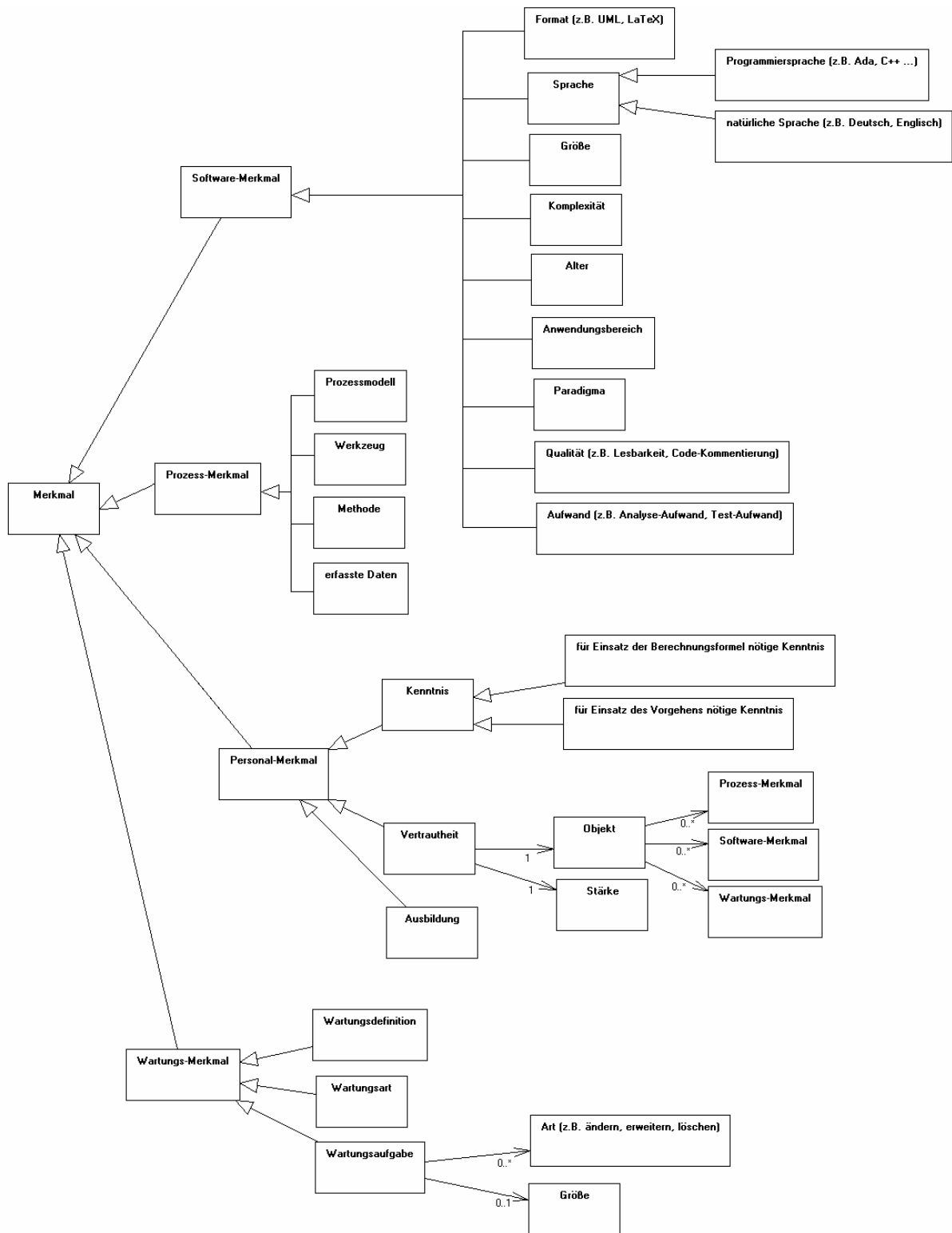


Abbildung 10: UML-Modell zur Erfassung der Merkmale von Modellen und Untersuchungen

6 Konzeption eines Systems zur Ermittlung geeigneter Modelle

In dem in Kapitel 4 beschriebenen Fragebogen, können die wesentlichen Informationen zu einem Modell erfasst werden. Diese Informationen sollen vor allem dazu dienen, zu entscheiden, ob ein bestimmtes Modell in einem konkreten Unternehmen oder Projekt anwendbar ist. Die Analyse der elf in Kapitel 3 vorgestellten Modelle mit dem Fragebogen, zeigte, dass eine Erfassung der Modelleigenschaften in Papierform schnell unbeherrschbar wird. Um einmal getätigte Modellanalysen sinnvoll verwenden zu können, wäre eine Speicherung der erfassten Daten in elektronischer Form und eine Unterstützung der Suche nach geeigneten Modellen durch eine Computeranwendung von Vorteil. Eine entsprechende Anwendung sollte folgende Aufgaben unterstützen:

- Erfassung von Informationen zu Modellen, Veröffentlichungen und Metriken
- Suche nach einem passenden Modell zur Schätzung des Wartungsaufwandes für ein Projekt mit bestimmten Rahmenbedingungen
- Suche nach einem passenden Modell zur Bestimmung der Wartbarkeit für ein Projekt mit bestimmten Rahmenbedingungen
- Suche nach einer passenden Metrik zum Einsatz in einem bestimmten Modell

In den folgenden Kapiteln möchte ich ein Konzept für ein entsprechendes System vorstellen, das Grundlage für spätere Entwicklungen sein soll. Die Entwicklung des Systems selbst ist nicht Teil dieser Diplomarbeit gewesen.

6.1 Elektronische Erfassung der Analysedaten

Der erste Schritt bei der Benutzung des hier konzipierten Systems ist natürlich die Erfassung der Informationen zu Modellen, Veröffentlichungen und Metriken. Als Spezifikationsgrundlage dieses Teils des Systems sollte der in Kapitel 4 beschriebene Fragebogen dienen. Die Möglichkeiten des Fragebogens müssen bei der Umsetzung des Systems mindestens erhalten bleiben. Hierzu gehören z. B. auch die Angabe des Datums und des Autors der letzten Änderung, eine Übersicht von Veröffentlichungen zu Modellen und Metriken, die Verknüpfung eingegebener Informationen mit den Veröffentlichungen aus denen diese stammen, eine Übersichtsmatrix zu in Modellen verwendeten Metriken, Hinweise zu einzelnen Fragen und Erläuterungen zur Verwendung des Fragebogens bzw. des Erfassungssystems. Einige dieser Informationen müssen natürlich nicht separat erfasst werden, sondern können aus anderen generiert werden.

Zur Speicherung der Daten erscheint mir die Verwendung einer Datenbank am sinnvollsten. Dies bietet vielfältige Möglichkeiten zur Kombination und Extraktion der erfassten Daten. Als Entwurfsgrundlage für die Datenverwaltung sollte das in Kapitel 5 beschriebene Metamodell herangezogen werden. Dabei ist darauf zu achten, dass an vielen Stellen des Metamodells n:m-Beziehungen auftreten. Außerdem sollte die Datenbank möglichst variabel gehalten werden, so dass z. B. jederzeit neue Fragen zu Veröffentlichungen, Modellen und Metriken eingefügt werden können.

Um einen leichten Zugriff auf das System zu gewährleisten, sollte die Benutzerschnittstelle als Webanwendung realisiert werden. So können Benutzer an verschiedenen Standorten jederzeit auf das System zugreifen, um Informationen abzufragen oder neue Informationen einzutragen. Um eine gute Qualität der erfassten Informationen zu gewährleisten, sollte bei der Umsetzung der Webanwendung über eine passende Benutzer- und Rechteverwaltung nachgedacht werden. Außerdem sollten natürlich die vielfältigen Möglichkeiten, die eine Webanwendung aufgrund von Verlinkungen und im Zusammenspiel mit einer Datenbank bietet sinnvoll genutzt werden. So sollte die Anwendung z. B. einen direkten Zugriff auf die im System erfassten Veröffentlichungen ermöglichen oder zumindest einen Link zu einer entsprechenden Veröffentlichungsstelle zur Verfügung stellen.

6.2 Konzept des Abfragesystems

Die Ermittlung eines passenden Modells für ein bestimmtes Projekt oder Unternehmen, sollte mit Hilfe eines Abfragesystems geschehen, das dem Benutzer gezielte Fragen zu den in seinem Projekt oder Unternehmen herrschenden Bedingungen stellt. Dieses System sollte natürlich ebenfalls als Webanwendung umgesetzt und mit dem Erfassungssystem kombiniert werden, um eine komfortable Benutzung des Gesamtsystems zu gewährleisten. Um das System möglichst einfach zu halten, sollten alle Fragen über Einfach- oder Mehrfachauswahlen beantwortbar sein, so dass keine Analyse eingegebener Texte nötig ist. Die dabei bereitgestellten Antwortmöglichkeiten sollten aus den in der Datenbank erfassten Daten generiert werden, so dass eine schnelle Eingrenzung auf passende Modelle möglich ist. Damit das System auch gerne benutzt wird, sollten dem Benutzer nicht unnötig viele Fragen gestellt werden. Die Reihenfolgen der Fragen sollte deshalb variabel gehalten werden, um immer solche Fragen als nächste zu stellen, deren Beantwortung mit hoher Wahrscheinlichkeit möglichst viele Modelle ausschließt. Da die Modelle teilweise sehr ungenau beschriebene Rahmenbedingungen haben oder Rahmenbedingungen fordern, die in dem konkreten Projekt zwar nicht existieren, aber leicht geschaffen werden können, sollte das System auch nicht versuchen, genau ein passendes Modell für den Benutzer zu finden. Stattdessen sollte die Befragung des Benutzers enden, wenn etwa noch 5 bis 10 Modelle übrig sind, die zum Projekt des Benutzers passen könnten. Diese

Modelle sollten dem Benutzer mit allen dazu verfügbaren Informationen präsentiert werden, so dass er selbst entscheiden kann, welches für ihn am besten anwendbar ist.

Eine mögliche Abfolge von Fragen für dieses System könnte z. B. folgende sein:

1. Welches Ziel haben Sie?

- a) Bestimmung der Wartbarkeit
- b) Schätzung des Wartungsaufwandes
- ... (weitere in der Datenbank erfasste oberste Ziele)

2. Welche Bestandteile der Software sollen betrachtet werden? (Mehrfachauswahl)

- a) Code
- b) Dokumente

2a1. Welche Paradigmen / Welche Programmiersprachen sollen betrachtet werden? (Mehrfachauswahl)

Paradigma:

Liste der Paradigmen aus der Datenbank generieren

Programmiersprache:

Liste der Programmiersprachen aus der Datenbank generieren

2a2. Wie groß ist der zu schätzende Code?

je nach noch übrigen Modellen Abfrage von Minimum, Maximum und Durchschnitt von Anzahl der Codezeilen, Anzahl der Statements und/oder Anzahl der Module

2b1. Welche Formate / Welche Sprachen soll betrachtet werden? (Mehrfachauswahl)

Format:

Liste der Formate aus der Datenbank generieren

Sprache:

Liste der Sprachen aus der Datenbank generieren

2b2. Wie groß sind die zu schätzenden Dokumente?

je nach noch übrigen Modellen Abfrage von Minimum, Maximum und Durchschnitt von Anzahl der Seiten und/oder Anzahl der Worte

3. Aus welchem Anwendungsbereich stammt die betrachtete Software?

Liste der Anwendungsbereiche aus der Datenbank generieren

4. Auf welche historischen Daten kann unter keinen Umständen zurückgegriffen werden? (Mehrfachauswahl)

Liste nötiger Daten der in der Datenbank erfassten Modelle generieren

5. Welche Wartungsdefinition trifft auf keinen Fall auf Ihre Software zu? (Mehrfachauswahl)

Liste der Wartungsdefinitionen der Modelle aus der Datenbank generieren

6. Welche Methoden setzen Sie bei der Entwicklung / Wartung ein? (Mehrfachauswahl)

Liste der eingesetzten Methoden aller Modelle aus der Datenbank generieren

7. Welche Werkzeuge setzen Sie bei der Entwicklung / Wartung ein? (Mehrfachauswahl)

Liste der eingesetzten Werkzeuge aller Modelle aus der Datenbank generieren

8. Wie vertraut ist das Wartungspersonal mit der zu wartenden Software?

Liste der möglichen Vertrautheiten mit entsprechenden Skalen aus der Datenbank generieren

6.3 Nutzen des Gesamtsystems

Das konzipierte System ermöglicht eine strukturierte Datensammlung und kann verschiedenen Zielgruppen umfassende Informationen zu Modellen bereitstellen und einen guten Überblick ermöglichen.

Für Projektmanager lässt sich leicht ein zu ihrem Projekt passendes Modell ermitteln. Die Verknüpfungen zu den zu diesem Modell analysierten Veröffentlichungen ermöglichen einen schnellen Zugriff auf alle zur Verfügung stehenden Informationen zu diesem Modell.

Für Personen, die an der Validierung von Modellen zur Wartbarkeit von Software arbeiten, lässt sich leicht ermitteln, an welcher Stelle sie noch genauere Informationen bereitstellen müssen oder welche weiteren Untersuchungen sinnvoll wären. Außerdem ergibt sich die Möglichkeit schnell einen Überblick über bereits erfolgte Arbeiten in diesem Bereich zu erhalten und eventuell Kontakte mit anderen Forschern zu knüpfen, um gemeinsam Modelle an größeren Datenmengen zu validieren.

7 Fazit und Ausblick

Während dieser Diplomarbeit ist deutlich geworden, dass sich verschiedenste Wissenschaftler schon seit einiger Zeit mit Modellen zur Wartbarkeit von Software beschäftigen. Ohne großen Aufwand fand ich etwa 100 Veröffentlichungen zu diesem Thema, weshalb ich recht schnell von einer eher unstrukturierten Suche nach Schlagwörtern wie „maintenance“ und „maintainability“ zu einer gezielten Suche nach speziellen Informationen zu ausgewählten Modellen überging. Ich machte die Erfahrung, dass es für die Analyse der vorgeschlagenen Modelle und der zugehörigen Untersuchungen und Veröffentlichungen sehr hilfreich ist, über eine Art Leitfaden zu verfügen, um die wesentlichen Informationen zu einem Modell schnell zu ermitteln. Hierzu entwickelte ich den Fragebogen (siehe Kapitel 4) und das Metamodell (siehe Kapitel 5) zu Modellen zur Wartbarkeit von Software. Beide können sehr gut verwendet werden, um alle zu einem Modell verfügbaren Informationen übersichtlich darzustellen. Häufig zeigte sich, dass die Veröffentlichungen zu den einzelnen Modellen Informationen zu wesentlichen Punkten ausließen, weshalb ich Kontakt zu den an der Validierung beteiligten Personen aufnahm. Teilweise bekam ich so weitere interessante Informationen zu den Modellen, mit denen ich meine Analysen vervollständigen konnte (siehe z. B. Kapitel 3.1 und 3.7). Insgesamt konnte ich während dieser Arbeit meine Fähigkeiten im Umgang mit wissenschaftlichen Veröffentlichungen und deren Analyse wesentlich ausbauen. Ich hoffe, dass diese Arbeit neben dem persönlichen Nutzen für mich auch einen Nutzen für die Forschungsarbeit auf dem Gebiet der Modelle zur Wartbarkeit von Software bringt. Deshalb sollten die Ergebnisse dieser Arbeit an andere Personen, die auf diesem Gebiet tätig sind weitergegeben werden. Außerdem wäre die Umsetzung des von mir in Kapitel 6 konzipierten Systems zur Erfassung und Abfrage von Informationen zu Modellen für die weitere Forschung auf diesem Gebiet sicher sehr hilfreich. Die elf von mir vorgestellten Modelle sollten in das System eingepflegt werden, da sie einen guten Überblick über den aktuellen Stand der Forschung auf diesem Gebiet geben und als guter Ausgangspunkt für weitere Untersuchungen dienen können.

Anhang A - Fragebogen

Auf den folgenden Seiten finden Sie den von mir entworfenen Fragebogen zur Analyse von Modellen zur Wartbarkeit von Software. Das Layout des Fragebogens ist so gestaltet, dass er aus diesem Anhang kopiert und direkt verwendet werden kann.

Fragebogen zur Analyse von Modellen zur Wartbarkeit von Software

Erläuterungen zum Fragebogen

Dieser Fragebogen entstand im Rahmen der Diplomarbeit „Modelle zur Wartbarkeit von Software“, die vom 14.11.2005 bis 16.05.2006 am Institut für Softwaretechnologie an der Universität Stuttgart durchgeführt wurde. Er dient zur Analyse von Modellen, die zur Bestimmung der Wartbarkeit oder zur Schätzung des Wartungsaufwandes von Software dienen. Außerdem dient er zur Analyse von Veröffentlichungen, Untersuchungen und Metriken, die mit diesen Modellen im Zusammenhang stehen. Genaue Erläuterungen zum Zweck der einzelnen Fragen dieses Fragebogens befinden sich im Bericht zur oben genannten Diplomarbeit.

Die folgenden Punkte sollten bei Verwendung dieses Fragebogens beachtet werden:

- Alle Fragen sollten so detailliert wie möglich beantwortet werden.
- Hinter einigen Fragen befinden sich eingeklammerte Zahlen wie z. B. (1), die auf ergänzende Erläuterungen zu diesen Fragen am Ende des Fragebogens hinweisen.
- Die zu einem Modell oder einer Metrik analysierten Veröffentlichungen sollten in der entsprechenden Tabelle unbedingt nummeriert werden. Diese Nummern sollten bei den einzelnen Antworten zu den Analysen von Modellen oder Metriken notiert werden, um so eine Verbindung zu den entsprechenden Veröffentlichungen zu schaffen

Veröffentlichung: _____

ausgefüllt von / am: _____

Analyse einer Veröffentlichung

1 Allgemeine Daten

Titel: _____

weitere Titel: _____

Autoren: _____

erschieden in / bei: _____

Datum der ersten Veröffentlichung: _____

beteiligte Firmen / Personen: _____

betrachtete Modelle: _____

Veröffentlichung: _____

ausgefüllt von / am: _____

betrachtete Metriken: _____

kurze Zusammenfassung: _____

Veröffentlichung: _____

ausgefüllt von / am: _____

2 Qualität der Veröffentlichung

Verwendbarkeit der Veröffentlichungen, auf die sich diese stützt:

Passen die aus den Veröffentlichungen verwendeten Daten zu der in dieser Veröffentlichung beschriebenen Untersuchung? ja | nein

Wurden verwendete Teile der Veröffentlichungen durch andere bestätigt?

ja | nein

wenn ja: Welche verwendeten Teile wurden bestätigt?: _____

Wurden verwendete Teile der Veröffentlichungen durch andere widerlegt?

ja | nein

wenn ja: Welche verwendeten Teile wurden widerlegt?: _____

Können die aus den Veröffentlichungen verwendeten Daten unter den gegebenen Rahmenbedingungen verwendet werden? ja | nein

wenn nein: Warum nicht?: _____

Veröffentlichung: _____

ausgefüllt von / am: _____

wichtige Aussagen und Schlussfolgerungen in der Veröffentlichung:

Sind nachvollziehbare Begründungen für alle Schlussfolgerungen vorhanden?

ja | nein

wenn nein: Wo fehlen Begründungen? Wo sind sie nicht nachvollziehbar?: _____

Sind gezeigte Diagramme, Tabellen u. ä. gut erkennbar? ja | nein

wenn nein: Welche nicht?: _____

Veröffentlichung: _____

ausgefüllt von / am: _____

Stimmen gezeigte Diagramme, Tabellen, Beispiele u. ä. und die Aussagen dazu überein? ja | nein

wenn nein: Wo gibt es Unstimmigkeiten?: _____

Bei Vergleich verschiedener Berechnungsformeln:

Wurden angegebene Gleichungen vollständig vereinfacht? ja | nein

wenn nein: Sind sie nach vollständiger Vereinfachung noch verschieden?

ja | nein

wenn nein: Liefern alle gleichen Gleichungen gleiche Ergebnisse? ja | nein

3 Ziel der Untersuchung, über die die Veröffentlichung berichtet

Untersuchung folgender Metriken auf Zusammenhänge mit folgenden Merkmalen (z. B. Wartbarkeit, Wartungsaufwand): _____

Validierung (von Teilen) folgender Modelle

bei Teil-Validierung: Welche Teile sollen validiert werden? (1): _____

Veröffentlichung: _____

ausgefüllt von / am: _____

Widerlegung (von Teilen) folgender Modelle

bei Teil-Widerlegung: Welche Teile sollen widerlegt werden? (1): _____

(Teil-)Automatisierung folgender Modelle

bei Teil-Automatisierung: Welche Teile sollen automatisiert werden?: _____

4 Ausführung der Untersuchung, über die die Veröffentlichung berichtet

Zeitraum (Alter + Dauer) der Untersuchung: _____

Untersuchungsobjekte:

Eigenschaften der untersuchten Programme, Dokumente, Projekte, Wartungsaufgaben etc. (Anzahl, Größe, Alter, Sprache, Formate, Anwendungsgebiet usw.; so detailliert wie möglich angeben): _____

Veröffentlichung: _____

ausgefüllt von / am: _____

Passen die ausgewählten Programme / Dokumente / Projekte zum

Untersuchungsziel? (2) ja | nein

wenn nein: Was passt nicht? _____

Wurden die Untersuchungsobjekte speziell für die Untersuchung ausgewählt?

ja | nein

wenn nein: Welches Ziel hatte die Untersuchung, für die die Untersuchungsobjekte ausgewählt wurden? _____

Aussagen zur Erfassung der Vergleichsdaten:

Wurden Experten-Schätzungen als Vergleich zugrunde gelegt? ja | nein

wenn ja:

Wie viele Personen haben geschätzt? _____

Wurden wirklich Experten für die untersuchten Programme / Dokumente

herangezogen? ja | nein

wenn nein: Wer führte die Schätzung durch? _____

Veröffentlichung: _____

ausgefüllt von / am: _____

Wurden die einzelnen Programme / Dokumente jeweils von mehreren Experten geschätzt und die Ergebnisse dann kombiniert? ja | nein

Wurden jeweils mehrere Schätzmethoden eingesetzt? ja | nein

Wurde eine Schätzgrundlage verwendet? (3) ja | nein

wenn ja: Was wurde als Grundlage verwendet? (detailliert Beschreibung):

Wurde die Schätzung speziell für diese Untersuchung durchgeführt?

ja | nein

Mit welchem Ziel wurde geschätzt? (4) _____

Wurden die Experten auf ihre Aufgabe vorbereitet? ja | nein

wenn ja: Wie?: _____

Veröffentlichung: _____

ausgefüllt von / am: _____

Wurden die Schätzungen in irgendeiner Weise an der Realität überprüft?

ja | nein

wenn ja: Wie? Mit welchem Ergebnis?: _____

Wurden realistische Vergleichsdaten erhoben? ja | nein

wenn ja: Aus welchen Quellen stammen die Daten? (5) Für welchen Zeitraum wurden Daten erhoben? Für was wurden Daten erhoben (Code, Dokumente, Wartungsaufgaben ...)?: _____

In welcher Granularität wurden Daten erhoben / Schätzungen gemacht?

a) zeitlich (z.B. monatsweise): _____

b) programm- / dokumentbezogen (z.B. methodenbezogen): _____

c) nach Aufwandsklassen (z.B. gesamte Wartung, Planung, Analyse): _____

Veröffentlichung: _____

ausgefüllt von / am: _____

**Bei Untersuchungen zu Modellen:
Aussagen zu in Modellen verwendete Metriken**

verwendete Metriken zur Validierung der Modelle

Welche Metriken wurden verwendet?: _____

Passen die verwendeten Metriken zum Untersuchungsziel? ja | nein

wenn nein: Welche Metriken passen warum nicht?: _____

Sind Metriken redundant? ja | nein

wenn ja: Welche?: _____

untersuchte Metriken, die nicht verwendet wurden

Welche Metriken wurden zwar untersucht, aber nicht verwendet? Aus welchen Gründen wurden die Metriken verworfen?: _____

Veröffentlichung: _____

ausgefüllt von / am: _____

nicht untersuchte Metriken

Welche Metriken werden in anderen (ähnlichen) Modellen häufig verwendet, wurden für diese Untersuchung aber nicht betrachtet? Warum wurden sie hier nicht untersucht?: _____

Veröffentlichung: _____

ausgefüllt von / am: _____

Welche Methoden wurden zur Validierung / Widerlegung von Zusammenhängen zwischen Metriken und untersuchten Merkmalen verwendet?: _____

einzelne Schritte des Vorgehens / der angewendeten Methodik: _____

Veröffentlichung: _____

ausgefüllt von / am: _____

Erscheint die Beschreibung des Vorgehens vollständig? ja | nein

wenn nein: Wo sind Lücken? _____

Bei Untersuchungen zu Modellen:

Sind alle zur Validierung / Widerlegung der Bestandteile des Modells verwendeten Schritte klar nachvollziehbar? ja | nein

wenn nein: Welche Schritte sind unklar? Wo fehlen Erläuterungen oder Beispiele?

Welche Schritte sind methodisch zweifelhaft? _____

Veröffentlichung: _____

ausgefüllt von / am: _____

6 Bewertung der Veröffentlichung

Deckt sich die Veröffentlichung inhaltlich mit einer anderen

Veröffentlichung? ja | nein

wenn ja: Autor(en), Titel, Jahr: _____

Enthält die Veröffentlichung nützliche Informationen? ja | nein

Sind alle nötigen Informationen verständlich dargestellt? ja | nein

**Auf welche Ergebnisse der geschilderten Untersuchungen kann
aufgebaut werden?** _____

**Auf welche Ergebnisse der geschilderten Untersuchungen sollte NICHT
aufgebaut werden? Warum?** _____

Zugehörige Veröffentlichungen - Modell: _____

Nr.	Autoren, Titel, Jahr, erschienen in / bei	(erwarteter) Inhalt (bezogen auf das Modell)	eingetragen von / am	ausgewertet von / am

Nr.	Autoren, Titel, Jahr, erschieden in / bei	(erwarteter) Inhalt (bezogen auf das Modell)	eingetragen von / am	ausgewertet von / am

Zugehörige Veröffentlichungen - Modell: _____

Modell: _____

ausgefüllt von / am: _____

Analyse eines Modells (6)

1 Modelleigenschaften

Name/Kurzname: _____

Name durch Entwickler des Modells vergeben? ja | nein

Art des Modells:

Modell zur Bestimmung der Wartbarkeit

Modell zur Schätzung des Wartungsaufwandes

für eine konkrete Wartungsaufgabe

für das nächste Release

für den restlichen Lebenszyklus

für einen festen Zeitraum: _____ (7)

in Abhängigkeit der Zeit (z.B. 3 Personen-Monate/Jahr)

in Abhängigkeit von LOC oder Statements

Sonstiges: _____

Kurzbeschreibung: _____

Namen von erweiterten Modelle: _____

Modell: _____

ausgefüllt von / am: _____

Wartung von was wird betrachtet / kann betrachtet werden?:

Code

Dokumente (8): _____

betrachtete Phase im Lebenszyklus:

Wartung während der Entwicklung

Wartung nach der Entwicklung

In welcher Granularität ist das Modell wie gut verwendbar?:

a) zeitlich (z.B. monatsweise): _____

b) programm- / dokumentbezogen (z.B. methodenbezogen): _____

c) nach Aufwandsklassen (z.B. gesamte Wartung, Planung, Analyse): _____

in den Veröffentlichungen angegebene Hinweise zur Verwendung des Modells:

Sind Berechnungsformeln / Formelstrukturen angegeben? ja | nein

wenn ja: Wie lauten sie?: _____

Modell: _____

ausgefüllt von / am: _____

Sind alle Berechnungsformeln / Formelstrukturen gleich gut? ja | nein

wenn nein: Wann ist welche Formel(-Struktur) am besten anwendbar?: _____

Gibt es eine Anleitung oder ein Beispiel für die Anpassung (von Teilen) des Modells an das Projekt / Unternehmen oder andere Sprachen (Programmiersprachen, natürliche Sprachen) und Formate (besonders bei Dokumenten)?

ja | nein

wenn ja: Anleitung / Beispiel für Anpassung: _____

Modell: _____

ausgefüllt von / am: _____

Welche Daten müssen über welchen Zeitraum für die Anpassung bereitstehen?:

Sind verwendbare Metriken angegeben?: ja | nein

wenn ja: Welche Metriken sind angegeben?: _____

Modell: _____
ausgefüllt von / am: _____

Wurden alle angegebenen Metriken auf ihren Zusammenhang mit dem
Wartungsaufwand / der Wartbarkeit (oder einem Einflussfaktor davon) validiert?:

ja | nein

Wenn nein: Welche nicht? _____

Wie können passende Metriken ausgewählt werden? _____

Wann / unter welchen Umständen müssen Neuanpassungen vorgenommen
werden (neue Metriken, neue Gewichtungen usw.)?: _____

Modell: _____

ausgefüllt von / am: _____

Welche Überprüfungen an der Software müssen durchgeführt werden? (9)

Automatisierung

Ist eine Automatisierung möglich?: ja | nein

wenn ja:

Für welche Teile des Modells ist eine Automatisierung möglich? Für welche existieren bereits entsprechende Werkzeuge? Welche sind dies? _____

Aufwand

	sehr hoch	hoch	mittel	niedrig	sehr niedrig
Aufwand für Vorbereitung des Modelleinsatzes (10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aufwand für die Durchführung der Vorgehen des Modells (11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aufwand für die reine Erhebung	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Modell: _____

ausgefüllt von / am: _____

Welche Erhebungshäufigkeit lässt der Aufwand (evtl. mit Automatisierung) zu?

praktisch ständig

mehrmals täglich

einmal täglich

mehrmals wöchentlich

einmal wöchentlich

mehrmals monatlich

einmal monatlich

Sonstiges: _____

bei Modellen zur Bestimmung der Wartbarkeit:

Ist eine Skala für die Wartbarkeit angegeben? ja | nein

wenn ja:

Wo liegt das Minimum, wo das Maximum der Skala? _____

Welcher Detailgrad ist möglich? _____

Welche Grenzwerte sind angegeben (für gute und schlechte Wartbarkeit usw.)? _____

Ist die Skaleneinteilung eindeutig oder gibt es Überschneidungen oder Widersprüche? _____

Modell: _____

ausgefüllt von / am: _____

2 Rahmenbedingungen für das Modell

Eigenschaften der Software:

- a) Größe der Programme / Dokumente (Minimum, Maximum, Durchschnitt)
- b) Anwendungsbereiche
- c) Programmiersprachen / Formate
- d) Alter der Programme / Dokumente (Minimum, Maximum, Durchschnitt)

bei Modell-Validierung: _____

bei Modell-Widerlegung: _____

bei Modell-Automatisierung: _____

Modell: _____

ausgefüllt von / am: _____

Eigenschaften des Personals:

Wie vertraut war das Wartungspersonal in den Untersuchungen mit:

	sehr gut	gut	nor- mal	kaum	gar nicht
den konkreten Programmen / Dokumenten	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
den Programmiersprachen / Formaten	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
den Anwendungsbereichen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
der Wartungs- / Entwicklungsumgebung	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Welche Kenntnisse werden zur Verwendung des Modells gebraucht (z. B. Kenntnisse über Regressionsanalyse)? _____

Welche Fähigkeiten hatte das Personal in den Untersuchungen?

bei Modell-Validierung: _____

bei Modell-Widerlegung: _____

Modell: _____

ausgefüllt von / am: _____

bei Modell-Automatisierung: _____

Welche Fähigkeiten werden vom Entwicklungs-/Wartungspersonal auf jeden Fall erwartet?: _____

Prozesseigenschaften:

Wie wird Wartung für das Modell definiert?: _____

Anforderungen an den Entwicklungs- / Wartungsprozess

- a) eingesetzte Methoden
- b) eingesetzte Werkzeuge
- c) Art der Wartung (adaptierend, korrigierend, perfektionierend, sonstige)
- d) Phase im Lebenszyklus (Wartung während der Entwicklung, Wartung nach der Entwicklung)

bei Modell-Validierung: _____

Modell: _____

ausgefüllt von / am: _____

bei Modell-Widerlegung: _____

bei Modell-Automatisierung: _____

für die reine Bestimmung der Wartbarkeit / des Wartungsaufwandes benötigte Daten (12):

- a) Über was werden Daten benötigt?
- b) Für was werden die Daten benötigt?
- c) Wie weit müssen die Daten zurückgehen?
- d) Wie detailliert müssen die Daten sein?

Modell: _____

ausgefüllt von / am: _____

Von welchen Voraussetzungen, wird bei dem Modell (direkt oder indirekt) ausgegangen (z. B. Vollzeit-Arbeiter)? Wie können diese geschaffen oder ausgeglichen werden, wenn sie nicht gegeben sind?: _____

Modell: _____

ausgefüllt von / am: _____

Genauigkeit des Modells unter verschiedenen Rahmenbedingungen oder bei Verwendung verschiedener Formeln:

Genauigkeit	Rahmenbedingungen / Formel

Modell: _____

ausgefüllt von / am: _____

Genauigkeit	Rahmenbedingungen / Formel

Sind alle Rahmenbedingungen geklärt?: ja | nein

wenn nein: Welche nicht?: _____

Modell: _____

ausgefüllt von / am: _____

3 Verwendbarkeit des Modells

notwendige Kriterien (13)

1. für direkte Verwendung der entwickelten Berechnungsformel (ohne Anpassung von Gewichtungen oder verwendeten Metriken)

	erfüllt	nicht erfüllt
vollständige Formel mit Metriken und Gewichtungen angegeben	<input type="checkbox"/>	<input type="checkbox"/>
bei Modellen zur Bestimmung der Wartbarkeit zusätzlich: vollständig definierte Skala mit Grenzwerten vorgegeben	<input type="checkbox"/>	<input type="checkbox"/>
genau nachvollziehbare Begründung zur Auswahl der Metriken	<input type="checkbox"/>	<input type="checkbox"/>
sehr umfangreiche Validierung der Zusammenhänge zwischen den gewählten Metriken und der Wartbarkeit / dem Wartungsaufwand	<input type="checkbox"/>	<input type="checkbox"/>
sehr umfangreiche Validierung der Gewichtungen	<input type="checkbox"/>	<input type="checkbox"/>
detaillierte Beschreibung der in den Untersuchungen verwendeten Vergleichsdaten (historische Daten oder Schätzungen)	<input type="checkbox"/>	<input type="checkbox"/>
detaillierte Beschreibung der Untersuchungsobjekte (Programme, Dokumente, Projekte, Personal, Art der Wartung usw.)	<input type="checkbox"/>	<input type="checkbox"/>
gesamt	von 6	von 6

2. für Verwendung der Struktur der entwickelten Berechnungsformel (mit Anpassung von Gewichtungen und evtl. Austausch der angegebenen Metriken nach bestimmten Kriterien)

	erfüllt	nicht erfüllt
Formelstruktur mit Metriken vorgegeben	<input type="checkbox"/>	<input type="checkbox"/>
genau nachvollziehbare Begründung zur Auswahl der Metriken	<input type="checkbox"/>	<input type="checkbox"/>
umfangreiche Validierung der Zusammenhänge zwischen den gewählten Metriken und der Wartbarkeit / dem Wartungsaufwand	<input type="checkbox"/>	<input type="checkbox"/>
detaillierte Beschreibung des Vorgehens zum Finden geeigneter Gewichtungen (bei Modellen zur Bestimmung der Wartbarkeit zusätzlich auch zum Finden geeigneter Grenzwerte, sofern nicht fest vorgegeben)	<input type="checkbox"/>	<input type="checkbox"/>

Modell: _____

ausgefüllt von / am: _____

detaillierte Beschreibung der in den Untersuchungen verwendeten Vergleichsdaten (historische Daten oder Schätzungen)	<input type="checkbox"/>	<input type="checkbox"/>
detaillierte Beschreibung eines Vorgehens zum richtigen Erfassen von historischen Daten zur evtl. nötigen Anpassung der Gewichtungen und Grenzwerte	<input type="checkbox"/>	<input type="checkbox"/>
detaillierte Beschreibung der Umstände, die eine Anpassung der Gewichtungen oder Grenzwerte erfordern	<input type="checkbox"/>	<input type="checkbox"/>
Angabe der Rahmenbedingungen, die durch Änderung der Gewichtungen oder Grenzwerte nicht ausgeglichen werden können (also in der Auswahl der Metriken repräsentiert sind)	<input type="checkbox"/>	<input type="checkbox"/>
detaillierte Beschreibung der Untersuchungsobjekte (Programme, Dokumente, Projekte, Personal, Art der Wartung usw.) sofern die Eigenschaften der Objekte nicht ausgleichbare Rahmenbedingungen sind	<input type="checkbox"/>	<input type="checkbox"/>
gesamt	von 9	von 9

3. für Verwendung des im Modell angewendeten Vorgehens zur Ermittlung einer unternehmens- oder projektspezifischen Berechnungsformel

	erfüllt	nicht erfüllt
detaillierte Beschreibung des Vorgehens zum Erkennen von Zusammenhängen zwischen Metriken und der Wartbarkeit / dem Wartungsaufwand und zur Auswahl der richtigen Metriken	<input type="checkbox"/>	<input type="checkbox"/>
detaillierte Beschreibung des Vorgehens zum Finden geeigneter Gewichtungen (bei Modellen zur Bestimmung der Wartbarkeit zusätzlich auch zum Finden geeigneter Grenzwerte, sofern nicht fest vorgegeben)	<input type="checkbox"/>	<input type="checkbox"/>
detaillierte Beschreibung eines Vorgehens zum richtigen Erfassen von historischen Daten zur evtl. nötigen Anpassung der Gewichtungen, Grenzwerte und Metriken	<input type="checkbox"/>	<input type="checkbox"/>
detaillierte Beschreibung der Umstände, die eine Anpassung der Gewichtungen oder Grenzwerte erfordern	<input type="checkbox"/>	<input type="checkbox"/>
detaillierte Beschreibung der Umstände, die eine Neuauswahl der Metriken erfordern	<input type="checkbox"/>	<input type="checkbox"/>
gesamt	von 5	von 5

Modell: _____

ausgefüllt von / am: _____

ermittelte Verwendungsmöglichkeit(en):

- direkte Verwendung der entwickelten Berechnungsformel
- Verwendung der Struktur der entwickelten Berechnungsformel
- Verwendung des Vorgehens zur Ermittlung der Berechnungsformel

4 Zusammenfassung

Welche Einflussfaktoren, werden durch die im Modell verwendeten Metriken und ermittelten Voraussetzungen oder Rahmenbedingungen explizit erfasst? Wie stehen sie hierarchisch zueinander? Wie werden sie aufgegliedert? (14): _____

Sind weitere Untersuchungen (z. B. zur weiteren Validierung) nötig?:

ja | nein

wenn ja: Welche konkreten Inhalte / Ziele sollten die Untersuchungen haben?:

Modell: _____

ausgefüllt von / am: _____

offene Fragen / Sonstiges: _____

Zugehörige Veröffentlichungen - Metrik: _____

Nr.	Autoren, Titel, Jahr, erschienen in / bei	(erwarteter) Inhalt (bezogen auf die Metrik)	eingetragen von / am	ausgewertet von /am

Nr.	Autoren, Titel, Jahr, erschieden in / bei	(erwarteter) Inhalt (bezogen auf die Metrik)	eingetragen von / am	ausgewertet von /am

Zugehörige Veröffentlichungen - Metrik: _____

Metrik: _____

ausgefüllt von / am: _____

Analyse einer Metrik

1 Metrikeigenschaften

Name/Kurzname: _____

Definition (Was misst die Metrik? Auf welcher Skala liegen die Messwerte? Was bedeutet eine Änderung des Metrikwertes? Wie können die Metrikwerte erhoben werden?): _____

Ist die Metrik objektiv? (Sind subjektive Einflüsse des Messenden ausgeschlossen?) ja | nein

Ist die Metrik zuverlässig? (Ergibt eine Wiederholung der Messung unter gleichen Umständen das gleiche Ergebnis?) ja | nein

Metrik: _____

ausgefüllt von / am: _____

2 Korrelationen

Für welche Merkmale von Software, Prozessen oder Personal wurde eine Korrelation zwischen der Metrik und dem Merkmal festgestellt?:

Wartungsaufwand

Wartbarkeit

Sonstiges: _____

Für welche Merkmale von Software, Prozessen oder Personal wurde eine Korrelation zwischen der Metrik und dem Merkmal widerlegt?:

Wartungsaufwand

Wartbarkeit

Sonstiges: _____

Metrik: _____

ausgefüllt von / am: _____

Validierung von Korrelationen:

Merkmal	Korrelation mit welchem Merkmal wurde validiert für was für Programme / Dokumente (Anzahl, Sprachen, Formate, Größen, Alter usw.)? Welche Methode wurde dabei eingesetzt? Wie stark war die Korrelation?

Metrik: _____

ausgefüllt von / am: _____

Merkmal	Korrelation mit welchem Merkmal wurde validiert für was für Programme / Dokumente (Anzahl, Sprachen, Formate, Größen, Alter usw.)? Welche Methode wurde dabei eingesetzt? Wie stark war die Korrelation?

Metrik: _____

ausgefüllt von / am: _____

Merkmal	Korrelation mit welchem Merkmal wurde validiert für was für Programme / Dokumente (Anzahl, Sprachen, Formate, Größen, Alter usw.)? Welche Methode wurde dabei eingesetzt? Wie stark war die Korrelation?

Metrik: _____

ausgefüllt von / am: _____

Widerlegung von Korrelationen:

Merkmal	Korrelation mit welchem Merkmal wurde widerlegt für was für Programme / Dokumente (Anzahl, Sprachen, Formate, Größen, Alter usw.)? Welche Methode wurde dabei eingesetzt?

Metrik: _____

ausgefüllt von / am: _____

Merkmal	Korrelation mit welchem Merkmal wurde widerlegt für was für Programme / Dokumente (Anzahl, Sprachen, Formate, Größen, Alter usw.)? Welche Methode wurde dabei eingesetzt?

Metrik: _____

ausgefüllt von / am: _____

Merkmal	Korrelation mit welchem Merkmal wurde widerlegt für was für Programme / Dokumente (Anzahl, Sprachen, Formate, Größen, Alter usw.)? Welche Methode wurde dabei eingesetzt?

In Modellen verwendete Metriken

zuletzt aktualisiert von	Datum	#Seiten	zuletzt aktualisiert von	Datum	#Seiten

Modell																																					
Metrik																																					

Hinweise zu einzelnen Fragen

- (1) Teil-Validierungen oder –Widerlegungen können sich z. B. auf Zusammenhänge zwischen Metriken und bestimmten Merkmalen einer Software oder eines Prozesses oder auf die Anwendbarkeit einer Berechnungsformel auf eine bestimmte Programmiersprache beziehen.
- (2) Wenn z. B. der Zusammenhang zwischen dem Alter der Software und dem Wartungsaufwand untersucht werden soll, dann muss auch verschieden alte Software untersucht werden.
- (3) Schätzgrundlagen können z. B. Vergleichssoftware, Vergleichstabellen, Fragebögen oder Checklisten für einzelne Kriterien sein. Sie geben dem Schätzer Anhaltspunkte dafür, was bezüglich dieser Untersuchung z. B. gut oder schlecht wartbar ist. Durch Verwendung von Schätzgrundlagen wird die Vergleichbarkeit von Schätzungen erhöht.
- (4) Wurde die Schätzung explizit für diese Studie durchgeführt? War sie ehrlich und risikofrei für den Schätzer? Oder wurde die Schätzung vielleicht durch den Willen, das eigene Unternehmen gut darzustellen oder eine gute Bewertung für eine selbst entwickelte Software zu bekommen, beeinflusst?
- (5) In Unternehmen werden häufig Arbeitsaufwände für bestimmte Projekte nicht auf diese Projekte abgerechnet, um für diese Projekte eine gute Bilanz zu erhalten. Bei einer Studie, die Zusammenhänge zwischen dem Aufwand und anderen Faktoren untersucht, dürfen natürlich keine so verrechneten Aufwandsdaten verwendet werden, sonst setzen die Ergebnisse der Studie voraus, dass die Verrechnung der Aufwände zwischen den Projekten immer genau so geschieht, wie es bei den untersuchten Daten der Fall war.
- (6) Zu jedem Modell soll dieser Teil des Fragebogens nur einmal ausgefüllt werden, um so die Informationen aus allen zugehörigen Veröffentlichungen zusammenzufassen.
- (7) Bitte geben Sie hier auch den konkreten Zeitraum an.
- (8) Listen Sie die betrachteten Dokumente hier auf. Falls praktisch alle Dokumente eines Projektes in die Betrachtung mit einbezogen werden, notieren Sie „alle“.
- (9) Muss z. B. die Qualität der Kommentare oder externer Dokumentation analysiert werden?
- (10) Hierzu gehören z. B. Aufwände zur Einführung neuer Werkzeuge oder Methoden zur Datenerfassung und Datensammlung.
- (11) Hierzu gehören z. B. Aufwände für die Analyse historischer Daten zur Ermittlung passender Metriken oder Gewichtungen, sowie zur Überwachung von Daten für evtl. Anpassungen.

- (12) Geben Sie hier sowohl explizit verlangte Daten als auch implizit vorausgesetzte Daten wie z. B. eine Abschätzung der Anzahl der hinzuzufügenden, zulöschenden und zu ändernden Codezeilen an.
- (13) Bitte kreuzen Sie „erfüllt“ oder „nicht erfüllt“ nur an, wenn dies eindeutig geklärt ist. Lassen Sie sonst die gesamte Zeile frei und suchen Sie nach Klärung.
- (14) Beispiel:
Wartbarkeit → Lesbarkeit des Quellcodes → LOC/LOM (Kommentare)
→ Verständlichkeit der Software → Anzahl der Symbole
→ Dokumentationsqualität → Wörter/Satz
→ Anteil Wörter mit > 3 Silben

Literatur

- [AggSinCha05] Aggarwal, K. K.; Singh, Y.; Chandra, P.; Puri, M.: Measurement of Software Maintainability Using a Fuzzy Model. In: Journal of Computer Sciences, Band 1 Heft 4, Science Publications, ISSN 1549-3636, S. 537-541, 2005
- [AggSinChh02] Aggarwal, K. K.; Singh, Y.; Chhabra, J. K.: An Integrated Measure of Software Maintainability. In: Proceedings Annual Reliability and Maintainability Symposium, Seattle, WA, USA, ISBN 0-7803-7348-0, S. 235-241, 2002
- [AhnSuhKim03] Ahn, Y.; Suh, J.; Kim, S.; Kim, H.: The software maintenance project effort estimation model based on function points. In: Journal of Software Maintenance, Band 15 Heft 2, S. 71-85, März 2003
- [BasBriCon96] Basili, V. R.; Briand, L.; Condon, S.; Kim, Y.; Melo, W. L.; Valett, J. D.: Understanding and Predicting the Process of Software Maintenance Releases. In: Proceedings of the 18th International Conference on Software Engineering (Berlin, Deutschland). International Conference on Software Engineering (ICSE'96). IEEE Computer Society, Washington, DC, USA, S. 464-474, März 1996
- [BasRom88] Basili, V. R.; Rombach, D.: The TAME Project: Towards Improvement-Oriented Software Environments. In: IEEE Transactions on Software Engineering, Band 14 Heft 6, S. 758-773, Juni 1988
- [BinSch98] Binkley, A. B.; Schach, S. R.: Validation of the Coupling Dependency Metric as a Predictor of Run-Time Failures and Maintenance Measures. In: Proceedings of the 20th International Conference on Software Engineering (Kyoto, Japan). International Conference on Software Engineering (ICSE'98). IEEE Computer Society, Washington, DC, USA, S. 452-455, April 1998
- [BoeBroKas78] Boehm, B. W.; Brown, J. R.; Kaspar, H.; Lipow, M.; MacLeod, G. J.; Merritt, M. J.: Characteristics of Software Quality, North-Holland Publishing Company, 1978
- [BriLanPfl98] Briand, L.; Lanubile, F.; Pfleeger, S. L.; Rothermel, G.; Schneidewind, N.: Empirical Studies of Software Maintenance: A Report from WESS '97. In: Empirical Software Engineering, Band 3 Heft 3, S. 299-307, September 1998
- [ColAshLow94] Coleman, D.; Ash, D.; Lowther, B.; Oman, P.: Using Metrics to Evaluate Software System Maintainability. In: Computer, Band 27 Heft 8, S. 44-49, August 1994

- [Fio01] Fioravanti, F.: A Metric Framework for the Assessment of Object-Oriented Systems. In: Proceedings of the IEEE International Conference on Software Maintenance (ICSM'01). IEEE Computer Society, Washington, DC, USA, S. 557-560, November 2001
- [FioNes01] Fioravanti, F.; Nesi, P.: Estimation and Prediction Metrics for Adaptive Maintenance Effort of Object-Oriented Systems. In: IEEE Transactions on Software Engineering, Band 27 Heft 12, S. 1062-1084, Dezember 2001
- [FioNesPer98a] Fioravanti, F.; Nesi, P.; Perlini, S.: A Tool for Process and Product Assessment of C++ Applications. In: Proceedings of the 2nd Euromicro Conference on Software Maintenance and Reengineering (CSMR'98). IEEE Computer Society, Washington, DC, USA, S. 89-95, März 1998
- [FioNesPer98b] Fioravanti, F.; Nesi, P.; Perlini, S.: Assessment of system evolution through characterization. In: Proceedings of the 20th International Conference on Software Engineering (Kyoto, Japan). International Conference on Software Engineering (ICSE'98). IEEE Computer Society, Washington, DC, USA, 456-459, April 1998
- [FioNesSto99] Fioravanti, F.; Nesi, P.; Stortoni, F.: Metrics for Controlling Effort during adaptive Maintenance of Object-Oriented Systems. In: Proceedings of the IEEE International Conference on Software Maintenance (ICSM'99). IEEE Computer Society, Washington, DC, USA, S. 483-492, August/September 1999
- [ForGroRos97] Foreman, J.; Gross, J.; Rosenstein, R.; Fisher, D.; Brune, K.: C4 Software Technology Reference Guide - A Prototype. CMU/SEI-97-HB-001. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 1997
- [HayPatZha04] Hayes, J. H.; Patel, S. C.; Zhao, L.: A Metrics-Based Software Maintenance Effort Model. In: Proceedings of the 8th Euromicro Working Conference on Software Maintenance and Reengineering (CSMR'04). IEEE Computer Society, Washington, DC, USA, 254-258, März 2004
- [HayZha05] Hayes, J. H.; Zhao, L.: Maintainability Prediction: A Regression Analysis of Measures of Evolving Systems. In: Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM'05) - Volume 00, IEEE Computer Society, Washington, DC, USA, S. 601-604, September 2005
- [IEEE90] IEEE Std. 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology. The Institute of Electrical and Electronics Engineers, New York, NY, USA, September 1990

- [IEEE98] IEEE Std. 1061-1998: IEEE Standard for a Software Quality Metrics Methodology. The Institute of Electrical and Electronics Engineers, New York, NY, USA, Dezember 1998
- [IFPUG99] IFPUG: Enhancement project function point calculation. Function Point Counting Practices Manual (Release 4.1). International Function Point Users Group, Mequon, WI, USA, 1999
- [ISO01] International Standard ISO/IEC 9126: Software engineering - Product quality. 2001
- [ISO95] International Standard ISO/IEC 12207: Information technology – Software life cycle processes. 1995
- [ISO99] International Standard ISO/IEC 14764: Information technology – Software maintenance. 1999
- [Jor04] Jørgensen, M.: A Review of Studies on Expert Estimation of Software Development Effort. In: Journal of Systems and Software, Elsevier Science, Band 70 Heft 1-2, 37-60, 2004
- [Kaj03] Kajko-Mattsson, M.: Problem Management with Corrective Maintenance. In: Advances in Software Maintenance Management: Technologies and Solutions. Polo, M.; Piattini, M.; Ruiz, F. (Hrsg.), Idea Group Publishing, September 2003
- [KieJinMue04] Kiewkanya, M.; Jindasawat, N.; Muenchaisri, P.: A Methodology for Constructing Maintainability Model of Object-Oriented Design. In: Proceedings of the Quality Software, 4th International Conference on (QSIC'04) – Volume 00. IEEE Computer Society, Washington, DC, USA, S. 206-213, September 2004
- [Lai96] Laitnen K.: Estimating Understandability of Software Documents. In: ACM SIGSOFT, Band 21, S. 81-92, Juli 1996
- [Leh01] Lehman, M. M.: FEAST/2 Final Report - Grant Number GR/M44101, Department of Computing, Imperial College, London, September 2001
- [Lis01] Liso, A.: Software Maintainability Metrics Model: An Improvement in the Coleman-Oman Model. Crosstalk. In: Journal of Defense Software Engineering, 15-17, August 2001

Literatur

- [LucPomSte02] De Lucia, A.; Pompella, E.; Stefanucci, S.: Effort Estimation for Corrective Software Maintenance. In: Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (Ischia, Italien). SEKE'02, Band 27, ACM Press, New York, NY, USA, S. 409-416, Juli 2002
- [Lud98] Ludewig, J.: Software Engineering – vorläufiges, unvollständiges Skript zur Vorlesung Software Engineering in der Fakultät Informatik der Universität Stuttgart. Kapitel 6 V2.1 (97-04-30), 1998
- [LudOpf04] Ludewig, J.; Opferkuch, S.: Software-Wartung - eine Taxonomie. In: Softwaretechnik-Trends, Band 24 Heft 2, Gesellschaft für Informatik, ISSN 0720-8928, Mai 2004
- [MutPonKon00] Muthanna, S.; Ponnambalam, K.; Kontogiannis, K.; Stacey, B: A Maintainability Model for Industrial Software Systems Using Design Level Metrics. In: Proceedings of the 7th Working Conference on Reverse Engineering (WCRE'00). IEEE Computer Society, Washington, DC, USA, S. 248-256, November 2000
- [PolPiaRui03] Polo, M.; Piattini, M.; Ruiz, F. (Hrsg.): Advances in Software Maintenance Management: Technologies and Solutions. Idea Group Publishing, Preface, September 2003
- [RamLeh00] Ramil, J. F.; Lehman, M. M.: Metrics of Software Evolution as Effort Predictors - A Case Study. In: Proceedings of the International Conference on Software Maintenance (ICSM'00), IEEE Computer Society, Washington, DC, USA, S. 163-172, Oktober 2000
- [TahGreKon99] Tahvildari, L.; Gregory, R.; Kontogiannis, K.: An Approach for Measuring Software Evolution Using Source Code Features. In: Proceedings of the 6th Asia Pacific Software Engineering Conference (APSEC'99), IEEE Computer Society, Washington, DC, USA, S. 10-17, Dezember 1999
- [TahKonMyl03] Tahvildari, L.; Kontogiannis, K.; Mylopoulos, J.: Quality-driven software re-engineering. In: Journal of Systems and Software, Elsevier Science, Band 66 Heft 3, S. 225-239, Juni 2003
- [VanSciSpr02] VanDoren, E.; Sciences, K.; Springs, C.: Maintainability Index Technique for Measuring Program Maintainability. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2002
http://www.sei.cmu.edu/activities/str/descriptions/mitmpm_body.html
(Stand 31.03.2006)

- [Wel01] Welker, K. D.: The Software Maintainability Index Revisited. Crosstalk. In: Journal of Defense Software Engineering, S. 18-21, August 2001
- [WelOma95] Welker, K. D.; Oman, P. W.: Software Maintainability Metrics Models in Practice. Crosstalk. In: Journal of Defense Software Engineering, Band 8 Heft 11, S. 19-23, November/Dezember 1995
- [WelOmaAtk97] Welker, K. D.; Oman, P. W.; Atkinson, G. G.: Development and Application of an Automated Source Code Maintainability Index. In: Journal of Software Maintenance: Research and Practice, Band 9 Heft 3, S. 127-159, Mai 1997
- [YiWuGan04] Yi, T.; Wu, F.; Gan, C.: A Comparison of Metrics for UML Class Diagrams. In: ACM SIGSOFT Software Engineering Notes, Band 29 Heft 5, S. 1-6, September 2004

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und nur die angegebenen Quellen verwendet habe.

Datum

Christiane Taras