

Jan Kurniawan

Kritische Erfolgsfaktoren von Full Function
Point Schätzungen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Jörg Raasch
Zweitgutachter : Prof. Dr. rer.nat. Kai von Luck

Abgegeben am 14.11.2007

Jan Kurniawan

Thema der Bachelorarbeit

Kritische Erfolgsfaktoren von Full Function Point Schätzungen

Stichworte

Aufwandschätzung, Functionpoints, Full Functionpoints, COSMIC-FFP, Widgetpoints

Kurzzusammenfassung

Im Projektmanagement sind Gründlichkeit und Professionalität Basis unternehmerischer Entscheidung. Dennoch stützen selbst renommierte Unternehmen die Planungsphase vorwiegend auf subjektive Einschätzungen Einzelner. Diese stark personenabhängige Einschätzung trägt weitreichende Konsequenzen für zeitliche und personelle Ressourcenaufteilung im Betrieb. Als Schätzgrundlage hat sich für das Softwarehaus intersoft AG daher die Notwendigkeit einer Verfahrensroutine ergeben.

Im Rahmen der Arbeit wird die Full Functionpoint Methode in dem Unternehmen zur Planung von Ressourcen eingeführt, welche Schätzabweichungen auf ein Minimum reduzieren soll. Es werden die theoretischen und praktischen Grundlagen der funktionalen Größenmessung dargelegt sowie Kriterien für die Implementierung im Unternehmen und des unterstützenden Tools definiert. Anhand der Untersuchungsergebnisse kann schlussendlich eine verfahrensunterstützende Software konzipiert und realisiert werden, die das Projektmanagement sowohl transparent gestalten und den Ressourcenanspruch realitätsnah einschätzen hilft.

Jan Kurniawan

Title of the paper

Key factors of Full Functionpoint estimation

Keywords

Cost Estimation, Functionpoints, Full Functionpoints, COSMIC-FFP, Widgetpoints

Abstract

Despite thorough and professional management, most companies rely on subjective estimations when scheduling capacity, time and human resources in projects. This technique has proven insufficient and too limited to intersoft AG.

Therefore it has become a necessity to establish a standard procedure for project measurement. Within the scope of this paper the Full Functionpoint Analysis is set up at intersoft AG to approach accuracy in resource planning and hence minimize discrepancies in cost estimation. In the following, theory and practise of functional measurement is being discussed. Furthermore the key criteria for the implementation in the company and supporting tools are being elaborated and set up. Its purpose is to support routines and procedures in the measurement process to create transparent estimations and help coordinating work.

Danksagung

Hiermit möchte ich mich vielmals für die vielen hilfreichen Anregungen bei meinem betreuenden Professor Jörg Raasch bedanken. Von der guten Zusammenarbeit hat diese Arbeit sehr profitiert.

Danken möchte ich auch meinen Lektoren Dr. Dr. Kathrin Sohr und Thomas Peter.

Und danken möchte ich auch meiner Familie, insbesondere meiner Schwester für die viele Unterstützung.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung	1
1.2.1	Die Zukunft der FP in Unternehmen am Beispiel der intersoft AG.	2
1.3	Aufbau dieser Arbeit	2
2	Grundsätze und Verfahren der Aufwandschätzung	3
2.1	Einleitung	3
2.2	Objektorientierte Aufwandschätzung	4
2.3	Cosmic FFP bei intersoft. Eine Zusammenfassung der Ergebnisse von Dominik Pohlschneider	4
2.3.1	Zählmethoden	4
2.3.2	Schätzmethoden	8
2.4	Veränderungen in der Firma seit Pohlschneider	12
2.5	Zusammenfassung Cosmic Full Functionpoints	12
2.5.1	Gewinnung der Informationen für die Functional User Requirements	14
2.5.2	Die Cosmic-FFP Mapping Phase	14
2.5.3	Die Messphase	18
2.5.4	Zählung von Weiterentwicklungen	19
2.6	Vergleich Widgetpunkte – COSMIC-FFP	20
2.7	Beispielzählung eines abgeschlossenen Projektes von intersoft	20
2.8	Beispielzählung einer graphischen Oberfläche mit Widgetpunkten	26
2.8.1	Zählen von Widgetpunkten	26
2.8.2	Beispielzählung anhand des GeVo TAB	27
2.8.3	Ergebnis der Zählung	29
2.8.4	Vergleich zu COSMIC-FFP	30
2.8.5	Offene Fragen Widgetpoints	31
2.9	Fazit	32
3	Anforderungen an eine Software zur Erfassung und Auswertung von Cosmic FFP	33
3.1	Einleitung	33
3.2	Kriterien	33
3.3	Fazit	33
4	Evaluation vorhandener Software zur Erfassung und Analyse von COSMIC-FFP	34
4.1	Einleitung	34
4.2	Bewertung der Software	34
4.3	Zusammenfassung der Bewertungen	37
4.4	Fazit	37
5	Konzeption eines Tools	38
5.1	Einleitung	38
5.1.1	Fachliche Architektur	38
5.1.2	Technische Architektur	41
5.2	Machbarkeitsstudie	42
5.3	Fazit	43
6	Toolentwicklung	44
6.1	Einleitung	44
6.2	Implementierung	44
6.3	Datenbank Design	45
6.3.1	Entity-Relationship Model	45
6.3.2	Businessobjekte	46
6.4	Eingabemaske des Prototypen	48
6.5	Metriken des Tools	48
6.5.1	Missbrauch von Metriken	49
6.5.2	Bereits integrierte Metriken	50
6.5.3	Zukünftige Metriken	54
6.6	Fazit	58
7	Die Einführung im Unternehmen	59

7.1	Eignung der Fachkonzepte für eine Zählung	59
7.2	Vorgehensweise zur Einführung von Aufwandschätzungen.....	61
7.2.1	Bildung eines Competence Centers	62
7.2.2	Bisherige Fachkonzepte auf Schätzbarkeit analysieren	62
7.2.3	Ressourcen und Termine planen	62
7.2.4	Teams zu den geplanten Termin schulen	63
7.2.5	Teams bei den ersten Messungen unterstützen.....	63
7.2.6	Dokumentation der Zählung	63
7.2.7	Feedback der Schulungsteilnehmer	64
7.2.8	Prüfen der Zählungen	64
7.2.9	Schätzprognosen erstellen	64
7.2.10	Schätzabweichung messen und Messfehler	64
7.2.11	Ergebnisse der Schätzung an die Testteams / ganze Firma publizieren ..	65
7.2.12	Vorgänge wiederholen	65
8	Ausblick und Fazit.....	66
	Abkürzungsverzeichnis.....	68
	Literaturverzeichnis	69
	Tabellenverzeichnis	70
	Abbildungsverzeichnis.....	71
	Anhang	73
	Präsentationsunterlagen der Schulung zum COSMIC-FFP Zähler.....	73
	Datenbanktabellen und Beziehungen	87
	Klassendiagramm	106

1 Einleitung

1.1 Motivation

Die intersoft AG entwickelt eine Standardgeschäftsprozesssoftware für diverse Versicherungsunternehmen. Die Fachkonzepte werden dabei in Drei-Monats-Iterationen, so genannten Meilensteinen, realisiert. Ein Fachkonzept enthält Aussagen über Informationen und Regeln, die das zukünftige System enthalten muss.

Der Aufwand für die Implementierung von Fachkonzepten wird im Unternehmen bisher subjektiv von verschiedenen Personen geschätzt. Dabei verfügen diese nicht immer über Informationen und Erfahrungswerte, die eine adäquate Einschätzung des Arbeitsunterfangens gewährleisten würden. Folglich schwankt die Genauigkeit der Aufwandschätzungen merklich mit Praxiserfahrung und Routine des Schätzenden. Schätzerfahrung droht z.B. durch personelle Änderungen verloren zu gehen.

Die durchschnittliche Abweichung des geschätzten und tatsächlichen Arbeitsvolumens von amerikanischen IT-Projekten im Jahr 2003 lag laut der der Standish Group [STANDISH95] bei 222%. Dementsprechend stiegen die Kosten für die Projekte im Schnitt 189% über das Geplante. Diese Diskrepanz führt zu Unzufriedenheit beim Kunden, dessen in Auftrag gegebene Projekte nicht rechtzeitig fertig gestellt werden können. Mit dem steigenden Termindruck sinkt die Qualität der Software. Diese unberechenbaren Schätzabweichungen möchte die intersoft AG minimieren, um fundiert Kosten und Aufwände planen zu können.

Die Kosten für ein einziges misslungenes Projekt sind höher als aller Aufwand, der nötig ist, um Aufwandschätzungen in einem Unternehmen einzuführen. Die Kosten für die Aufwandschätzung selbst beträgt ca. 0,5 – 1% des Gesamtaufwands. Gemessen an den bisherigen Abweichungen und ihren Mehrkosten ein zu vernachlässigender Faktor [BUNDSCHUH04 S.116]. Die Gegenüberstellung dieses Verhältnisses zeigt die Einsparmöglichkeiten, die das Unternehmen durch Einsatz eines Metrikprogramms zur Aufwandeinschätzung erzielen könnte. Die Höhe der Schätzabweichungen verdeutlichen, dass eine Einführung nicht früh genug stattfinden kann, da ohne präzise Einschätzung dem Unternehmen viel Kapital verloren gehen würde.

Um diesen Zustand zu ändern, möchte die intersoft AG Schätzmethoden für die Berechnung von Aufwänden benutzen. In seiner Diplomarbeit ermittelte Dominik Pohlschneider ein passendes Schätzverfahren für die intersoft AG [POHLSCH06]. Als geeignetes Verfahren stellten sich die Cosmic Full Functionpoints (im Weiteren CFFP) heraus. Das Verfahren misst anhand von Datenbewegungen den funktionalen Umfang einer Software. Durch die Bestimmung des funktionalen Umfangs ließe sich unter anderem dem Kunden besser veranschaulichen, wofür die geforderten Mittel eingesetzt werden.

1.2 Zielsetzung

Ziel dieser Arbeit ist es, die kritischen Erfolgsfaktoren für eine erfolgreiche Einführung von FFP in einem Unternehmen zu definieren. Es muss ermittelt werden, ob in der Firma diese Faktoren umgesetzt werden können. Sollte dies nicht der Fall sein, gilt es, die benötigten Voraussetzungen zu finden und zu schaffen.

Es soll dementsprechend ein Tool zur Erfassung und Analyse der Functionpoints konzipiert, entwickelt und eingeführt werden.

Zur Bestimmung zukünftiger Aufwände sollen Daten aus bereits realisierten Projekten in das Tool eingepflegt werden. Mittels der historischen Daten lassen sich dann zukünftige Aufwände schätzen.

Im Rahmen einer Schulung werden mehrere Testteams zu 'Zählern' ausgebildet. Diese Zähler erfassen dann Projekte ihres Teams und pflegen die Daten in das Tool ein. Der Analyst im Competence Center stellt dann mit Hilfe der Daten Schätzprognosen für die Projekte auf.

1.2.1 Die Zukunft der FP in Unternehmen am Beispiel der intersoft AG.

Wenn nach dieser Bachelorarbeit bei der intersoft AG die ersten Testschätzungen abgeschlossen sind und die Schätzabweichungen weit unter den Differenzen der früheren subjektiven Schätzmethoden liegen, teilen die Testteams ihre Erfolge ihren Kollegen mit. Dadurch erhöht sich die Akzeptanz gegenüber den CFFP enorm. Die CFFP Schätzung kann nun unternehmensweit eingeführt werden.

Die Erfahrungsdatenbank ermöglicht aufgrund der steigenden Menge von historischen Projektdaten nun immer genauere Schätzwerte. Dies reduziert mit der Zeit die Schätzabweichung.

Die Qualität der Software des Unternehmens steigt im Allgemeinen, weil Abgabetermine eingehalten werden. Die Software wird nicht mehr unter Termindruck fertig gestellt, weshalb die Fehlerquote sinkt. Die Zufriedenheit des Kunden steigt, weil dieser sein Produkt termingerecht und mit voller Funktionalität erhält. Desweiteren kann der Kunde anhand der Umfangsmessung nachvollziehen, wofür sein Geld investiert wurde.

Hat sich das Verfahren etabliert, so können die Entwickler auch täglich ihre Fortschritte der jeweiligen Implementierung im Tool dokumentieren. Dem Projektmanager / Teamleiter würden somit viele Optionen des Projektcontrolling zur Verfügung stehen.

War die Einführung ein Erfolg und hat sich die CFFP Aufwandschätzung in den Alltag des Unternehmens integriert, wäre es sogar möglich, dass die intersoft AG als Berater für andere Firmen fungiert.

Weiterhin wäre es möglich, die erhobenen Daten einer Metrik Organisation zu Verfügung zu stellen und somit einen Beitrag zur Etablierung von FFP allgemein zu leisten.

1.3 Aufbau dieser Arbeit

Im folgenden Kapitel werden die Grundlagen von Aufwandschätzungen behandelt sowie verschiedene Methoden und Verfahren zur Umfangsmessung von Projekten vorgestellt. Zudem wird ein passendes Verfahren für die intersoft AG ermittelt und dargestellt.

In Kapitel 3 werden die Anforderungen an ein Tool zur Unterstützung des Schätzprozesses definiert. Auf dem freien Markt verfügbare Software wird in Kapitel 4 auf Eignung hinsichtlich dieser Anforderungen geprüft.

In Kapitel 5 werden die fachliche und technische Architektur für die Eigenentwicklung eines Tools erarbeitet. Das darauffolgende Kapitel behandelt die Realisierung des Tools. Nach Konzeption und Entwicklung des Tools folgt in Kapitel 7 die Einführung des Verfahrens im betreuenden Unternehmen.

Abgeschlossen wird die Arbeit mit Ausblick und Fazit in Kapitel 8.

2 Grundsätze und Verfahren der Aufwandschätzung

2.1 Einleitung

Die Aufwandschätzung ist Teil jedes Softwareprojektes. Mit ihr wird versucht, den Umfang und damit die benötigten Ressourcen einer Projektanforderung zu schätzen. Die grundlegenden Funktionen von Aufwandschätzungen sind Messen und Schätzen.

Als Basis für die Schätzung dient die Messung der Größe des Produktes. Anhand dieser Umfangsmessung kann die Größe des Projektes geschätzt werden.

Gemessen wird also der Umfang eines Produktes. Aufgrund dieser Messungen werden die Schätzungen vorgenommen. Grundsätzlich gibt es zwei verschiedene Arten von Umfangsmessungen: die Messung der funktionalen Vorgaben und die Messung des Programmumfangs.

Bei der Messung des Programmumfangs werden die Lines of Code (LOC) gemessen. Diese Methode gilt in der Literatur mittlerweile als umstritten. Die Nachteile bestehen darin, dass die LOC erst zu einem späten Zeitpunkt bekannt sind, und dass diese trotz Vergleichstabellen nicht über Programmiersprachen hinweg vergleichbar sind, da kein Standard für die Zählung von LOC existiert [BUNDSCHUH S.203]. Capers Jones und viele andere Metrik-Spezialisten lehnen deshalb die LOC-Methoden ab [vgl. BUNDSCHUH04 S.135].

Die Messung der funktionalen Vorgaben hingegen kann zu einem frühen Zeitpunkt stattfinden. Auch ist diese Methode unabhängig von der Programmiersprache (siehe Abbildung 1) [BUNDSCHUH04 S.135].

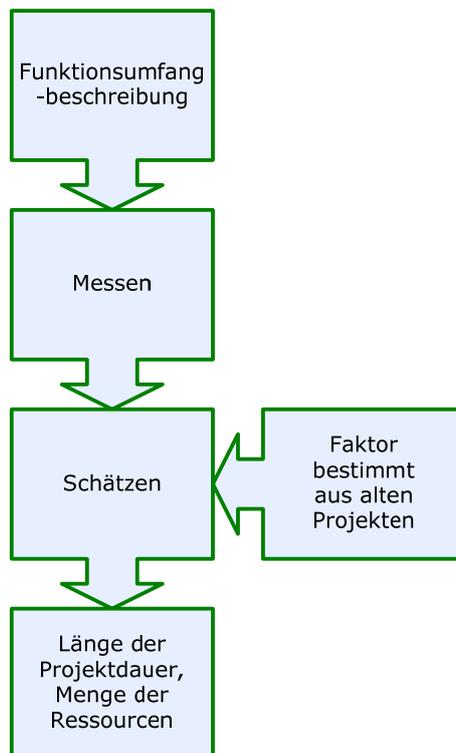


Abbildung 1: Ablauf einer Schätzung

Im nächsten Kapitel wird eine Variante der funktionalen Aufwandschätzung betrachtet: die Object Point Methode. Danach folgt eine Zusammenfassung der Diplomarbeit von Pohlschneider [POHLSCH06], welcher für die intersoft AG das geeignete Verfahren zur Umfangschätzung und Schätzprognose ermittelt hat.

Anschließend wird in Kapitel 2.3 das von POHLSCHNEIDER ermittelte Verfahren vorgestellt.

Es folgt eine Marktanalyse der Tools, welche das gewählte Verfahren unterstützen.

Am Ende des Kapitels wird ein Projekt der intersoft AG mit einer weiteren Variante der funktionalen Messung, den Widgetpunkten, gemessen. Es folgt ein Vergleich mit einer COSMIC-FFP Messung.

2.2 Objektorientierte Aufwandschätzung

Eine Variante der Function-Point Methode ist die Object-Point-Methode. Diese ist eine Möglichkeit zur Schätzung objektorientierter Systementwicklungen.

Es gibt eine Vielzahl von Object-Point Methoden, von denen sich keine zu einem Standard etablieren konnte. Die meisten der Ansätze wurden nach wenigen Jahren nicht mehr weiter verfolgt oder scheinen noch nicht ausgereift zu sein [BUNDSCHUH04 S.357].

Ein weiteres Problem des objektorientierten Ansatzes liegt in seiner Natur: Die Methode eignet sich nur zum Schätzen von objektorientierten Systemen und deshalb nicht für prozedurale Sprachen oder Scriptsprachen. Firmen, welche über Jahre iterativ die gleiche Software entwickeln, könnten dieses Verfahren nicht übergreifend einsetzen, weil meist Teile der Software in nicht objektorientierter Umgebung implementiert wurden. Beim betreuenden Unternehmen liegen Teile der Software aus Legacy Gründen in einer prozeduralen Sprache vor.

Ein weiterer Grund für die schwache Durchsetzung dieser Verfahren könnte sein, dass die IFPUG in einer Fallstudie gezeigt hat, dass objektorientierte Systeme sich problemlos mit der bewährten Function-Point Methode messen lassen [BUNDSCHUH04 S.359].

Die funktionsorientierte COSMIC-FFP Aufwandschätzung hingegen ist schon als ISO-Norm etabliert und unabhängig von der Implementierungsumgebung. [BUNDSCHUH04 S.357-359]

2.3 Cosmic FFP bei intersoft. Eine Zusammenfassung der Ergebnisse von Dominik Pohlschneider

2.3.1 Zählmethoden

Pohlschneider hat die Schätzmethoden IFPUG, Cosmic FFP und Mark II bezüglich 13 Kriterien getestet und bewertet.

Legende

Bewertung	Beschreibung
+++	Die Anforderung wird voll erfüllt
++	Die Anforderung wird erfüllt
+	Die Anforderung wird begrenzt erfüllt
O	Die Anforderung wird nicht erfüllt

Tabelle 1: Legende der Bewertung von Schätzverfahren

Funktionelles Einsatzgebiet

Methode	Bewertung
IFPUG	+
COSMIC	++
Mark II	+

Tabelle 2: Bewertung - Funktionelles Einsatzgebiet

IFPUG und Mark II eignen sich nur eingeschränkt für Kontroll- / Echtzeitsysteme. Sie eignen sich nur für Geschäftsanwendungen und Managementsysteme. Die Cosmic Methode unterstützt hingegen alle Anwendungstypen.

Projektart (Evolutionsprojekte / Neuentwicklungen)

Methode	Bewertung
IFPUG	+++
COSMIC	++
Mark II	++

Tabelle 3: Bewertung – Projektart

Auf Basis der IFPUG FPA hat die NESMA eine Zählvariante extra für Evolutionsprojekte entwickelt. Deshalb eignet sich die IFPUG Methode besser für Evolutionsprojekte als die anderen Methoden. Die verbleibenden Methoden erfüllen die Anforderungen ebenfalls.

Standardisierung

Methode	Bewertung
IFPUG	++
COSMIC	++
Mark II	++

Tabelle 4: Bewertung – Standardisierung

Ein Standard stellt sicher, dass die Methode klar definiert ist und somit weniger unter Akzeptanzproblemen leidet. Alle Methoden sind durch einen internationalen Standard anerkannt.

Geographische Verbreitung / Nutzerzahl

Methode	Bewertung	Länder
IFPUG	+++	International verbreitet
COSMIC	++	England, Australien, Kanada, Finnland, Japan, USA, Indien
Mark II	+	England

Tabelle 5: Bewertung - Geographische Verbreitung

Durch den Netzwerkeffekt wächst mit steigender Anzahl der Benutzer der Nutzen an einem Standard. Mit einer größeren Anzahl von Nutzern wächst z.B. auch die Menge der verfügbaren Vergleichswerte von Schätzungen oder die Menge der Personen, die Fehler in der Methode entdecken könnten.

Mark II konnte sich bisher nur in England durchsetzen und erfüllt die Anforderungen deshalb nur bedingt. IFPUG hat eine höhere Verbreitung als die Cosmic Methode und erfüllt deshalb die Anforderung voll.

Weiterentwicklung der Methoden

Methode	Bewertung
IFPUG	++
COSMIC	+++
Mark II	0

Tabelle 6: Bewertung - Weiterentwicklung der Methoden

Die Informatik unterliegt ständigen Innovationen. Die Schätzmethoden müssen gegebenenfalls an Neuerungen angepasst werden, um weiterhin zuverlässige Ergebnisse liefern zu können.

Die IFPUG Methode wird aktuell noch weiterentwickelt und erfüllt somit diese Anforderung. Die Cosmic Methode ist sehr jung, und wird von einem internationalen Team auf Basis von modernen Entwicklungsprinzipien vorangetrieben. Deshalb erfüllt sie die Anforderungen voll. Die Cosmic FFP Methode ist darauf ausgelegt, Mark II abzulösen. Die Mark II Methode wird nicht mehr weiterentwickelt und erfüllt deshalb die Anforderungen nicht.

ISBSG Unterstützung

Methode	Bewertung
IFPUG	++
COSMIC	++
Mark II	++

Tabelle 7: Bewertung - ISBSG Unterstützung

Die International Software Benchmarking Standards Group bietet eine Datenbank mit neutralen Benchmark-Studien. Diese kann zu Vergleichszwecken hinzugezogen werden und erleichtert somit das Auffinden von Unregelmäßigkeiten eigener Daten. Alle drei Methoden werden durch die ISBSG unterstützt.

Aufwand

Methode	Bewertung
IFPUG	++
COSMIC	++
Mark II	++

Tabelle 8: Bewertung – Aufwand

Der Aufwand der Erfassung wird hauptsächlich durch die Qualität der Dokumentation bestimmt. Deswegen werden alle Methoden gleichermaßen neutral bewertet.

Toolunterstützung

Methode	Bewertung
IFPUG	+++
COSMIC	+++
Mark II	+

Tabelle 9: Bewertung – Toolunterstützung

Bei der Einführung von Functionpoints ist der unterstützende Tooleinsatz ein ausschlaggebender Faktor, um Akzeptanzprobleme zu mindern. IFPUG und die Cosmic FFP Methode bieten eine Vielzahl von unterstützenden Tools. Für Mark II ließen sich kaum Tools finden.

Leichtigkeit

Methode	Bewertung
IFPUG	+
COSMIC	++
Mark II	++

Tabelle 10: Bewertung – Leichtigkeit

Mit der Mark II und die Cosmic FFP Methode lassen sich die BFC leicht abgrenzen. Bei der IFPUG Methode ist die korrekte Zählung für einen ungeübten Zähler sehr schwierig. Deshalb wird IFPUG gegenüber den anderen Methoden schlechter bewertet.

Robustheit

Methode	Bewertung
IFPUG	0
COSMIC	++
Mark II	+

Tabelle 11: Bewertung – Robustheit

Eine Fehlinterpretation eines Elementarelementes der Cosmic FFP Methode hat keine Auswirkungen auf das Ergebnis.

Bei der Mark II Methode wirken sich Fehlinterpretationen in der Verarbeitung auf das Ergebnis aus. Die Anforderung wird damit nur eingeschränkt erfüllt.

Die IFPUG Methode besitzt Schwellenwerte, so dass ein geringer Unterschied des tatsächlichen Umfangs zu einem großen Unterschied der Bewertung führen kann.

Transparenz / Nachvollziehbarkeit

Methode	Bewertung
IFPUG	+
COSMIC	++
Mark II	++

Tabelle 12: Bewertung – Transparenz

Die Cosmic FFP und die Mark II Methode erfüllen die Anforderungen, da die Algorithmen für diese Verfahren frei verfügbar sind. Die Erfahrungswerte auf denen IFPUG beruht, können heute nicht mehr nachvollzogen werden. Deswegen erfüllt IFPUG die Anforderungen nur bedingt.

Konstanz

Methode	Bewertung
IFPUG	0
COSMIC	++
Mark II	++

Tabelle 13: Bewertung - Konstanz

Wie schon unter dem Punkt Robustheit erwähnt (siehe Tabelle 11), können geringe Unterschiede in der Eingabe zu großen Unterschieden in der Bewertung führen. Dies widerspricht dem Prinzip der Konstanz. Bei Cosmic FFP und Mark II führen kleine Änderungen bei der Eingabe nur zu kleinen Änderungen bei der Ausgabe. Sie erfüllen damit die Anforderungen.

Objektivität

Methode	Bewertung
IFPUG	++
COSMIC	++
Mark II	++

Tabelle 14: Bewertung – Objektivität

Alle drei Methoden beinhalten keine subjektiven Bewertungen und erfüllen somit gleichermaßen die Anforderungen.

Zusammenfassung

Anforderung	IFPUG	COSMIC	Mark II
Funkt. Einsatzgebiet	+	++	+
Projektart	+++	++	++
Standardisierung	++	++	++
Geograph. Verbreitung	+++	++	+
Weiterentwicklung	++	+++	0
ISBSG-Unterstützung	++	++	++
Aufwand	++	++	++
Toolunterstützung	+++	+++	+
Leichtigkeit	+	++	++
Robustheit	0	++	+
Transparenz	+	++	++
Konstanz	0	++	++
Objektivität	++	++	+
Gesamtbewertung	+(+)	++	+(+)

Tabelle 15: Zusammenfassung der Bewertung

Die IFPUG FPA erfüllt die Anforderungen der Robustheit und Konstanz nicht. Die Mark II FP Methode erfüllt die Forderung der Weiterentwicklung nicht. Die Cosmic FFP hingegen erfüllen alle Anforderungen und kann sich damit von den anderen Methoden absetzen.

2.3.2 Schätzmethode

Schätzmethode sind Verfahren, die mittels des gezählten Umfangs einer Software eine Schätzung der benötigten Personenstunden vornehmen.

Es gibt zwei Arten von Schätzmethode: zum einen erfahrungsbasierte Modelle, welche ihre Prognose auf Vergleiche zu alten Projekten stützen, zum anderen spezifische Prognosemodelle, die auf Basis von Erfahrungsdaten repräsentativer Unternehmen entwickelt wurden. Durch die Identifikation von Kostentreibern und Skalenfaktoren wird eine Aufwandsprognose erstellt. Diese Methode eignet sich besonders für Unternehmen, denen keine historischen Projektdaten vorliegen.

Erfahrungsbasierte Aufwandschätzung

Erfahrungsbasierte Schätzungen gehen davon aus, dass Einflussfaktoren für Kosten und Aufwand in dem Unternehmen stabil sind. Die Methode berücksichtigt nur den funktionalen Umfang einer Software als Aufwandstreiber. Der Aufwand bereits realisierter Projekte wird in einer Datenbank gesammelt.

Aus der Relation von Functionpoints zu den benötigten Personenstunden lässt sich mittels statistischer Verfahren eine Schätzkurve erstellen, die auf einer Regressionskurve basiert. Die Regressionsanalyse zeigt mit dem Bestimmtheitsmaß (R^2) an, wie hoch die durchschnittliche Abweichung der Punkte zu der Kurve ist. Es gilt, für das Unternehmen den passenden Typ Regressionskurve mit der niedrigsten durchschnittlichen Abweichung zu finden. Die verschiedenen Typen der Regression sind z.B.: logarithmisch, polynomisch oder exponentiell.

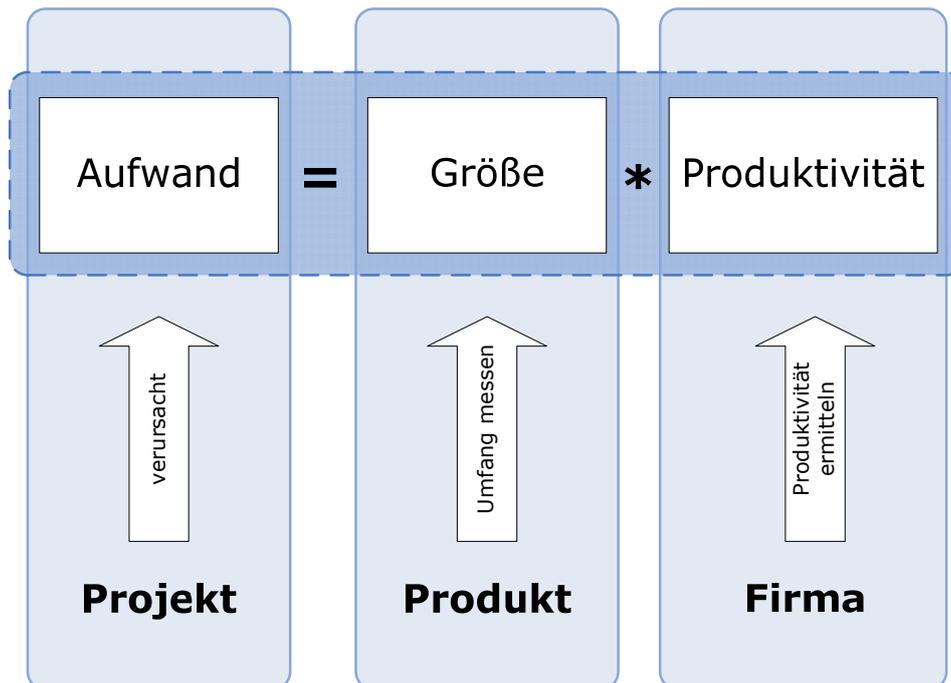


Abbildung 2: Verfahren zur Berechnung des Aufwands

Abbildung 2 zeigt das Verfahren zur Ermittlung des Aufwands eines Projektes. Der Aufwand wird über zwei Faktoren bestimmt. Der Faktor Größe wird über die objektive Umfangsmessung des Projektes bestimmt. Dieser Faktor sollte bei der gleichen Aufgabe in unterschiedlichen Unternehmen identisch sein. Der Faktor Produktivität wird von der Unternehmenskultur bestimmt und ist von Unternehmen zu Unternehmen unterschiedlich.

Das COCOMO Modell

Das COCOMO Modell gliedert sich in zwei Untermodelle. Mit dem Early-Design Modell (kurz EDM) lassen sich Schätzungen vornehmen, wenn noch keine eigenen Erfahrungsdaten vorliegen. Das Post-Architecture Modell (kurz PAM) berechnet den Aufwand mittels 17 Kostentreibern und fünf Skalenfaktoren. Diese entscheiden, ob der Aufwand mit Größe des Projektes über- oder unterproportional steigt.

Die Methode ist nur für erfahrene Schätzer geeignet und eignet sich vor allem für Weiterentwicklungen sowie Wartungsprojekte.

Kommerzielle Tools für die Aufwandschätzung mit COSMIC-FFP

KnowledgePLAN

KnowledgePLAN erstellt spezifische Prognosemodelle (siehe oben).

Die Anzahl der Aufwand- und Kostentreiber ist sehr hoch, weshalb eine ausführliche Einarbeitung problematisch ist.

Die Methode erlaubt eine frühe Schätzung anhand weniger Angaben.

ISBSG – Practical Project Estimation

Das Tool nimmt Schätzungen anhand einer Datenbank mit 3000 Projektdaten aus der ganzen Welt [ISBSG] vor. Das Toolkit bietet Methoden und Techniken (Macro- und Micro-Estimating) für die Schätzung eigener Projekte aufgrund dieser Datenbasis.

Im Folgenden werden die verschiedenen Methoden zur Umfangsmessung Bewertet.

Aufwand

Methode	Bewertung
Erfahrungsb. AS	+++
COCOMO	+
KnowledgePLAN	+
ISBSG	++

Tabelle 16: Bewertung von Tools – Aufwand

Für eine erfahrungsbasierte Aufwandschätzung müssen prinzipiell nur die Anzahl der Functionpoints sowie die Menge der benötigten Personenstunden dokumentiert werden. Der Aufwand ist also sehr gering.

Der Aufwand für die Einführung von COCOMO und KnowledgePLAN ist zu Beginn sehr hoch, anschließend gestaltet sich der Aufwand aber gering. Die Anforderungen werden deshalb nur begrenzt erfüllt.

Der Aufwand für die ISBSG Methode ist gering, weshalb die Anforderungen erfüllt werden.

Toolunterstützung

Methode	Bewertung
Erfahrungsb. AS	++
COCOMO	+++
KnowledgePLAN	+++
ISBSG	+++

Tabelle 17: Bewertung von Tools – Toolunterstützung

Für COCOMO, KnowledgePLAN und ISBSG gibt es Softwaretools. Erfahrungsbasierte Schätzungen können schon mit Tabellenkalkulationsprogrammen durchgeführt werden.

Leichtigkeit

Methode	Bewertung
Erfahrungsb. AS	+++
COCOMO	+
KnowledgePLAN	+
ISBSG	++

Tabelle 18: Bewertung von Tools – Leichtigkeit

Für erfahrungsbasierte Aufwandschätzung sind keine Vorkenntnisse erforderlich. COCOMO und KnowledgePLAN bedürfen einer genauen Einarbeitung und Vorkenntnissen. ISBSG bedarf keiner Einarbeitung, benötigt aber einen Schätzer mit Vorkenntnissen.

Projektsteuerung

Methode	Bewertung
Erfahrungsb. AS	++
COCOMO	++
KnowledgePLAN	+++
ISBSG	++

Tabelle 19: Bewertung von Tools – Projektsteuerung

Alle Verfahren ermöglichen vielfältige Projektsteuerungsmöglichkeiten. KnowledgePLAN ermöglicht zusätzlich die Ermittlung der größten Einflussfaktoren auf das Projekt.

Schätzgenauigkeit

Methode	Bewertung
Erfahrungsb. AS	k.A.
COCOMO	k.A.
KnowledgePLAN	k.A.
ISBSG	k.A.

Tabelle 20: Bewertung von Tools – Schätzgenauigkeit

Über die durchschnittliche Schätzgenauigkeit der Verfahren liegen keine Informationen vor.

Tranzparenz

Methode	Transparenz
Erfahrungsb. AS	++
COCOMO	++
KnowledgePLAN	0
ISBSG	++

Tabelle 21: Bewertung von Tools – Transparenz

Alle Methoden bis auf KnowledgePLAN sind transparent. Es kann also nachvollzogen werden, wie die Tools ihre Schätzungen berechnen.

Objektivität

Methode	Bewertung
Erfahrungsb. AS	+++
COCOMO	0
KnowledgePLAN	0
ISBSG	0

Tabelle 22: Bewertung von Tools – Objektivität

Alle Verfahren bis auf erfahrungsbasiertes Aufwandschätzen berücksichtigen subjektive Einflussfaktoren und werden somit nie objektiv sein. Das erfahrungsbasierte Aufwandschätzen berücksichtigt keine subjektiven Faktoren.

Flexibilität

Methode	Bewertung
Erfahrungsb. AS	++
COCOMO	+
KnowledgePLAN	+
ISBSG	+

Tabelle 23: Bewertung von Tools – Flexibilität

Die Regressionskurve passt sich automatisch an veränderte Bedingungen an. Sollten die Projekte zu stark differieren, so kann eine zusätzliche Regressionsgerade aufgebaut werden. Fehlt bei den übrigen Verfahren der passende Parameter zum Justieren, so lässt sich das Ergebnis nicht anpassen.

Zusammenfassung

Anforderung	Erfahrungsbasierte AS	COCOMO	KnowledgePLAN	ISBSG
Typ	Cost Database	Formalbasiert	Formalbasiert	Cost Database
Aufwand	+++	+	+	+++
Toolunterstützung	++	+++	+++	+++
Leichtigkeit	+++	+	+	++
Projektsteuerung	++	++	+++	++
Schätzgenauigkeit	k.A.	k.A.	k.A.	k.A.
Transparenz	++	++	0	++
Objektivität	+++	0	0	0
Flexibilität	++	+	+	+
Gesamtbewertung	++	+(+)	+	+(+)

Tabelle 24: Zusammenfassung der Bewertung von Tools

Aufgrund der hohen Ergebnisqualität bei einem geringen Aufwand und geringer Einarbeitungszeit ist das erfahrungsbasierte Verfahren für die intersoft AG am geeignetsten.

2.4 Veränderungen in der Firma seit Pohlschneider

Bei der intersoft AG gab es seit der Diplomarbeit von Pohlschneider [POHLSCH06] keine Veränderungen bezüglich des Projektprozesses, welche sein Ergebnis betreffen. An dem Projektablauf sowie der Software hat sich nichts verändert. Lediglich die Anzahl der Projektiterationen pro Jahr soll von vier auf drei Iterationen pro Jahr reduziert werden. Dies hat keinen Einfluss auf die Einführung von Aufwandschätzungen.

Im Allgemeinen kann gesagt werden, dass es sich bei den COSMIC-FFP um das momentan modernste und kompatibelste [VOGELE S.9] Verfahren von Functionpoint Schätzungen handelt. Laut Vogelesang werden die COSMIC-FFP die Functionpoint Analyse in den nächsten Jahren ablösen [VOGELE S.9]. Da das Verfahren Informationssysteme unterstützt, ist die Wahrscheinlichkeit hoch, dass es zu der intersoft AG passen wird. Dies wird im Laufe der Arbeit noch evaluiert.

2.5 Zusammenfassung Cosmic Full Functionpoints

Um Qualität und Produktivität von Software zu analysieren, muss man messen. Funktionale Schätzungen müssen technologieunabhängig sein. Sie können benutzt

werden, um die Produktivität von verschiedenen Technologien zu vergleichen. Die Functionpoint Analyse ist ein Beispiel für das funktionale Messen. Sie erfasst erfolgreich die Charakteristiken von Management-Informationssystemen (MIS). Die Functionpoint Analyse (FPA) steht aber unter der Kritik, nicht für alle Arten von Software anwendbar zu sein, weil sie nicht alle Charakteristiken von Echtzeitsoftware erfassen kann und sich deswegen nicht zum Messen eignet.

Die Full Function Point Methode wurde 1997 veröffentlicht. Ihr Ziel war es, auch die Messung von Echtzeitsoftware anzubieten. Die Full Function Point Methode unterstützt nicht nur zusätzlich Echtzeitsoftware sondern auch technische- und Systemsoftware.

Motiviert durch vielversprechende Feldversuche entwickelten eine Reihe von Autoren die Cosmic-FFP Methode. Diese befindet sich bereits in der Version 2.2. Seit der Version 2.0 erfüllen die Cosmic-FFP die ISO/IEC 14143-1: 1998 Norm. [COSMIC 13]

Die Cosmic-FFP eignen sich für:

Geschäftsanwendungen, wie z.B. Bankensoftware, Versicherungssoftware, Buchhaltungssoftware, Einkaufsabwicklungssoftware, Distributionssoftware sowie Fabrikationssoftware. Diese Arten von Software zeichnen sich durch ein hohes Datenaufkommen aus.

Echtzeitsysteme, die mit Ereignissen aus der wirklichen Welt mithalten oder sie kontrollieren müssen, wie z.B. Software zur Prozesskontrolle, Software für Automatisierungstechniken.

Mischformen beider Anwendungsarten, wie z.B. Reservierungssysteme von Hotels oder Fluggesellschaften.

Die Cosmic-FFP Methode eignet sich zurzeit noch nicht für:

- Algorithmisch komplexe Software, wie sie in Expertensystemen, Simulationssoftware oder in der Wettervorhersage benutzt wird
- Software mit Prozessen, welche einen hohen Grad an Datenmanipulation haben, wie zum Beispiel Audiosoftware oder Videosoftware

Die Cosmic-FFP Methode umfasst Modelle, Regeln und Prozeduren, die aus Sicht des Benutzers auf die Software angewandt werden. Das Ergebnis dieser Methode gibt den 'Wert der Menge' (definiert durch die ISO-Norm) an, welcher die funktionale Größe darstellt. [S15]

Die Cosmic-FFP Methode wurde so entwickelt, dass sie unabhängig von Implementierungsentscheidungen ist, da sie nur den funktionalen Umfang der Software betrachtet. Um dies zu erreichen, wird die Messung nur auf die funktionalen Benutzeranforderungen (eng. Functional User Requests / FUR) der zu messenden Software angewandt.

Die COSMIC-FFP Methode stellt zwei Arten von Herangehensweisen zur Verfügung:

- Die eine Methode setzt komplett beschriebene FURs zum Schätzen voraus
- Die andere geht von noch nicht komplett umschriebenen FURs aus. Diese Methode soll ein noch früheres Schätzen möglich machen, hat aber den Nachteil, dass die Schätzabweichungen sehr groß sein können. In der betreuenden Firma ist das Ziel, mit komplett beschriebenen Funktionsumfangsbeschreibungen (FUBen) zu arbeiten, um möglichst genaue Schätzungen zu erzielen. [COSMIC S.16]

2.5.1 Gewinnung der Informationen für die Functional User Requirements

Aus Sicht der Cosmic-FFP Methode interessiert nur die Funktionalität, welche die Software dem User bietet. Die vom Benutzer geforderte Funktionalität an die Software wird 'Functional User Requirements' (kurz FUR) genannt. Diese geben an, welche Funktionalität die Software dem User bieten muss, nicht aber technische Voraussetzungen oder Qualitätsansprüche, die sagen, auf welche Art das Projekt realisiert werden muss.

Diese FUR können in verschiedenen Formen vorliegen oder erstellt werden:

- Als Pflichtenheft oder Fachkonzept, in welchem die funktionalen Anforderungen komplett definiert sind
- Liegt keine FUB bereits implementierter Projekte vor, so kann diese aus dem bestehenden Programm hergeleitet werden

Der Aufwand für die Erstellung von FUR aus einem undokumentierten System ist natürlich bedeutend höher [COSMIC, S17].

Für die COSMIC-FFP Zählung muss ein komplett beschriebenes Fachkonzept vorliegen. Liegt dieser Idealfall nicht vor, so kann das Fachkonzept in Teilmengen zerlegt werden. Sind Teile dieser Menge komplett beschrieben, so sind diese schätzbar. Für nicht komplett beschriebene Teilbereiche können evtl. Vergleiche zu ähnlichen Projekten gezogen werden.

2.5.2 Die Cosmic-FFP Mapping Phase

Identifikation von Schichten (Layer)

Die COSMIC-FFP Methode bietet gegenüber der IFPUG-Methode die Möglichkeit, aus Entwicklerperspektive zu messen (siehe **Fehler! Verweisquelle konnte nicht gefunden werden.**). Dies setzt voraus, dass ein Konzept der Architektur vorliegt. Der Nachteil dieser Art zu messen ist, dass ein Konzept für die Architektur zu einem späteren Zeitpunkt als die FUB vorliegt. Da die intersoft AG zu einem möglichst frühen Zeitpunkt messen möchte, wird aus Endbenutzersicht gemessen. Hierfür ist nur die Identifikation der klassischen Endbenutzersicht, also der Softwaregrenze zwischen Frontend und Backend nötig. Die übrigen Schichten einer Systemarchitektur werden nicht berücksichtigt (siehe Abbildung 3). [BUNDSCHUH04 S.351]

Die Cosmic-FFP Mapping Phase benutzt vordefinierte Regeln und Verfahren, um aus den Functional User Requirements ein spezifisches Softwaremodell zu erstellen, welches sich zum Messen der funktionalen Größe eignet.

Als erstes muss bestimmt werden, was gemessen werden soll und was zur Umgebung der Software gehört (z.B. Betriebssystem, Virtuelle Maschine).

Die Cosmic-FFP Methode unterscheidet dabei zwischen Frontend und Backend. Das Frontend wird vom User über die IO-Hardware benutzt. Dies können Dinge wie Maus, Drucker oder Bildschirm sein.

Das Backend stellt einen persistenten Speicher wie z.B. eine Datenbank dar.

Die Cosmic-FFP Methode betrachtet Datenbewegungen als Maßeinheit für ihre Schätzungen. Die Bewegungen 'Entry' und 'Exit' finden nur im Frontend, die Bewegungen 'Read' und 'Write' nur im Backend statt (siehe Abbildung 3 und Abbildung 6). [COSMIC 18]

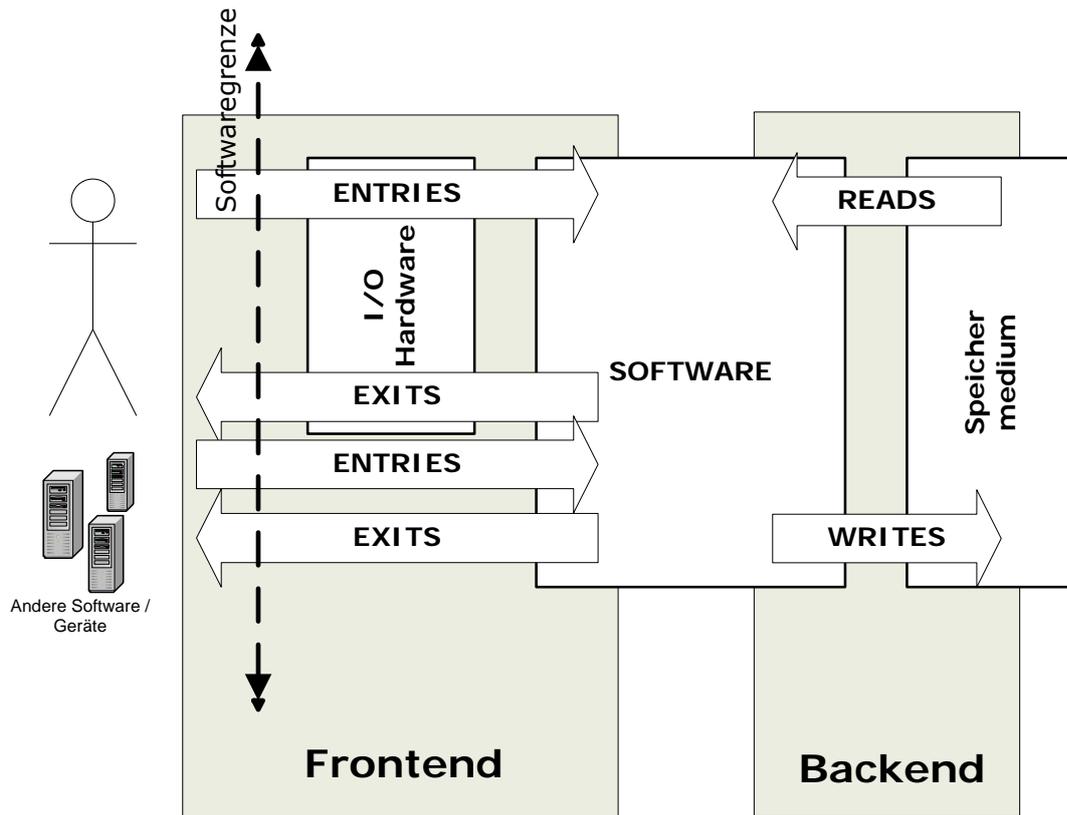


Abbildung 3: Datenbewegungen und Schichten (vgl. [COSMIC] S.19)

Des Weiteren muss identifiziert werden, was zu der zu messenden Software gehört und was zu der Umgebung. Diese Abgrenzung nennt man Softwaregrenze (siehe Abbildung 3 - Softwaregrenze).

Man identifiziere das Auslöseereignis (eng. trigger event) und den funktionalen Prozess, welcher von diesem angestoßen wurde. Die Softwaregrenze liegt dann zwischen dem Auslöser und dem funktionalem Prozess. [COSMIC 33]

Das Auslöseereignis kann z.B. ein Benutzer, eine andere Software, ein anderes Gerät oder aber ein zeitgesteuerter Auslösemechanismus (engl. Cronjob) sein.

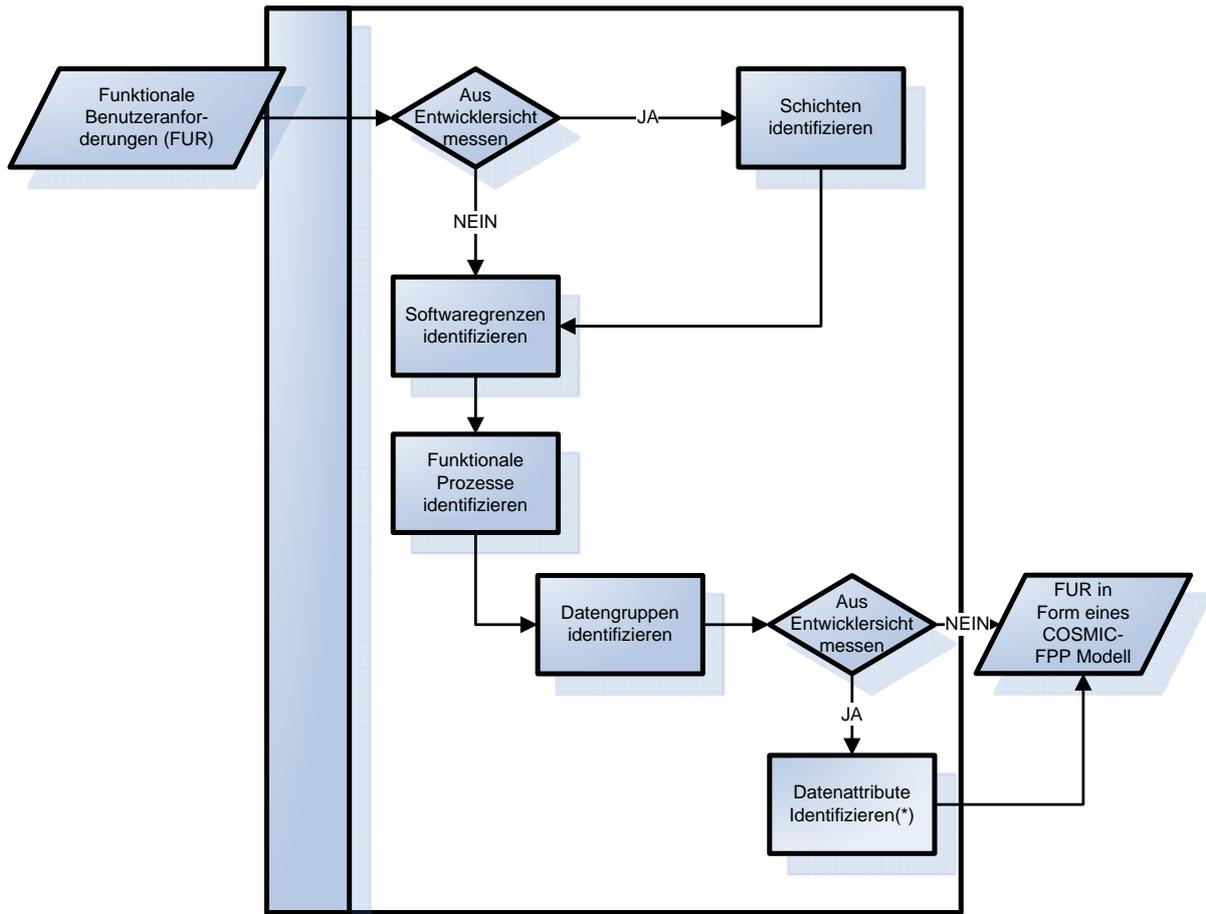


Abbildung 4: Die 'COSMIC-FFP Mapping' Phase (vgl. [COSMIC] S. 30)

Was wird gemessen?

Die Cosmic-FFP Methode geht davon aus, dass die zu messende Software folgende Kriterien erfüllt:

Grundprinzip 1: Die zu messende Software wird mit Eingaben gespeist und liefert dem Benutzer eine sinnvolle Ausgabe.

Grundprinzip 2: Die zu messende Software verändert Teile von Informationen, die aus Datengruppen bestehen. Eine Datengruppe wiederum besteht aus mehreren Datenattributen.

Funktionalprozesse

Es werden nun die funktionalen Prozesse aus den FUR bestimmt. Eine FUR wird in kleinste mögliche Funktionalprozesse zerlegt. Datenbewegungen dürfen nur innerhalb dieses Funktionalprozesses stattfinden. Ein funktionaler Prozess ist eine elementare Komponente, die Teil einer Menge von funktionalen Benutzeranforderungen ist. Er wird direkt oder indirekt von einem oder mehreren Ereignissen ausgelöst. Er ist abgeschlossen, wenn er alles ausgeführt hat, was zur Antwort auf das Auslöseereignis benötigt wird [COSMIC 36].

Ein Funktionalprozess muss dabei folgende Kriterien erfüllen:

- Ein Funktionalprozess muss von einer FUR abgeleitet sein
- Ein Funktionalprozess wird ausgeführt, wenn das passende Auslöseereignis auftritt
- Ein Funktionalprozess muss mindesten zwei Datenbewegungen haben: einen Entry + Exit / Entry + Write
- Ein Funktionalprozess endet, sobald alles ausgelöst / ausgegeben wurde, was mit dem FP ausgelöst / ausgegeben werden soll
- Datenbewegungen dürfen nur innerhalb eines Funktionalprozesses ablaufen. Gibt es Datenbewegungen zwischen zwei Funktionalprozessen, so wurde der Prozess zu stark zerlegt

Funktionsumfangbeschreibungen (FUB)

engl. Functional user requirements (FUR)

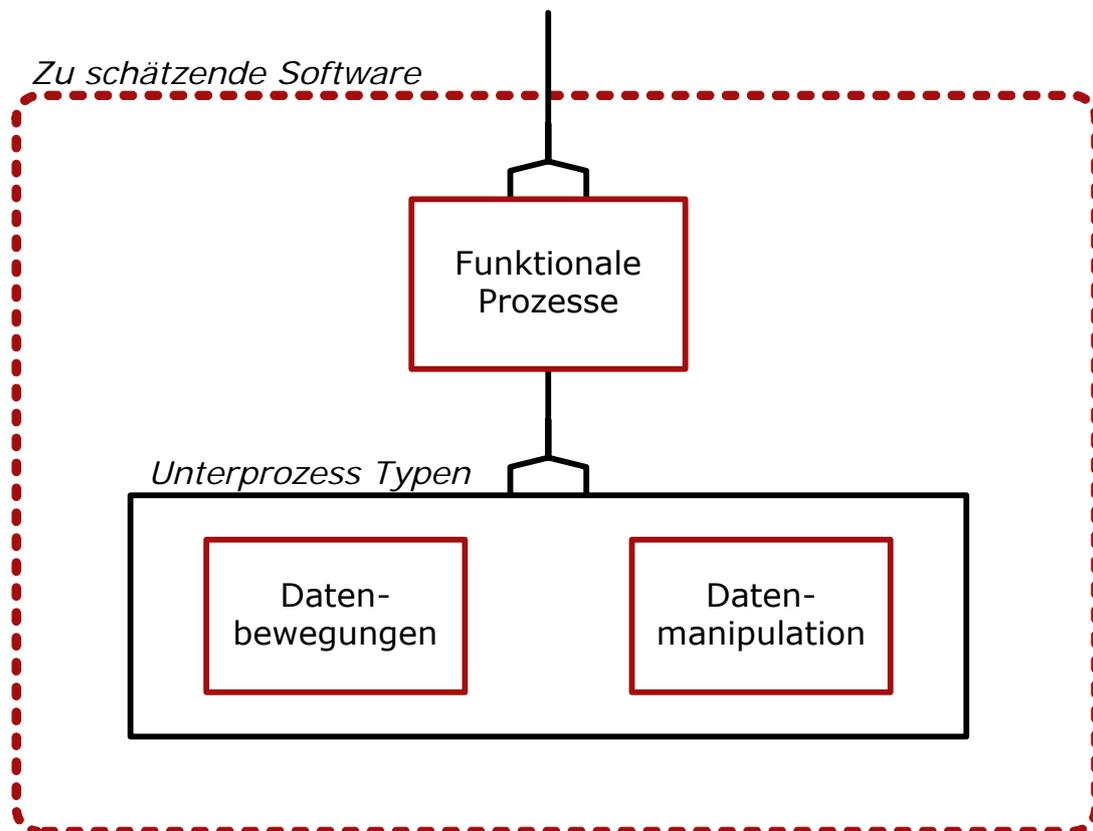


Abbildung 5: Allgemeines Softwaremodell zum Messen der funktionalen Größe (vgl. [COSMIC] S. 22)

Ein funktionaler Prozess besteht aus Datenbewegungen und Datenmanipulationen der bewegten Daten (siehe Abbildung 5). Datenmanipulationen werden in der Version 2.2 der COSMIC-FFP nicht gezählt. Ein Zählverfahren ist jedoch in Planung.

Datengruppen

Eine Datengruppe ist eine eindeutige, nicht leere, nicht redundante unsortierte Menge von Datenattributen. Dabei charakterisiert sie sich u.a. durch ihre Persistenz:

- Transient: Datengruppe existiert nur so lange wie der entsprechende funktionale Prozess, der sie benutzt
- Kurz: Datengruppe existiert, bis die Software beendet wird
- Indefinite: Datengruppe überdauert auch ein Programmende

Eine Datengruppe enthält Daten eines Interessenobjektes (engl. object of interests). Eine Datengruppe kann mit einer Tabelle in der dritten Normalform oder einer Entität gleichgesetzt werden [COSMIC 37].

2.5.3 Die Messphase

In der Messphase gilt es, mittels der in der Mapping Phase ermittelten Datengruppen und Funktionsprozesse die Datenbewegungen zu identifizieren.

Es existieren nur vier verschiedene Arten von Datenbewegungen, die alle gleichermaßen gewichtet werden:

- Read: liest aus einer Datenquelle. Reads sind nur im Backend zu finden.
- Write: schreibt in eine Datenquelle. Writes sind nur im Backend zu finden.
- Entry: eine Eingabe von außen über die Boundary. Entrys sind nur im Frontend zu finden.
- Exit: eine Ausgabe der Software nach außen über die Boundary. Exits sind nur im Frontend zu finden.

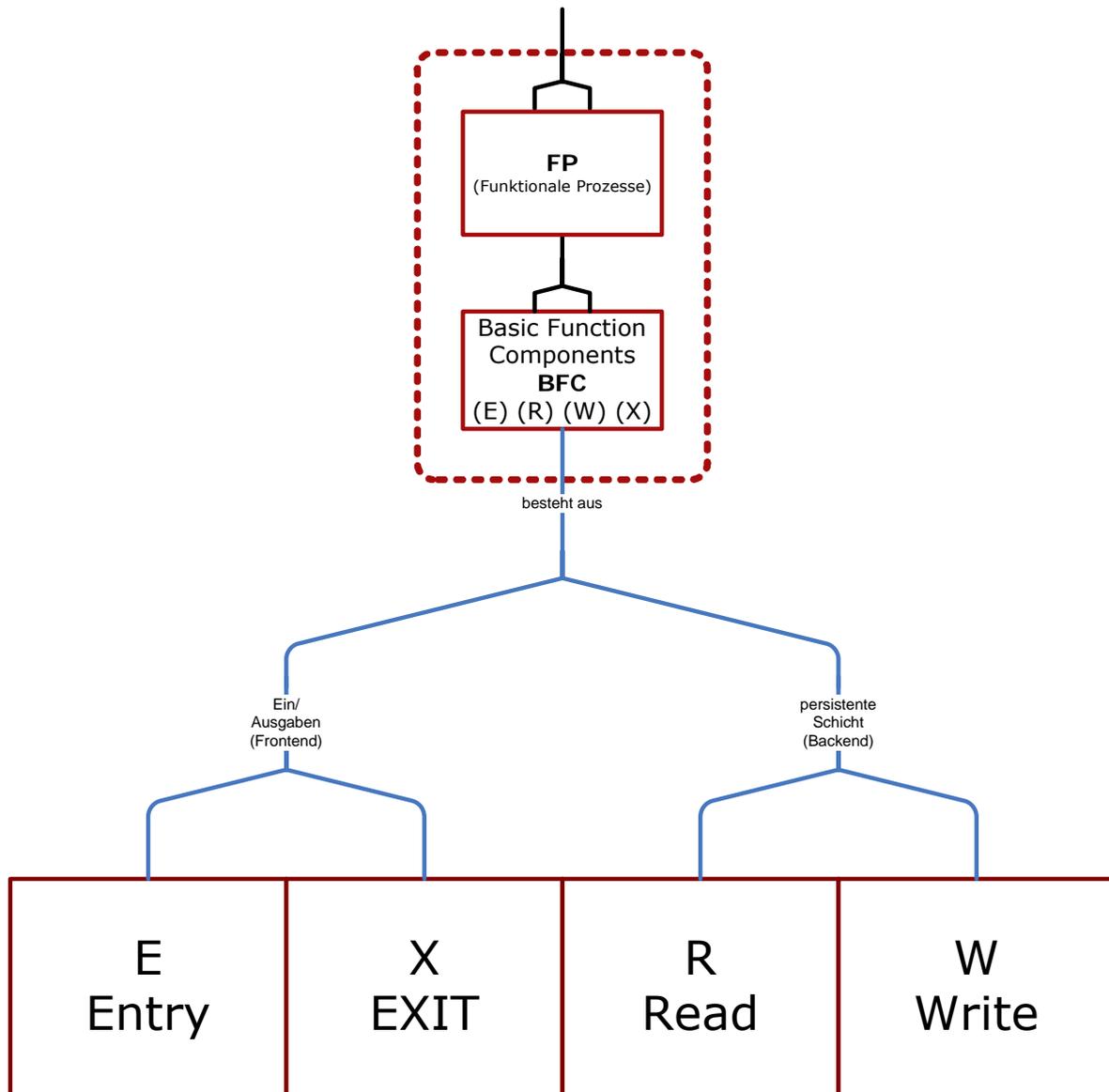


Abbildung 6: Die vier Datenbewegungen im Frontend / Backend

Alle Datenbewegungen werden nun nach den Obenstehenden Arten klassifiziert. Jede einzelne Bewegung erhält die gleiche Gewichtung von einer 'Cosmic functional size unit', kurz Cfsu.

Nun werden alle Datenbewegungen addiert. Die Summe gibt die funktionale Größe der Software an und ist somit proportional zur Anzahl der Datenbewegungen.

$$U = \sum D$$

mit U = Umfang der Software, gemessen in COSMIC-FFP [CFFP]

D = Datenbewegung, gemessen in COMISC-FFP [CFFP]

In Abbildung 1 ist der Ablauf einer Schätzung zu sehen. Um möglichst viele Vergleichsdaten zur Bestimmung zukünftiger Aufwände zu bekommen, ist ein zyklischer Ablauf ähnlich Abbildung 35 sinnvoll.

2.5.4 Zählung von Weiterentwicklungen

Der Aufwand von Weiterentwicklungen wird anhand der hinzugefügten und veränderten Datenbewegungen gezählt.

Vorerst werden alle Werte für Cfsu des Basisprojektes auf Null gesetzt, denn es soll nur der Aufwand für die Weiterentwicklung gezählt werden.

Wird eine Datenbewegung zu einem Funktionalprozess hinzugefügt, so addiert sich ein Cfsu zu dem Umfang der Weiterentwicklung. Wird eine Datenbewegung verändert, so addiert sich ein Cfsu zu dem Umfang der Weiterentwicklung. Wird eine Datenbewegung gelöscht, wird kein Cfsu zu dem Umfang addiert. Das Löschen einer Datenbewegung verursacht laut Cosmic-FFP also keinen Aufwand.

Der Aufwand für die Weiterentwicklung ergibt sich durch die Addition aller veränderten Datenbewegungen und der hinzugefügten Datenbewegungen.

$$U = (\sum N) + (\sum V)$$

mit U = Umfang der Software, gemessen in COSMIC-FFP [CFFP]

N = neue Datenbewegung, gemessen in COMISC-FFP [CFFP]

V = veränderte Datenbewegung , gemessen in COMISC-FFP [CFFP]

Der neue Umfang der Software ergibt sich durch die Summe aller hinzugefügten und veränderten Datenbewegungen minus der gelöschten Datenbewegungen plus dem Umfang des Basisprojektes.

$$U = (\sum N) + (\sum V) - (\sum G) + (\sum A)$$

mit U = Umfang der Software, gemessen in COSMIC-FFP [CFFP]

N = neue Datenbewegung, gemessen in COMISC-FFP [CFFP]

V = veränderte Datenbewegung , gemessen in COMISC-FFP [CFFP]

G = gelöschte Datenbewegung, gemessen in COMISC-FFP [CFFP]

A = Umfang des alten Projektes, gemessen in COMISC-FFP [CFFP]

2.6 Vergleich Widgetpunkte – COSMIC-FFP

Funktionspunktschätzungen unterliegen aufgrund ihrer Gewichtungen subjektiven Meinungen. Eine Schätzung ist also nicht reproduzierbar. Ähnlich den COSMIC-FFP unterliegen die Widgetpunktschätzungen keinen subjektiven Einflüssen, weil sie ohne Gewichtungen sind. Sie sind somit reproduzierbar.

Wie der Name der Methode schon sagt, befasst sich die Methode der Widgetpunkte mit GUI Elementen. Definition: Die Widgetpunkte eines Programms ergeben sich durch Abzählen der Menge aller Widgets in allen programmierten Fenstern des Programms.

Die Methode unterliegt Einschränkungen:

- Widgets sind Elemente graphischer Oberflächen, also muss das Programm eine GUI besitzen
- Ist die GUI (Funktionsumfang) wie bei der Full-Function-Point Methode nicht komplett spezifiziert, so ist eine zuverlässige Schätzung nicht durchführbar.
- Auch hier ähnlich der Full Function Point Methode: die Komplexität der Benutzerschnittstelle muss die der algorithmischen weit übersteigen.

Die Widgetpunkte sind einfacher als die COSMIC-FFP zu zählen. Braucht der COSMIC-FFP Zähler schon wenig Berufserfahrung, so können Widgetpunkte sogar von fachfremden Personen oder maschinell gezählt werden.

Die einzelnen Widgets sind dabei alle Elemente, die in einer GUI zu finden sind wie z.B. Fenster, Eingabefelder, Comboboxen, Menüs, Textfelder, Tabellen, Trennlinien, Scrollbalken usw.

Der Umfang eines Programms ergibt sich durch die Summe aller Widgets der GUI.

Es liegt nahe, die Zählung von Widgetpunkten zu automatisieren. Es müssten im Prinzip nur GUI Elemente eines Programms gezählt werden. In HTML z.B. könnte ein Programm die vorkommenden HTML Tags einer Website zählen.

Die Zählung von Widgetpunkten zur Aufwandschätzung würde sich für die intersoft AG mitunter sehr gut eignen. Die GUIs könnten schon in der Fachkonzeptphase erstellt werden und würden dann automatisch gezählt werden. Der Aufwand für die Zählung wäre daher minimal. Kosten, die sich während der Zählphase auf praktisch Null belaufen, lassen sich zudem besser bei der Unternehmensführung durchsetzen.

Leider eignet sich die Methode nur bedingt für die intersoft AG, da diese auch Software entwickelt, die keine GUI besitzt, wie z.B. Batchprozesse. Mit den COSMIC-FFP hingegen lassen sich alle Arten von Software der intersoft AG schätzen. [KRASEMANN01 309-313]

2.7 Beispielzählung eines abgeschlossenen Projektes von intersoft

Für die Beispielzählung wird ein gut dokumentierter Geschäftsvorfall (GeVo) genommen, der bereits implementiert wurde. Der Geschäftsvorfall enthält bereits ein Zustandsdiagramm. Das Zustandsdiagramm liegt in Form eines endlichen Automaten vor. Dabei stellen die einzelnen Masken der GUI die Zustände dar. Die Schaltflächen in den Masken dienen als Zustandsübergang. Für Schätzer, die mit dem Versicherungsjargon nicht vertraut sind, bleibt somit der Geschäftsvorfall schätzbar. Durch ein Zustandsdiagramm vereinfacht sich auch die Mappingphase (siehe Abbildung 7), da viele Funktionalprozesse schon in dem Diagramm definiert, bzw. leicht zu identifizieren sind.

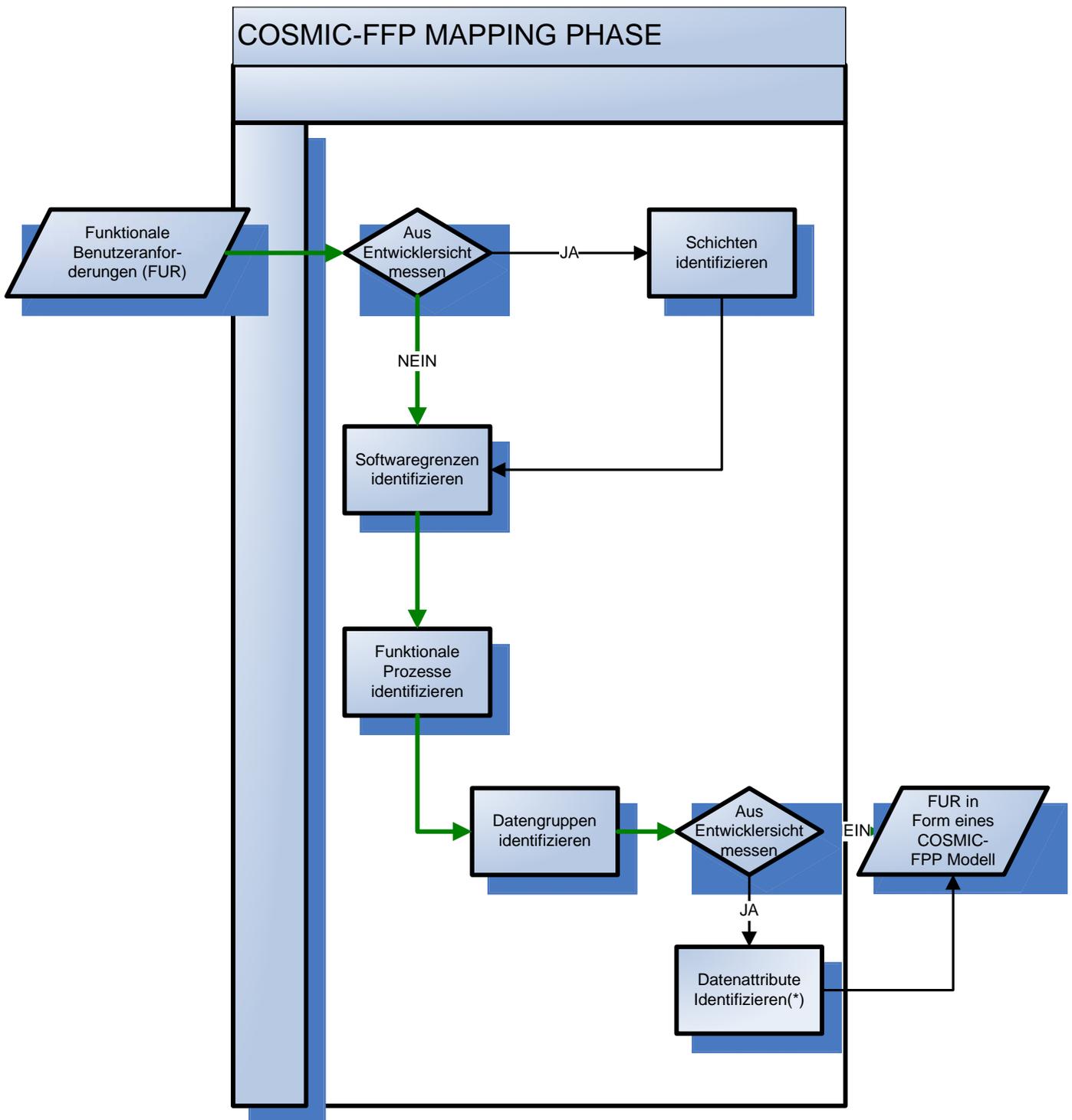


Abbildung 7: Die Mapping-Phase bei der intersoft AG(vgl. Abb. 2)

Die funktionale Benutzeranforderung liegt vor. Eine Zerlegung in Schichten ist nicht erforderlich, da aus Benutzersicht gemessen werden soll (siehe Entscheidungsknoten Abbildung 7).

Als Softwaregrenzen gibt es in unserem Fall aufgrund der fehlenden Schichten nur die Grenze zwischen User und Software.

Als Beispielt dient der Geschäftsvorfall TAB der intersoft AG. Bei dem GeVo TAB handelt es sich um ein klassisches Management-Informationssystem (MIS). Der Nutzer kann

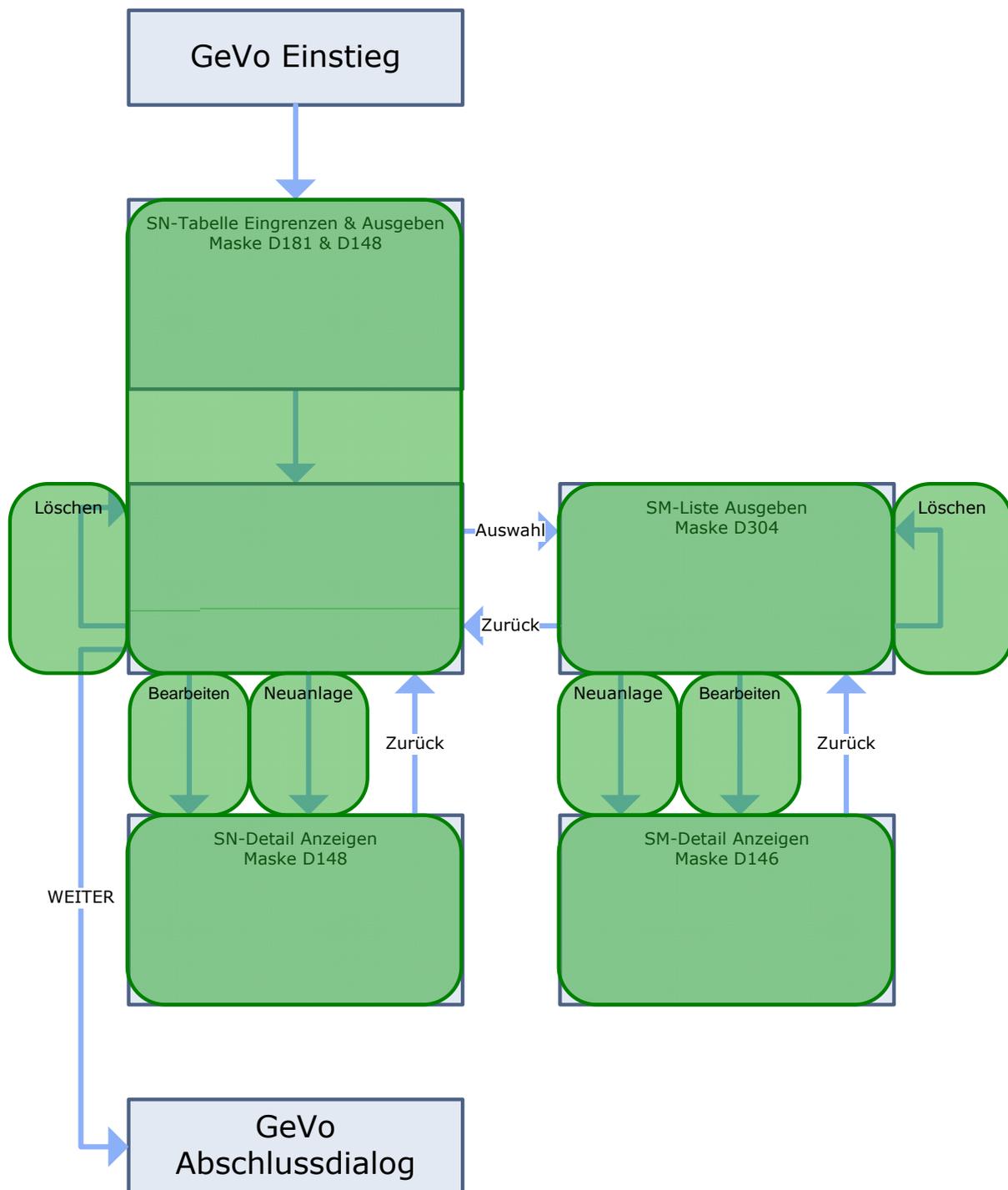


Abbildung 9: Unterteilung des GeVo TAB in Funktionalprozesse

Die grünen Container in Abbildung 9 symbolisieren jeweils einen Funktionalprozess. Man kann erkennen, dass Funktionalprozesse in einigen Fällen über mehrere Masken gehen können. Wir zählen zehn Funktionalprozesse. Der 'GeVo Abschlussdialog' ist kein Funktionalprozess, weil keine Daten bewegt werden. Es handelt sich um eine statische Maske. Jeder der Funktionalprozesse lässt sich in seiner Funktion nicht weiter unterbrechen, ohne mit einem anderen Funktionalprozess Daten austauschen zu müssen.

Alle Funktionalprozesse greifen jeweils nur auf eine Datenbanktabelle und Datengruppe zu. Die Datengruppen befinden sich alle in der dritten Normalform und müssen deshalb nicht zerlegt werden.

Die Mapping Phase (siehe Abbildung 7) ist somit abgeschlossen. Die FUR liegen nun in COSMIC-FFP Form vor.

Löschen eines Eintrages aus der Maske 304 des GeVo TAB

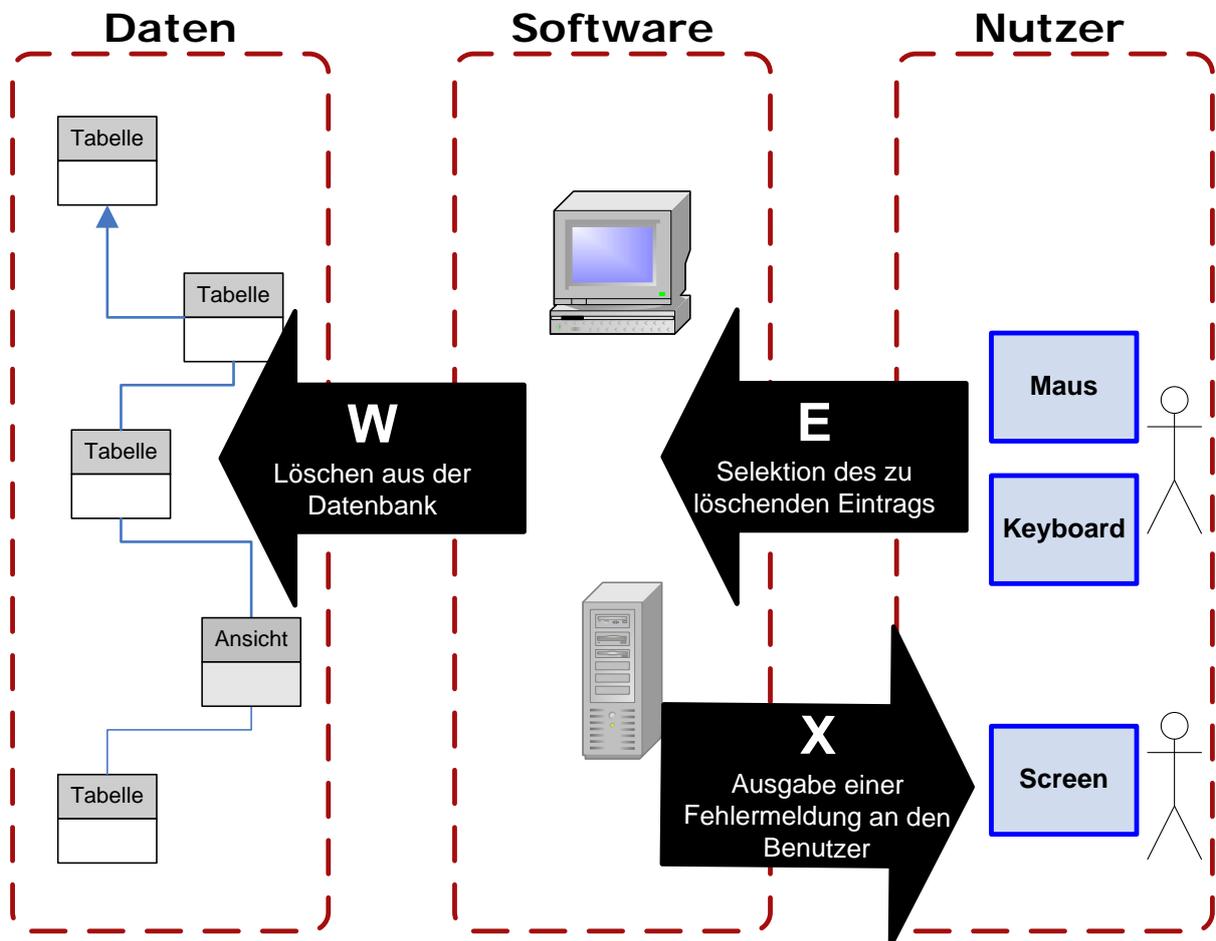


Abbildung 10: Die Datenbewegungen der Löschfunktion des GeVo TAB

Beispiel an einer Löschfunktion:

Für die Löschfunktion muss z.B. ein Eintrag selektiert, der Eintrag aus der Datenbank gelöscht und evtl. eine Fehlermeldung ausgegeben werden.

Nun werden die Datenbewegungen typisiert.

Für das Löschen muss ein Eintrag selektiert werden. Dies ist eine Eingabe eines Benutzers, also ein 'Entry'. Den Eintrag aus der Datenbank zu löschen ist ein 'Write'. Eine eventuelle Fehlermeldung ist eine Ausgabe an den Benutzer als ein 'Exit' (siehe Abbildung 10).

Es handelt sich bei diesem Beispiel um ein typisches Informationssystem mit den klassischen Funktionen: Liste ausgeben, Element neu eingeben, Element verändern, Element löschen. Diese Funktionen haben, sofern sie sich nur auf ein 'Object of Interest' beziehen, drei BFC.

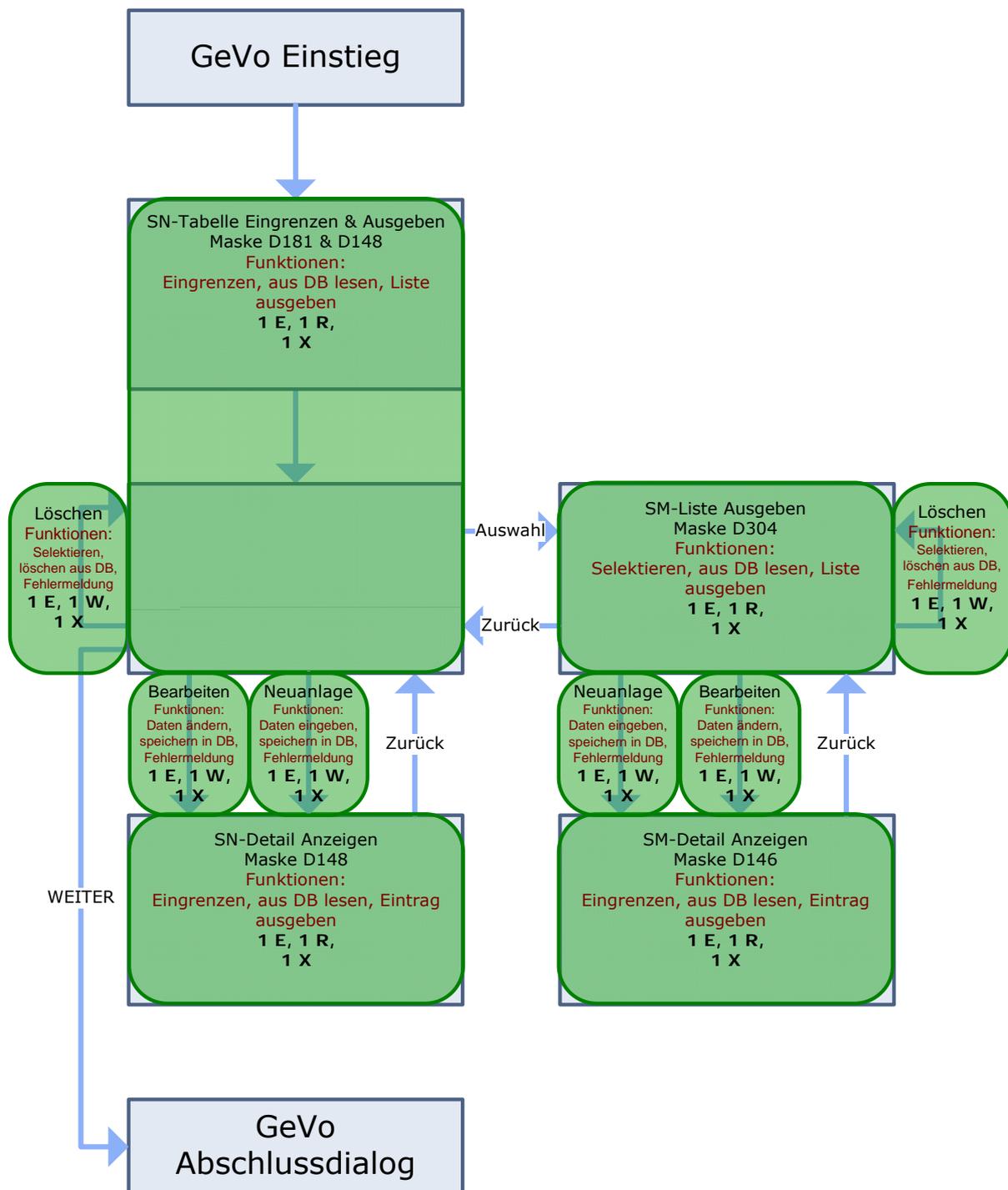


Abbildung 11: Anzahl und Typ der benötigten Basisfunktionskomponenten

Jede Datenbewegung, egal welcher Art, wird mit einer 'Cosmic Functional Size Unit' bewertet (siehe Abbildung 11). Danach werden alle Cfsu addiert. Die Summe ergibt den funktionalen Umfang der Software.

In diesem Beispiel gibt es 30 Cfsu bei zehn funktionalen Prozessen. Die Implementierung hat 313 Personenstunden in Anspruch genommen. Für jede Cfsu wurden also ca. zehn Personenstunden benötigt. Haben zukünftige Entwicklungen den gleichen funktionalen Umfang, so lässt sich eine erste vage Prognose über die benötigten Personenstunden machen. Für fundierte statistische Prognosen bedarf es aber mehrerer Vergleichsdaten.

2.8 Beispielzählung einer graphischen Oberfläche mit Widgetpunkten

Beispiel: Geschäftsvorfall TAB der intersoft AG, Maske SN Tabelle eingrenzen

2.8.1 Zählen von Widgetpunkten

Die Widgetpunkte bedürfen gegenüber den COSMIC-FFP keiner Mappingphase. Das einzige Kriterium für die Zählung von Widgetpoints sind komplett definierte graphische Oberflächen. Fehlen den Masken noch Elemente, so kann der Umfang natürlich auch nicht genau bestimmt werden.

Widgetpunkte zu zählen ist relativ einfach. Es gibt eine feste Definition von einigen wenigen Widgets. Man versucht, zu jedem Element in der graphischen Oberfläche, ein passendes Widget aus der Definition zu finden (siehe Tabelle 25).

Kategorie	Widgets
Rahmen-Widgets	Fenster und Notizbücher
Eingabe-Widgets	Eingabefeld (auch für Ausgaben), ComboBox, Menüknopf, Radioknopf, Aktionsknopf, CheckBox, TextFeld
Beschreibende Widgets	Label, Trennlinie, Gruppierungsbox
Zusammengesetzte Widgets	Tabelle, Liste, Rollbalken
Menü	Das Menü selbst und jeder Menüeintrag

Tabelle 25: Die verschiedenen Kategorien von Widgets (vgl. [KRASEMANN01] S. 311)

Ist man sich unschlüssig, ob ein Widget in eine bestimmte Kategorie gehört, so kann man es einfach irgendeiner Kategorie zuordnen, da alle Widgets gleich schwer gewichtet werden. Die Zuordnung zu den Kategorien wäre für die Umfangsmessung also unbedeutend.

2.8.2 Beispielzählung anhand des GeVo TAB

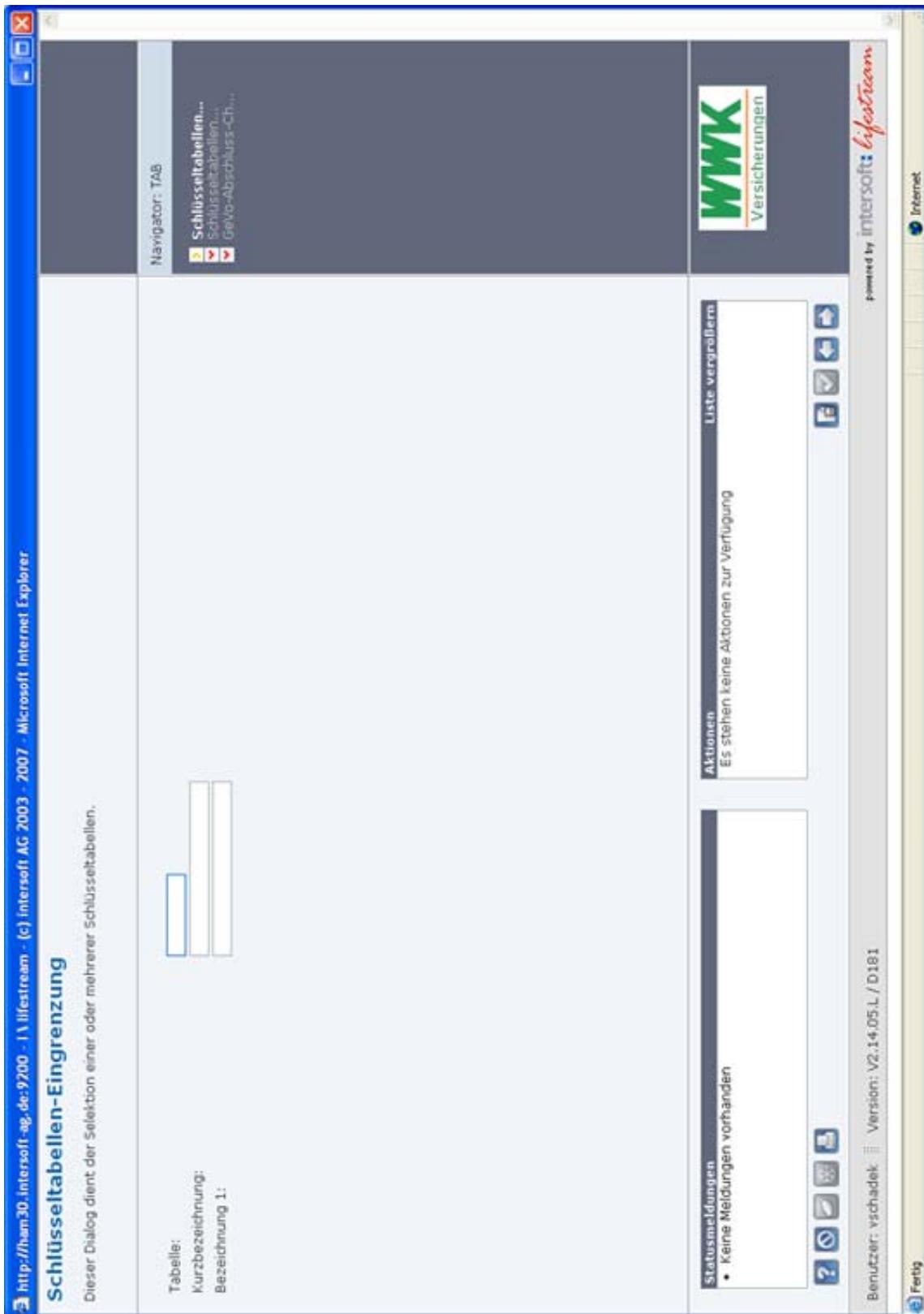


Abbildung 12: Graphische Oberfläche des Geschäftsvorfalles TAB

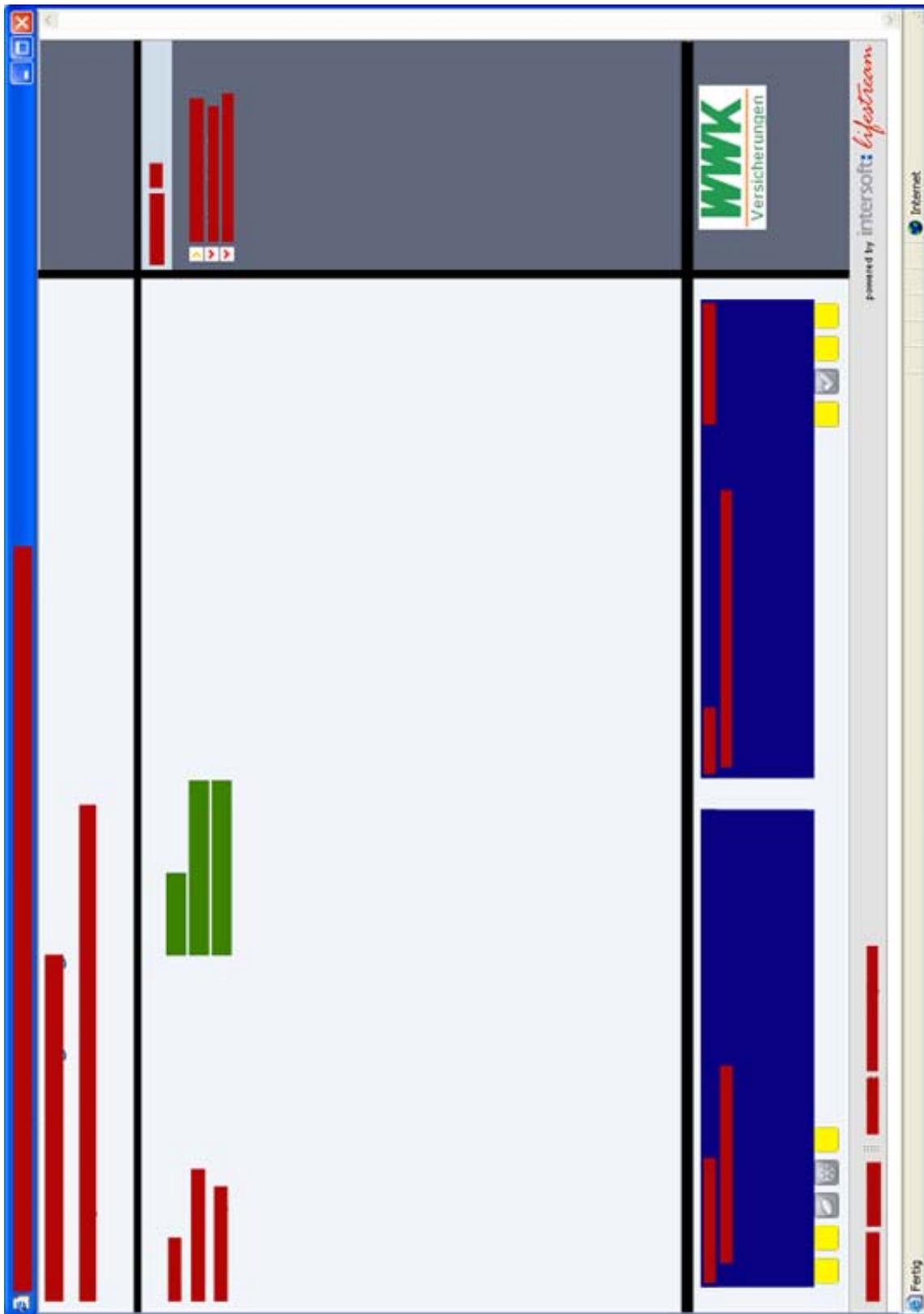


Abbildung 13: Vorkommende Widgets farblich markiert

Alle Widgets wurden farblich markiert (siehe Abbildung 12 und Abbildung 13). An der Farbe der Markierung (siehe Tabelle 26) ist auch die Art des Widgets zu erkennen. Diese Methode erleichtert das Zählen der unterschiedlichen Arten von Widgets und verhindert zudem Doppelzählungen.

Legende

Farbe	Widget Beschreibung	Widget Gruppe
Gelb	Buttons	Eingabe-Widgets
Grün	Eingabefeld	Eingabe-Widgets
Rot	Label	Beschreibende Widgets
Blau	Tabelle / Liste	Zusammengesetzte Widgets
Orange	Radiobuttons / Checkboxes	Eingabe-Widgets
Schwarz	Trennlinie	Beschreibende Widgets

Tabelle 26: Legende der Zählung von Widgets

Probleme bei der Bewertung der Widgets

Da es kein Measurement Manual für die Widgets gibt, gab es einige Unklarheiten bei der Bewertung der Widgets. Folgende Unklarheiten wurden wie folgt interpretiert:

1. Dynamische Elemente in einer ListBox oder ComboBox werden nicht einzeln gezählt. Lädt eine ComboBox 100 Elemente aus der Datenbank, so wird nur ein Element gezählt. Setzt sich das Listenelement aus mehreren Elementen zusammen, so werden diese einzeln gezählt.
2. Labels, die aus dynamischen und statischen Elementen bestehen, werden nicht als ein Widget gezählt. Es wird das statische Element und das dynamische Element jeweils einzeln gezählt.
3. Inaktive Buttons werden nicht gezählt

2.8.3 Ergebnis der Zählung

Name der Oberfläche	Beschreibende Widgets	Eingabe-Widgets	Rahmen Widgets	Zusammengesetzte Widgets	Menü	Summe
SN-Tabelle eingrenzen	23	9	1	2	0	35

Tabelle 27: Ergebnisse der Zählung von Widgets

Die Oberfläche 'SN-Tabelle eingrenzen' hat 35 Widgetpunkte.

Ist eine volle Unterteilung in die verschiedenen Widget Gruppen möglich, so kann man weitere Metriken / Projektcontrolling mit den Daten erstellen.

Es ließen sich Fehleranalysen über eine Zählung fahren. So könnte man Suchalgorithmen entwickeln, die unlogische Situationen ausschließen können. Eine graphische Maske die z.B. Eingabefelder enthält, aber keinen Button, um die Eingaben abzusenden, wäre fehlerhaft. Der Fehler wäre dann im Design der Maske zu suchen oder aber in der Zählung.

2.8.4 Vergleich zu COSMIC-FFP

Zur besseren Vergleichbarkeit wurden zwei weitere Projekte mit COSMIC-FFP und Widgetpoints gezählt.

GeVo RECHT	Beschreibende Widgets	Eingabe-Widgets	Rahmen Widgets	Zusammenges. Widgets	Menü	Summe		
Rollenzuordnungsliste	30	10	1	3	0	44	Summe Widgets	241
Rollenzuordnungs-Details	27	11	1	2	0	41		
							Summe	
Rollen-Liste	24	11	1	3	0	39	COSMIC-FFP	40
Objektberechtigung-Liste	30	9	1	3	0	43		
Objektberechtigung-Details	40	31	1	2	0	74	Personenstunden	271
GeVo DAUFT	Beschreibende Widgets	Eingabe-Widgets	Rahmen Widgets	Zusammenges. Widgets	Menü	Summe	Summe Widgets	245
DA Details	39	31	1	2	0	73		
							Summe	
DA-Liste	31	10	1	3	0	45	COSMIC-FFP	46
Eingrenzung	32	29	1	2	0	64		
Konto Liste	49	10	1	3	0	63	Personenstunden	218
GeVo TAB	Beschreibende Widgets	Eingabe-Widgets	Rahmen Widgets	Zusammenges. Widgets	Menü	Summe		
SM-Detail	41	30	1	2	0	74	Summe Widgets	296
SM-Liste	39	17	1	3	0	60		
							Summe	
SN-Detail	42	29	1	2	0	74	COSMIC-FFP	30
SN-Liste	35	14	1	3	0	53		
SN-Tabelle eingrenzen	23	9	1	2	0	35	Personenstunden	313

Tabelle 28: Gegenüberstellung Widgetpoints - Cosmic-FFP

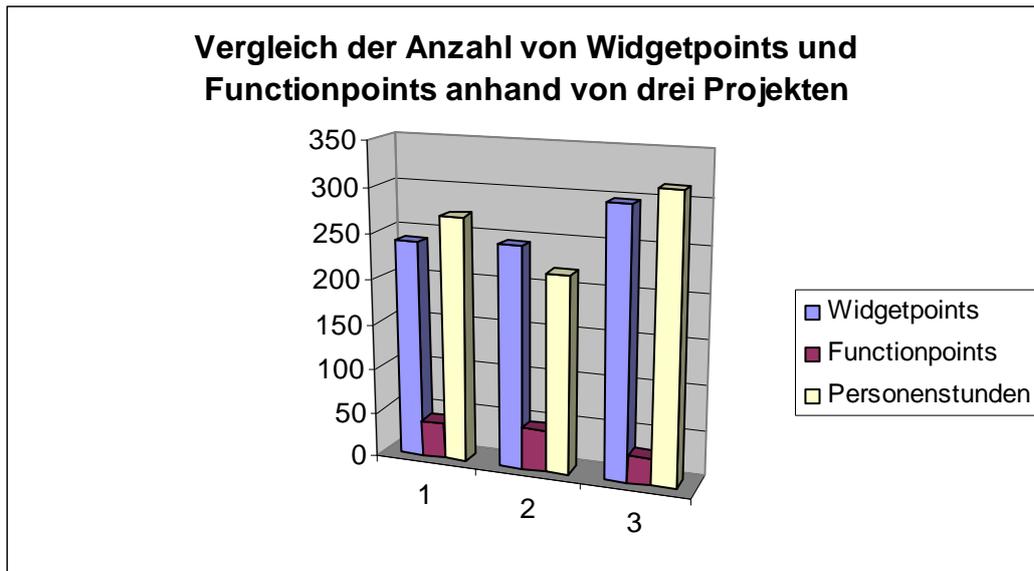


Abbildung 14: Gegenüberstellung der Zählung von Widgetpoints und COSMIC-FFP

Legende

X-Achse: Nummer des Beispielprojektes

Y-Achse: Anzahl der FP, Anzahl der Personenstunden

Der Graph in Abbildung 14 deutet an, dass zwischen den Widgetpoints und den tatsächlich benötigten Personenstunden eine höhere Korrelation besteht, als zwischen COSMIC-Functionpoints und den tatsächlich benötigten Personenstunden. Für eine genauere Aussage ist es zwingend nötig, weitaus mehr Projekte in die Auswertung mit einzubeziehen.

Beachtet wird nur, ob der Verlauf der Kurve sich ähnelt, nicht aber, wo die Kurve im Graphen positioniert ist. Dies ist aufgrund der unterschiedlichen Verfahren selbstverständlich.

2.8.5 Offene Fragen Widgetpoints

Die Widgetpunkte-Methode scheint durch Krasemann [KRASEMANN01] recht ausführlich beschrieben. Beim tatsächlichen Messen kommen jedoch viele Fragen auf, die durch Krasemann nicht beantwortet werden können. Eine Suche nach einer genauen Schätzanleitung blieb erfolglos. Folgende Fragen sind während des Schätzens aufgekomen:

Graphische Labels

Ein Button mit Textlabel bekommt zwei Widgetpunkte. Einen für den Button, einen für das Label. Ein graphischer Button ist laut Definition aber nur ein Widgetpunkt für den Button. Der Informationsgehalt ist aber der gleiche, da z.B. ein Label durch ein Symbol oder ähnliches dargestellt werden kann. Dabei ist der Aufwand für die Erstellung eines graphischen Buttons höher.

Inhalt (Elemente) von ComboBoxen / ListBoxen

Wie wird gezählt, wenn Widgets, bei denen aus mehreren Elementen gewählt werden kann (ComboBox, ListBox), dynamisch gefüllt werden? Eine Zuordnung von Widgets pro Listenelement ist dann nicht möglich, weil man während der Zählung nicht weiß, wie viele Elemente geladen werden. Die Implementierung für den Füllmechanismus ist auch konstant und steigt nicht mit der Menge der zu ladenden Elemente. Bsp.: Der

Implementierungsaufwand für eine ListBox, die fünf oder 1000 Elemente aus einer Datenbank lädt und darstellt, ist immer gleich.

Modularität / Wiederkehrende Elemente

Was ist mit wiederkehrenden GUI Elementen? Werden diese auch dann voll gezählt, wenn sie keinen Implementierungsaufwand verursachen [KRASEMANN01 S.312]? Bsp.: Zwei Oberflächen haben die gleichen wiederkehrenden Elemente, die mit je 100 Widgetpunkten bewertet werden. Die eine Oberfläche Nr.1 hat einen Widgetpunkt, Oberfläche Nr.2 hat zwei Widgetpunkte. Vernachlässigt man, dass es wiederkehrende Widgetpunkte gibt, so hat die eine Oberfläche 101 Widgetpunkte, die andere 102. Der Aufwand der Implementierung wäre also fast identisch. Zählt man aber wiederkehrende Widgetpunkte / Module nicht mit, weil diese bereits implementiert wurden, so hat Oberfläche Nr.2 doppelt so viele Widgetpunkte wie Oberfläche Nr. 1.

Zählen von Ausgaben

Pro Funktion und Zeile in einer Tabelle wird ein Widgetpunkt gezählt. Bsp.: Eine Tabelle bietet pro Zeile fünf Widgets und lädt 200 Zeilen, so werden 1000 Widgets gezählt. Programmiert werden muss aber nur die eine Zeile.

Trennlinien

Trennlinien werden in Zeiten voll graphischer Benutzeroberflächen nicht mehr vom Programmierer gesetzt, sondern von einem Grafiker. Trennlinien werden deswegen nicht mehr als Modul einer Programmiersprache, sondern als Grafik implementiert. Trennlinien zu deuten wäre wieder subjektiv. Die Widgetpunkt-Methode behauptet aber von sich, dass sie objektiv sei. Auch eine automatisierte Zählung könnte graphische Trennlinien nicht oder nur schwer erfassen.

Ein genaues Messen ist wegen dieser offenen Fragen leider nicht möglich. Zu oft muss nach eigenem Ermessen entschieden werden, wodurch die Methode ihre Objektivität verliert. Prinzipiell können diese Definitionslücken durch eigene Regeln geschlossen werden. Dies wurde aber vernachlässigt, da sich die Methode nicht für die Anforderungen der Zählbarkeit von Hintergrundprozessen eignet.

2.9 Fazit

Die Umfangsmessung aus funktionaler Sicht eignet sich für die intersoft AG am besten. Einzige Voraussetzung für die Schätzung ist ein komplett beschriebenes Fachkonzept. Somit kann zu einem sehr frühen Zeitpunkt gemessen werden. Liegt kein ausreichend beschriebenes Fachkonzept vor, so bleibt die Möglichkeit, den Umfang über die Widgetpoint Methode zu messen. Einzige Voraussetzung hierfür ist eine komplett beschriebene graphische Oberfläche der Software.

Ein Vergleich mit Widgetpunkten hat ergeben, das auch dieses Verfahren sich für die intersoft AG eignen würde, wenn diese nicht auch Projekte ohne graphische Oberfläche hätte.

Unter den funktionalen Schätzmethode hat sich die COSMIC-FFP Methode als am besten geeignet für die intersoft AG herausgestellt.

Im Weiteren (Kapitel 3) muss geprüft werden, welche Anforderungen intersoft an ein Tool zur Erfassung und Analyse von COSMIC-FFP hat. Eine Marktanalyse soll klären, ob es kommerzielle Tools gibt, die diesen Anforderungen entsprechen können.

3 Anforderungen an eine Software zur Erfassung und Auswertung von Cosmic FFP

3.1 Einleitung

Das passende Verfahren zur Aufwandschätzung wurde gefunden. Es gilt, Kriterien an eine Software zu stellen, die den Prozess des Erfassens und Analysierens unterstützen soll.

Die Anforderungen der intersoft AG an eine Software für die Erfassung und Analyse von COSMIC-FFP sind die folgenden:

3.2 Kriterien

Erfassung von COSMIC-FFP Projekten

Das Programm muss COSMIC-FFP erfassen und zentral speichern können. Auf die Daten müssen mehrere Benutzer gleichzeitig zugreifen können. Beispiel: die Analysten und Zähler benutzen zur selben Zeit die gleiche Datenbasis. Dabei muss das Tool den verschiedenen Benutzergruppen Vertraulichkeit bieten, so dass z.B. Teams nicht auf Informationen zugreifen können, die Vergleiche der Produktivität zwischen den Teams zulassen. Auch sollte das Tool mandantenfähig sein, so dass Kunden Zugriff auf eigene Schätzungen haben, nicht aber Schätzungen anderer Firmen einsehen können.

Fremde Applikationen haben die Möglichkeit, über ein Interface oder eine Exportfunktion auf die Daten zuzugreifen.

Erfassung von Projektdaten

Die Software sollte eine Vielzahl von Projektmerkmalen erfassen. Mit Hilfe dieser Merkmale lassen sich später verschiedene Analysen durchführen.

Schätzen und Analysieren

Mit der Software sollten sich Schätzprognosen zur Bestimmung der Projektdauer aufstellen lassen. Des Weiteren sollten verschiedene Analyseverfahren zum Projektcontrolling und Benchmarking zur Verfügung stehen.

Erweiterbarkeit

Eine Software zu finden, die in allen Kriterien voll den Anforderungen entspricht, ist unwahrscheinlich. Gerade bei den benötigten Analyseverfahren wird es wahrscheinlich starke Abweichungen geben. So sollten fehlende Funktionalitäten vom Kunden selbst nachgepflegt werden können.

Kosten

Ziel ist es, die Kosten durch Lizenzen möglichst gering zu halten.

3.3 Fazit

Die Anforderungen an eine prozessunterstützende Software sind definiert. Im folgenden Kapitel werden bereits verfügbare Tools auf ihre Eignung für die intersoft AG geprüft.

4 Evaluation vorhandener Software zur Erfassung und Analyse von COSMIC-FFP

4.1 Einleitung

Nach den Kriterien aus Kapitel 3.2 wird auf dem freien Markt verfügbare Software bewertet.

Diese Kriterien werden mit einer vierstufigen Skala bewertet (siehe Tabelle 29).

Legende

Bewertung	Beschreibung
+++	Die Anforderung wird voll erfüllt
++	Die Anforderung wird erfüllt
+	Die Anforderung wird begrenzt erfüllt
O	Die Anforderung wird nicht erfüllt

Tabelle 29: Legende der Bewertung von COSMIC-FFP Tools

4.2 Bewertung der Software

MeterIT-Cosmic

Die Firma TELMACO bietet speziell für die Erfassung und Analyse von COSMIC-FFP eine Software an.

Ein Test der 30-Tage Trial Version entsprach jedoch nicht der Produktbeschreibung. Das Tool war trotz der Vorkenntnisse in COSMIC-FFP-Schätzungen nicht zu bedienen. Die vom Tutorial versprochenen Funktionen waren nicht zu aktivieren oder führten zum Absturz der Software. Die Gestaltung der Oberfläche und Useability waren ebenso wenig befriedigend.

Preis für eine Einzellizenz: 350€

Eine Eignung für intersoft ist nicht gegeben.

MeterIT-Cosmic

Kriterium	Beschreibung
Erfassung COSMIC-FFP	+
Erfassung Projektdaten	+
Schätzen und Analysieren	+
Erweiterbarkeit	O
Kosten	+

Tabelle 30: Bewertung des Tools MeterIT-Cosmic

Experience Pro 3.1

Die Software bietet Projektschätzungen, Projektcontrolling, Softwarezählungen und Datenverwaltung. Unter anderem unterstützt die Software ISBSG Projektdaten und weitere Messverfahren.

Keine Preisangaben verfügbar.

Keine Testversion verfügbar.

Eine Eignung für intersoft ist nicht gegeben.

Experience Pro 3.1

Kriterium	Beschreibung
Erfassung COSMIC-FFP	++
Erfassung Projektdaten	++
Schätzen und Analysieren	0
Erweiterbarkeit	0
Kosten	0

Tabelle 31: Bewertung des Tools Experience Pro 3.1

KnowledgePLAN

Die Produktbeschreibungen von KnowledgePLAN sind sehr vielversprechend. Die Software bietet die Möglichkeit zur Erfassung einer Vielzahl von Zählmethoden. Tools zum Projektcontrolling, Benchmarking und Schätzverfahren sind auch vorhanden.

Die Software ist kostenpflichtig, aber zu den Kosten werden keine Angaben gemacht

Eine Testversion ist nicht verfügbar.

Laut Beschreibung eignet sich die Software für die Zwecke der intersoft AG. Mangels einer Testversion ist die Eignung für die intersoft AG leider nicht gegeben.

KnowledgePLAN

Kriterium	Beschreibung
Erfassung COSMIC-FFP	++
Erfassung Projektdaten	++
Schätzen und Analysieren	+++
Erweiterbarkeit	0
Kosten	0

Tabelle 32: Bewertung des Tools KnowledgePLAN

Siesta

Die Software bietet eine umfangreiche Umgebung zur Erfassung von COSMIC-FFP. Die Oberfläche ist übersichtlich gestaltet und leicht zu bedienen. Leider bietet das Tool keine Funktionen zur Analyse. Auch werden die Daten nicht zentral für mehrere Benutzer gehalten, sondern einzeln in Dateien gespeichert.

Die Software ist kostenfrei.

Eine Eignung für intersoft ist nicht gegeben.

Siesta

Kriterium	Beschreibung
Erfassung COSMIC-FFP	++
Erfassung Projektdaten	+++
Schätzen und Analysieren	0
Erweiterbarkeit	0
Kosten	+++

Tabelle 33: Bewertung des Tools Siesta

Eigenentwicklung

Eine Eigenentwicklung hat den Vorteil, dass die Software nach den Anforderungen der intersoft AG entwickelt werden kann. Dem gegenüber stehen die Kosten für eine Entwicklung.

Hohe Entwicklungskosten, keine Lizenzkosten.

Eine Eigenentwicklung von und für die intersoft AG wäre hinsichtlich aller erforderlichen Kriterien für die intersoft AG geeignet.

Eigenentwicklung

Kriterium	Beschreibung
Erfassung COSMIC-FFP	+++
Erfassung Projektdaten	+++
Schätzen und Analysieren	+++
Erweiterbarkeit	+++
Kosten	0

Tabelle 34: Bewertung des Tools bei Eigenentwicklung

CosmicXpert, COSMIC-RUP, ROSE, XForms-Format Ontological Formalization und OO-Method RMFFP waren leider nicht mehr im Internet zu finden.

4.3 Zusammenfassung der Bewertungen

Zusammenfassung

Anforderung	MeterIT-Cosmic	Experiance Pro	Knowldge Plan	Siesta	Eigenentwicklung
Erfassung COSMIC-FPP	+	++	++	++	+++
Erfassung Projektdaten	+	++	++	+++	+++
Schätzen und Analysieren	+	0	+++	0	+++
Erweiterbarkeit	0	0	0	0	+++
Kosten	+(350€)	0 (keine Angabe)	0 (keine Angabe)	+++ (kostenfrei)	0 (hohe Entwicklungskosten)
Gesamtbewertung	+	+	+(+)	+(+)	++(+)

Tabelle 35: Zusammenfassung der Bewertung von COSMIC-FPP Tools

4.4 Fazit

Von den getesteten Tools konnte keines die Anforderungen der intersoft AG an ein prozessunterstützendes Tool erfüllen. Von der oben genannten Software eignet sich laut Bewertungsschema eine Eigenentwicklung am besten für die intersoft AG.

Im folgenden Kapitel werden die Anforderungen an eine Eigenentwicklung noch genauer definiert.

5 Konzeption eines Tools

5.1 Einleitung

Laut Fachliteratur mindert Toolunterstützung die Akzeptanzprobleme, die bei der Einführung von Aufwandschätzungen auftreten könnten. *Man könne nicht mit Papier und Bleistift zu den Projekten gehen, die die neuste Technologie einsetzen* [BUNDSCHUH04 S.190].

Am Anfang des Kapitels werden die fachliche und technische Architektur gemäß der Anforderungen aus Kapitel 3.2 definiert. Danach wird geprüft, welche technischen Mittel sich für die Realisierung eignen.

5.1.1 Fachliche Architektur

Benutzerrollen:

Messer (Zähler)

- Schnelle und unkomplizierte Dokumentation der gemessenen Functionpoints.
- Schnelle und unkomplizierte Erfassung der geleisteten Personenstunden.
- Kann Projekte fremder Teams öffnen, um eigene Unklarheiten zu klären. Hat dabei keinen Zugriff auf Daten die Performancevergleiche zulassen.
- Hat keinen Zugriff auf Benchmarks und Analysen.

Schätzer

- Einfaches Ablesen der Umfangsprognose eines Projektes.
- Graphische Metriken, die das Projektcontrolling ermöglichen.

Kunde

- Kann Funktionsumfang sowie Schätzprognosen seiner Projekte einsehen.
- Darf keine fremden Projekte oder Analysen einsehen.
- Darf keine tatsächlich benötigten Personenstunden einsehen, da der Kunde bei Überschätzung geneigt sein könnte, sein Geld zurückzufordern.

Allgemeine Kriterien:

- Keine Inkonsistenzen durch mehrfache, dezentrale Datenhaltung. Zentrales Speichern aller Daten in einer Datenbank.
- Automatische Erstellung der graphischen Metriken.
- Vergleiche zwischen historischen Projektdaten für Analysen und Projektcontrolling.
- Vergleiche zwischen Teilmengen aller Projekte für Analysen und Projektcontrolling.
- Graphisches Userinterface zur Dokumentation der FP.
- Vertraulichkeit zur Vermeidung von Produktivitätsvergleichen zwischen den Teams. Die Teams können andere Projekte zur Klärung von Unklarheiten einsehen. Zugriff auf Daten, die Schlüsse auf Produktivität geben, bleiben jedoch verwehrt.
- Übersichtliche Baumstruktur zur Darstellung eines Projektes mit seinen Unterknoten. Die hierarchische Struktur von Funktionalprozessen und Basisfunktionskomponenten kann so einfach und übersichtlich dargestellt werden.
- Vereinen aller Funktionen in einem Tool mindert Aufwand und Fehler durch redundante Datenhaltung.

Funktionsumfang der Benutzerrollen:

Messer (Zähler)

- Anlegen von Projekten und Aufbauprojekten
- Öffnen von Projekten und Aufbauprojekten
- Anlegen von Funktionalprozessen (FP)
- Anlegen von Basisfunktionskomponenten (BFC)
- Erfassen von Personenstunden

Schätzer

- Kompletter Funktionsumfang des Schätzers
- Selektierung von einzelnen oder mehreren Projekten, die mit historischen Werten verglichen werden können
- Aufwand aus der Schätzkurve eines Graphen ablesen. Graphen basieren auf historischen Daten.
- Projektcontrolling
 - o Auswertung der BFC Verteilung, Vergleich zu älteren Projekten
 - o Fortschritt an Cfsu gemessen an der geplanten Projektdauer
 - o Fortschritt an abgearbeiteten Cfsu pro geplante Personenstunden
 - o Abweichung geplanter Cfsu von tatsächlich benötigten Cfsu

Anwendungsfälle:

Messer (Zähler)

- Legt im Tool ein aktuell zu schätzendes Projekt an und gibt die Projektdaten mit den Funktionalprozessen und den Basisfunktionskomponenten ein
- Legt historische Projekte im Tool an, um die Anzahl der Vergleichswerte zu verbessern
- Öffnet / lädt Projekt
- Erfasst Aufbauprojekt (ähnlich Neuanlage)
- Öffnet /lädt Aufbauprojekt
- Erfasst Projekttermine: Implementierungsbeginn, geschätztes Ende der Implementierung, tatsächliches Ende der Implementierung
- Erfasst tatsächlich benötigte Personenstunden

Schätzer

- Prognostiziert mittels Schätzkurve den Aufwand eines Projektes
- Prüft mittels BFC Verteilungsvergleich, ob der Messer Projekte richtig gezählt hat
- Kontrolliert während der Implementierung, ob der Abgabetermin eingehalten werden kann und ob das Projekt mehr Personenstunden benötigt

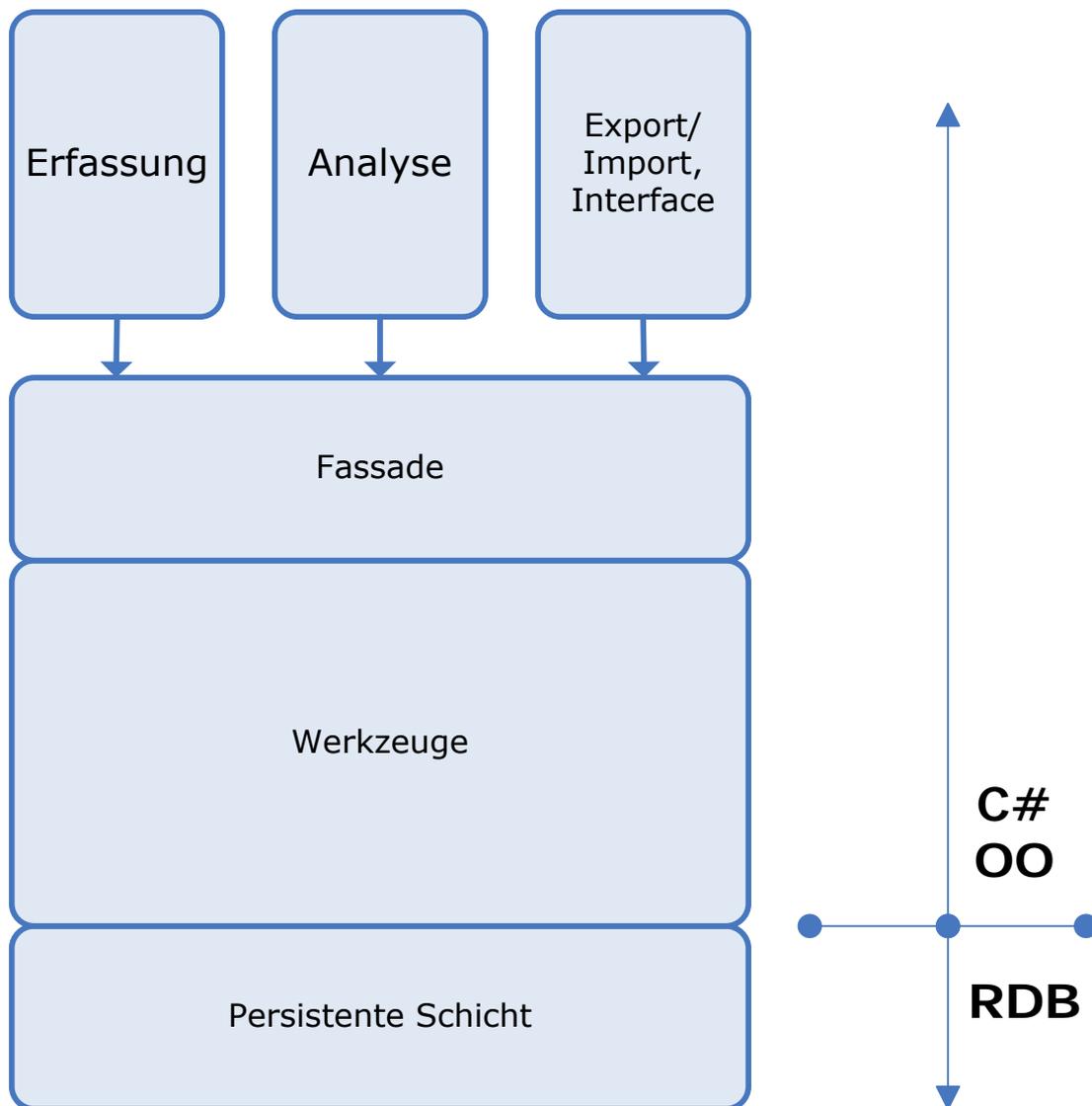


Abbildung 15: Fachliche Architektur

Da bei der Integration sowohl die Funktionalität der Werkzeuge als die Art der Datenhaltung eine entscheidende Rolle spielen, wurde das Tool nach funktionsorientierten und datenorientierten Ansätzen implementiert.

Die Daten der Projekte werden entsprechend persistent abgelegt, so dass sich andere Applikationen ohne weiteres der Datenbasis bedienen können. Die Werkzeuge bereiten die Daten so auf, dass der Benutzer Analysen und Projektcontrolling mit ihnen fahren kann.

Die Fassade bietet nach außen eine vereinfachte Schnittstelle der technischen Werkzeuge.

Die Werkzeuge wurden in C# implementiert. Die Objektorientierung von C# wurde nicht verwendet.

5.1.2 Technische Architektur

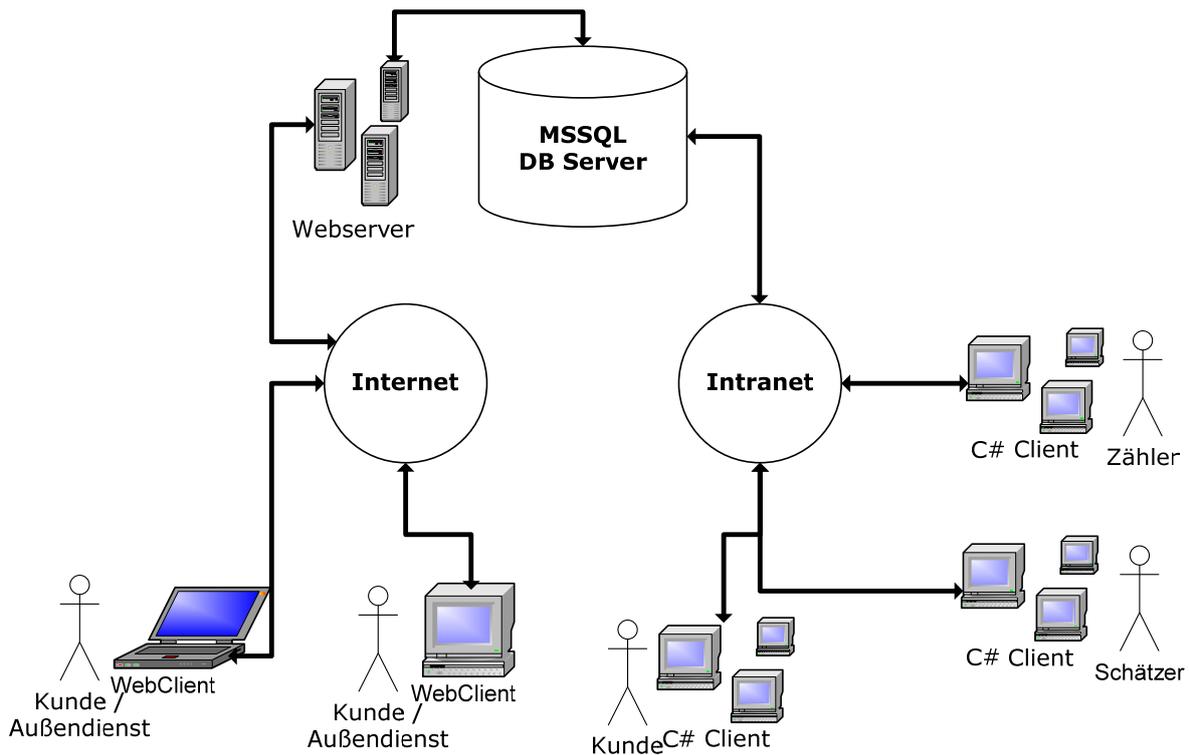


Abbildung 16: technische Architektur

Alle Benutzer sollen aus verschiedenen Arbeitsplatzsituationen auf das Tool zugreifen können. Die Benutzer können über eine bestehende Internetverbindung das Tool benutzen, welches sich über das Intranet / Internet mit dem Datenbankserver verbindet. Steht das Tool nicht zur Verfügung, so sollen die Daten über einen Webclient pflegbar sein (siehe Abbildung 16).

Die Programmlogik des Intranetclients steckt in dem Client. Eine Kontrolle prüft bei jeder Neuverbindung zum Datenbankserver, ob die Version des Clients veraltet ist. Für den Webclient muss eine Serveranwendung entwickelt werden.

Da der Webclient noch nicht geplant ist, wird er in der Architektur nicht berücksichtigt.

5.1.2.1 Daten persistent ablegen

Die Daten werden in einer relationalen Datenbank gespeichert. Diese ist zentral und von mehreren Usern gleichzeitig benutzbar. Eine Datenbank ermöglicht einen hohen Grad an Flexibilität, da nahezu jede Programmiersprache ein Interface zu Datenbanken bietet. Diese Art der Datenhaltung würde somit anderen Applikationen, die sich evtl. der Datenbasis bedienen wollen, nicht im Weg stehen. Auch ist der Zugriff über das Internet auf eine Datenbank sehr einfach zu bewerkstelligen. Die Datenhaltung bleibt durch die Absicherung von Rollen und Rechten seitens der Datenbank sehr sicher. Eine zentrale Datei könnte hingegen sehr leicht gelöscht werden, wodurch der gesamte Datenbestand verloren gehen würde (z.B. zentrale XML Datei). Ein großer Nachteil ist, dass zum Testen des Tools von fremden Personen erst eine Datenbank aufgesetzt werden muss. Dies nimmt viel Zeit in Anspruch, ist kompliziert und dürfte somit viele Personen abschrecken,

das Tool zu testen (Trial-Version anbieten). Diese Art der Datenhaltung erfüllt die Anforderungen an die Architektur.

5.1.2.2 Kriterien an die Architektur und ihre Realisierung

Mehrbenutzerfähigkeit

- Client – DB Server Architektur
- Clients laden und speichern Projekte in DB
- Vertraulichkeit wird durch die Logik in den Clients sichergestellt

Mandantenfähigkeit / Vertraulichkeit

- Wird durch Client gewährleistet
- Rollen und Rechtssystem
- Gewährleistung durch Login und Passwort

Skalierbarkeit

- Die Programmlogik des Intranettools befindet sich im Client. Steigt die Anzahl der Nutzer, so steigt lediglich die Anzahl der Transaktionen auf dem Datenbankserver
- Bei Bedarf lassen sich die Ressourcen des Datenbankserver aufrüsten, um eine höhere Performance zu gewährleisten
- Die Programmlogik des Webclients befindet sich im Server. Steigt die Anzahl der Nutzer, so muss der Server der steigenden Performanceanforderung angepasst werden oder es müssen weitere Server hinzugefügt werden

Wartbarkeit

- Da die Programmlogik sich in den Clients befindet, müssten bei Updates alle Clients getauscht werden

Erweiterbarkeit

- da das Tool eine komplette Eigenentwicklung ist, ist diese problemlos erweiterbar. Durch ein .NET Framework können sogar sprachfremde Klassen und Bibliotheken benutzt werden
- Durch Speicherung der Daten in einer Datenbank können andere Programme auf den Datenbestand zugreifen

Portierbarkeit

- Ein .NET Framework / Java ist betriebsystemunabhängig. Zudem bietet Visual Studio Möglichkeiten, Applikationen auf Handhelds und Smartphones zu portieren.

5.2 Machbarkeitsstudie

Für den Vergleich wurden Technologien gewählt, die dem Autor dieser Arbeit bekannt sind.

Java

- GUI zeitintensiv zu erstellen
- Keine vordefinierte graphische Baumstruktur zur Darstellung von FP und BFC
- Performance der Virtual Machine nicht mehr zeitgemäß
- Hohe Verbreitung, viele Codebeispiele
- Einfach zu installierende VM
- Portierbar auf mobile Geräte
- Kostenfrei

C#

- GUI sehr leicht zu bauen mittels MS Visual Studio
- Vordefinierte Baumstruktur in Model und View zur Darstellung von FP und BFC
- Hohe Performance der Virtual Machine
- Mittelmäßige Verbreitung, C# Entwickler noch relativ rar
- Codebeispiele rar, da Quellcode nicht mehr dekompileierbar
- Model Entwicklung dauert etwa gleich lang, da Java Syntax
- Einfach zu installierende VM
- Portierbar auf mobile Geräte
- Kostenfrei

Datenbank

MS SQL Server Express

- Unterstützt Obermenge von SQL-92 (ISO Standard von 1992)
- Freeware
- GUI Manager
- DB Designer
- Dem Autor ist keine freie DB mit diesem Funktionsumfang bekannt
- Wenig Einschränkungen gegenüber der Vollversion

MySQL

- keine ISO Unterstützung
- keine direkte Anbindung an MS Visual Studio
- keine GUI zum erstellen des Datenbankdesigns
- Inkonsistenzen beim Umgang mit NULL-Werten
- General Public License

Die Kombination Visual C# mit MS SQL Server Express passt am besten zu den fachlichen und technischen Anforderungen.

5.3 Fazit

Die fachliche und technische Architektur wurden gemäß den Anforderungen definiert. Eine Machbarkeitsstudie hat eine Entwicklungsumgebung und eine Datenbank gewählt, welche den Anforderungen der Architektur entsprechen.

Im folgenden Kapitel werden die Vorgehensweise der Implementierung die Art der persistenten Speicherung und einige graphische User Interfaces vorgestellt.

6 Toolentwicklung

6.1 Einleitung

In diesem Kapitel wird das zur Implementierung gewählte Vorgehensmodell vorgestellt. Es folgt das Datenbankdesign, sowie die Unterteilung in die Businessobjekte. Am Ende des Kapitels befinden sich einige Screenshots der Eingabemaske sowie des Analysebereichs. Ein Klassendiagramm des Tools ist im Anhang Kapitel 0 zu finden.

6.2 Implementierung

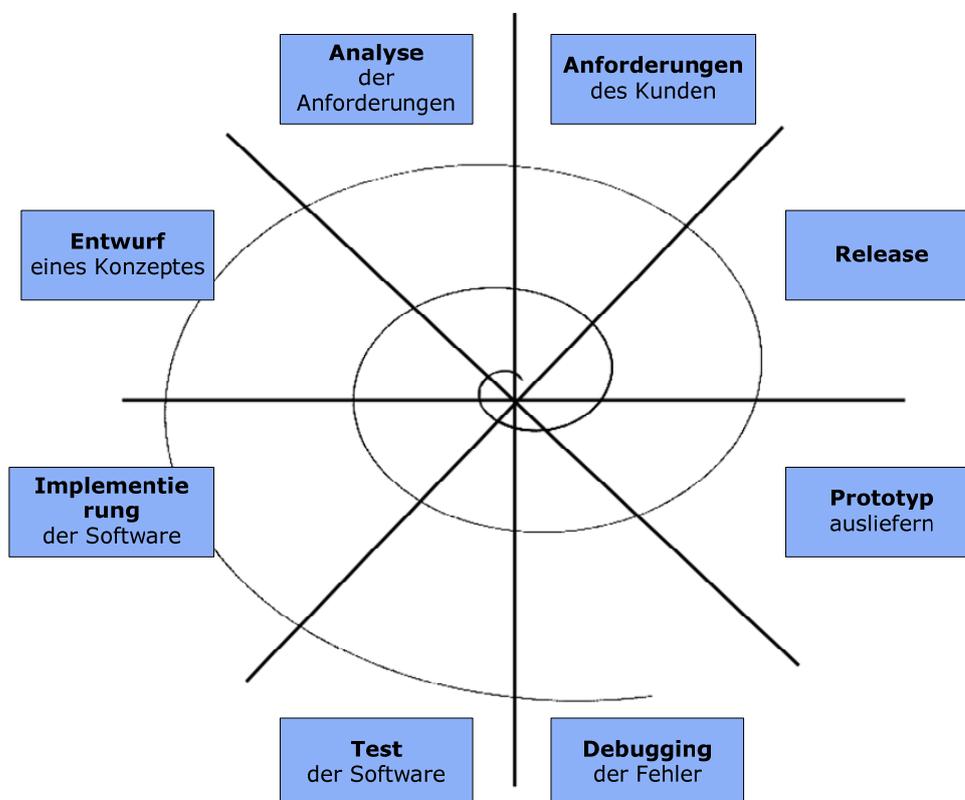


Abbildung 17: Spiralmodell nach Barry W. Boehm [BOEHM]

Aufgrund der ständig hinzukommenden Anforderungen wurde als Vorgehensweise für die Implementierung das Spiralmodell nach Barry W. Boehm gewählt (siehe Abbildung 17). Für jede zusätzliche Anforderung erweitert sich die Spirale um einen Außenkreis, die schließlich im Release endet.

6.3 Datenbank Design

6.3.1 Entity-Relationship Model

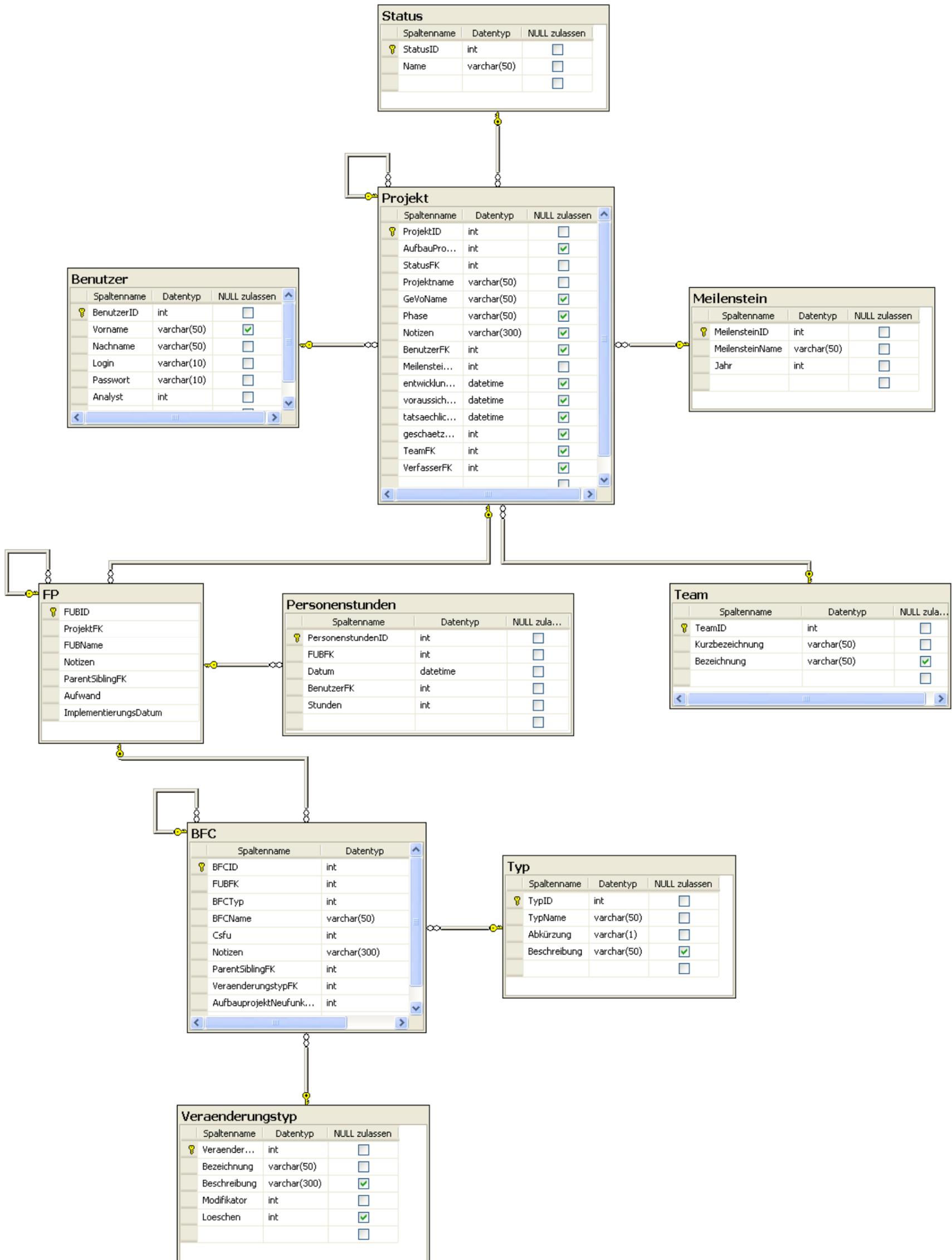


Abbildung 18: Datenbankdiagramm des Tools

7.3.2 Businessobjekte

Das Businessobjekt 'Projekt'

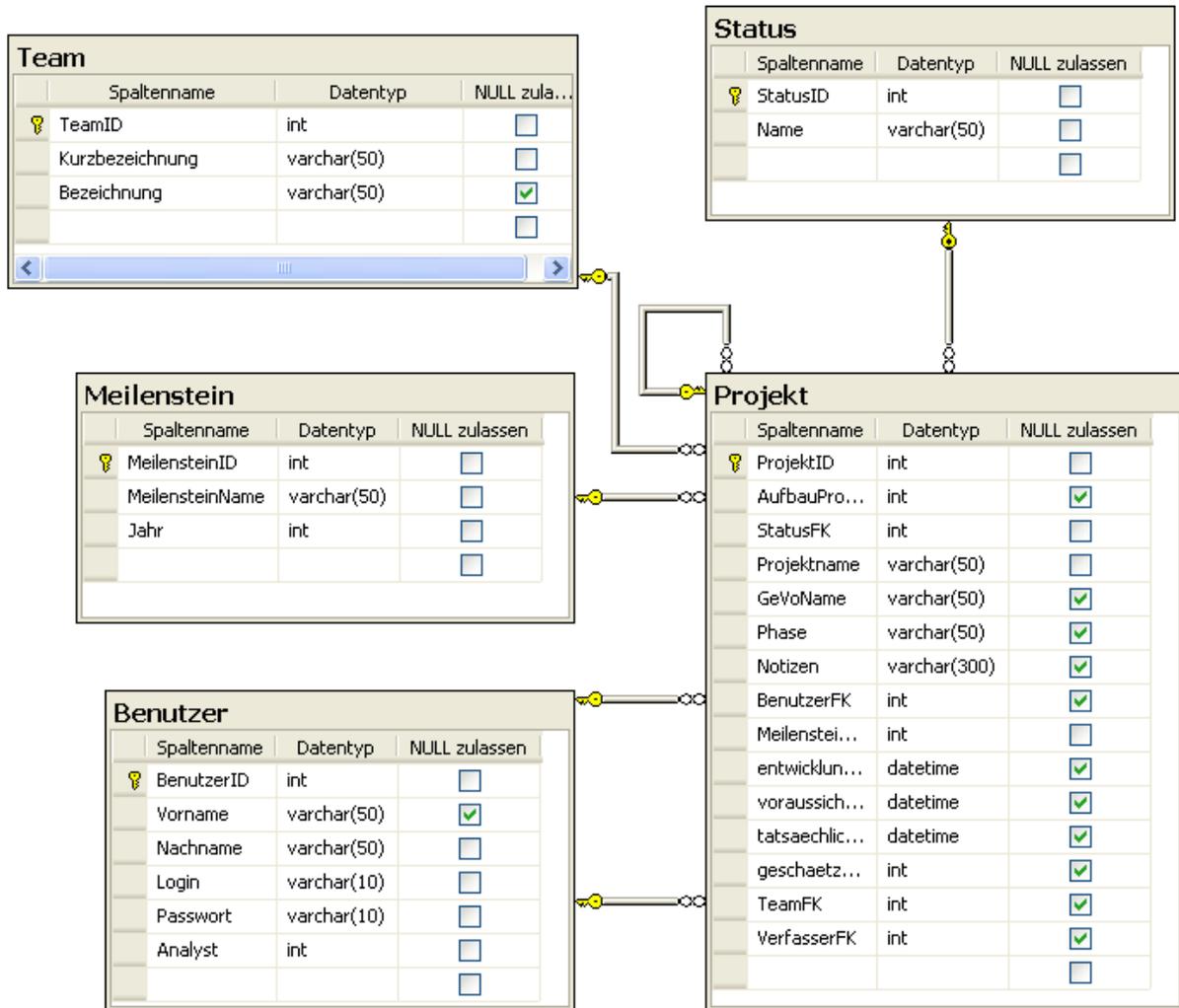


Abbildung 19: Businessobjekt 'Projekt'

In dem Businessobjekt 'Projekt' (siehe Abbildung 19) werden alle das Projekt betreffenden allgemeinen Daten gespeichert. Gemäß der Dritten Normalform wurden redundante Daten in eigenen Tabellen angelegt.

Die Tabelle 'Projekt' speichert Informationen, die das Projekt beschreiben. Referenziert die Tabelle 'Projekt' mit dem Feld 'AufbauprojektFK' auf ein anderes Projekt, so handelt es sich nicht um ein Neuprojekt, sondern um ein Aufbauprojekt, welches das referenzierte Projekt als Basisprojekt hat.

Eine Referenz auf die Tabelle 'Status' zeigt, welchen Status die Schätzung hat.

Eine Referenz 'Meilenstein' gibt an, bis zu welchem Zeitpunkt (Meilenstein) das Projekt realisiert werden soll.

Zwei Referenzen auf die Tabelle 'Benutzer' geben an, welcher Benutzer das Projekt aktuell editiert und wer das Projekt angelegt bzw. gezählt hat.

Eine Referenz auf die Tabelle 'Team' gibt an, welches Team das Projekt realisiert.

Aus statistischen Gründen muss die Tabelle 'Team' noch um Felder erweitert werden, die dokumentieren, zu welchem Zeitpunkt das Teams aus welchen Mitgliedern bestand.

Genauere Angaben zu den Tabellen und Beziehungen befinden sich im Anhang in Kapitel 0.

Das Businessobjekt 'Funktionalprozess' (FP)

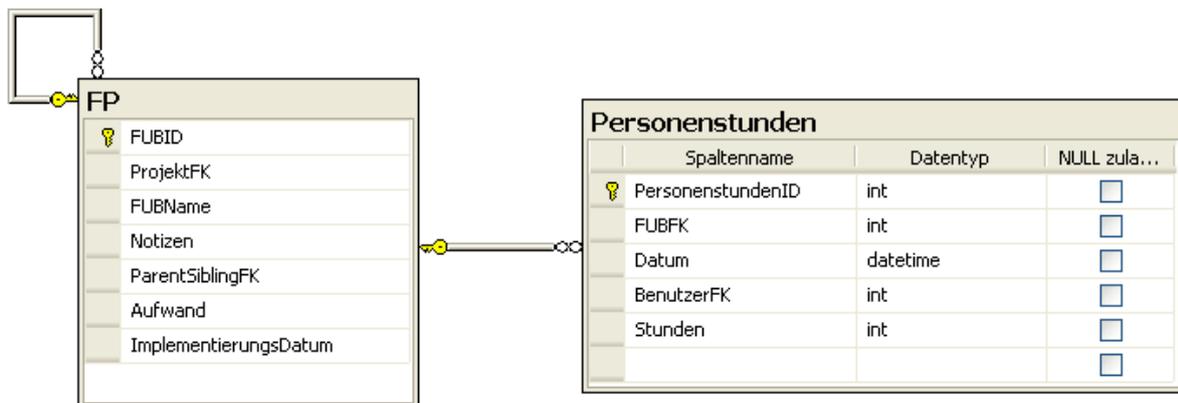


Abbildung 20: Businessobjekt Funktionalprozess (FP)

Das Businessobjekt 'Funktionalprozess' enthält Daten über das zugehörige Projekt (siehe Abbildung 19), sowie Informationen über Personenstunden die für die Implementierung des Funktionalprozesses benötigt wurden (siehe Abbildung 20). Über die Referenzierung auf die eigene Tabelle (ParentSiblingFK) wird eine Reihenfolge hergestellt. Dies dient nur der Übersichtlichkeit.

In der Tabelle 'Personenstunden' werden der Entwickler sowie die benötigten Personenstunden gespeichert.

Das Businessobjekt Basisfunktionskomponente (BFC)

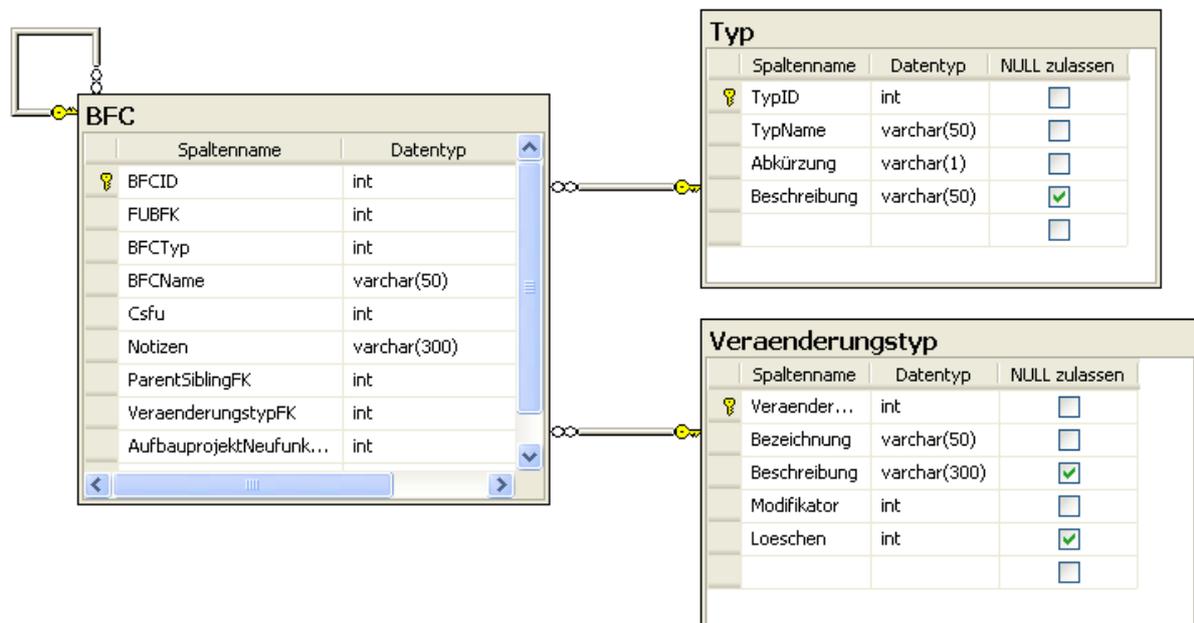


Abbildung 21: Businessobjekt Basisfunktionskomponente (BFC)

Das Businessobjekt 'Basisfunktionskomponente' (BFC) enthält Daten über den übergeordneten Funktionalprozess (siehe Abbildung 20), den Typ der BFC, die Menge der

CFSU sowie darüber, ob es sich um eine neu angelegte, veränderte oder gelöschte Basisfunktionskomponente handelt (siehe Abbildung 21).

In der Tabelle 'Typ' werden die vier Typen von Basisfunktionskomponenten gespeichert.

Die Tabelle 'Veränderungstyp' enthält Informationen darüber, ob bei Weiterentwicklungsprojekten die Basisfunktionskomponente verändert wurde.

6.4 Eingabemaske des Prototypen

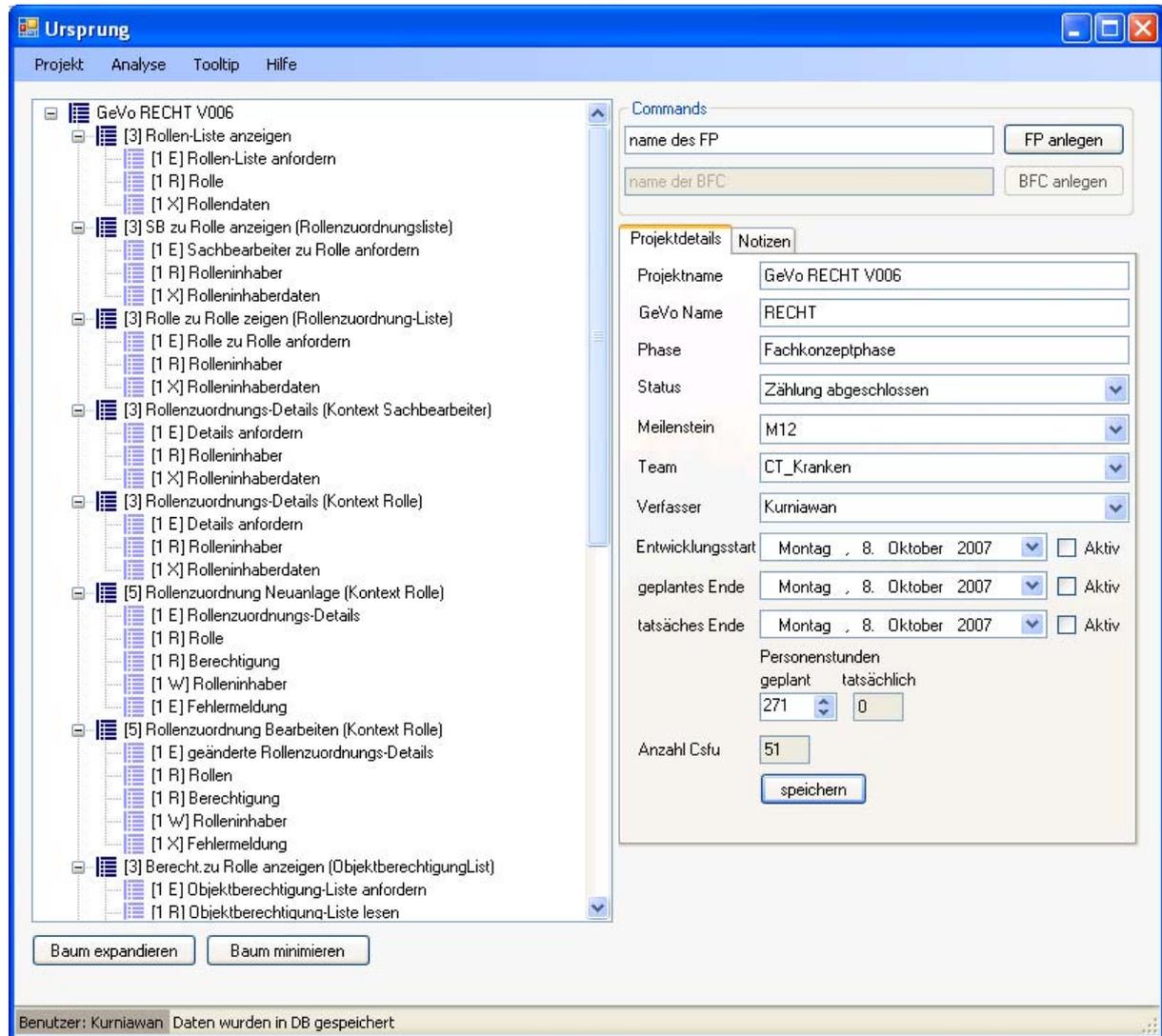


Abbildung 22: Eingabemaske zur Erfassung der FP

In Abbildung 22 ist die Eingabemaske des Zählers zu sehen. In der linken Hälfte ist ein bereits geschätztes und erfasstes Projekt in Form einer Baumstruktur zu sehen. Die linke Seite verändert sich, je nach dem welches Element im Baum angewählt wird. Momentan ist der oberste Knoten im Baum, das Projekt, selektiert. Dementsprechend sind auf der rechten Seite Eingabefelder bezüglich der Projektdaten zu sehen.

6.5 Metriken des Tools

Die Metriken des Tools stehen, gemäß der Architektur, nur den Analysten zur Verfügung. Der Analyst kann einen Teamleiter sein, oder eine außenstehende Person, die keinen direkten Bezug zu den Projekten hat.

6.5.1 Missbrauch von Metriken

Metriken, welche die Produktivität von einzelnen Entwicklern und Projektleitern bewerten, sind mit besonderer Vorsicht einzusetzen. Bei den betroffenen Personen können solche Metriken auf Misstrauen stoßen, da die Performanceanalysen zu Leistungsvergleichen missbraucht werden können.

Die Überwachung und Bewertung der Produktivität der Mitarbeiter trifft im Allgemeinen auf Ablehnung der bewerteten Personen. Werden Functionpoint Verfahren mit Produktivitätsbewertung eingeführt, so kann dies zu Akzeptanzproblemen bei den Mitarbeitern führen und somit die Einführung des Verfahrens gefährden.

Keinesfalls dürfen Projektleiter und Entwickler sich mit anderen Teams vergleichen können, da das Ansehen einzelner Personen der Teams darunter leiden könnten. Dies könnte sogar zu Minderwertigkeitskomplexen führen (Bsp.: Ein Mitarbeiter erfährt durch Kollegen, dass er die Person mit der niedrigsten Produktivität in der Firma ist).

Sollten Performanceanalysen zwischen Teams gefahren werden, so ist sicherzustellen, dass alle Parameter für den Software-Entwicklungsprozess identisch sind [BUNDSCHUH04, S.216]. Solche Analysen sollten von der Unternehmenskultur gestützt werden. Idealerweise erfolgt die Zustimmung über derlei Messungen im Rahmen der lokalen Datenschutzbestimmungen erweitert durch ein anonymes Mitarbeiter-Votum.

6.5.2 Bereits integrierte Metriken

Schätzkurve

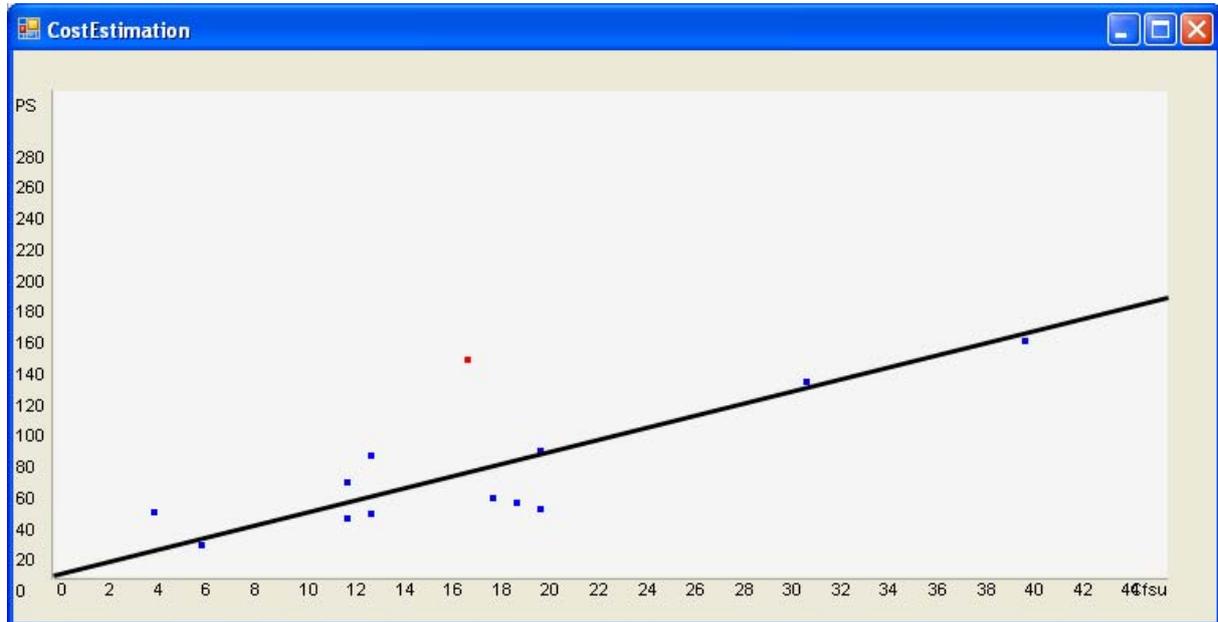


Abbildung 23: Historische Projektwerte mit Projektkurve

Legende

Y-Achse: Anzahl der der benötigten Personenstunden

X-Achse: Funktionaler Umfang der Software in der Maßeinheit Cfsu

Schwarze Linie: Lineare Regressionskurve, welche sich den dargestellten Punkten nähert

Die Schätzkurve ist die Metrik der Aufwandschätzungen. Dargestellt werden historische Projekte mit ihrem funktionalen Umfang sowie den benötigten Personenstunden (siehe Abbildung 23).

Aus der Relation von Functionpoints zu den benötigten Personenstunden lässt sich mittels statistischer Verfahren eine Schätzkurve erstellen, die auf einer Regressionskurve basiert. Die Regressionsanalyse zeigt mit dem Bestimmtheitsmaß (R^2) an, wie hoch die durchschnittliche Abweichung der Punkte zu der Kurve ist. Es gilt, für das Unternehmen den passenden Typ Regressionskurve mit der niedrigsten durchschnittlichen Abweichung zu finden. Die verschiedenen Typen der Regression sind z.B.: logarithmisch, polynomisch oder expotenziell.

Eine Extrapolation aus dem Stützintervall hinaus führt zu unzuverlässigen Ergebnissen und wird deshalb nicht angewendet. Angezeigt werden nur Regressionskurven, die innerhalb des Stützintervalls verlaufen. Eine Schätzung, die außerhalb des Stützintervalls liegt, wird somit bereits technisch unterbunden.

Die Regressionskurve ist im Tool bisher noch nicht implementiert, da der Entwicklungsumgebung ein Parser für mathematische Formeln fehlt. Die Funktion wird nach Beendigung der Abschlussarbeit nachimplementiert.

BFC Verteilung

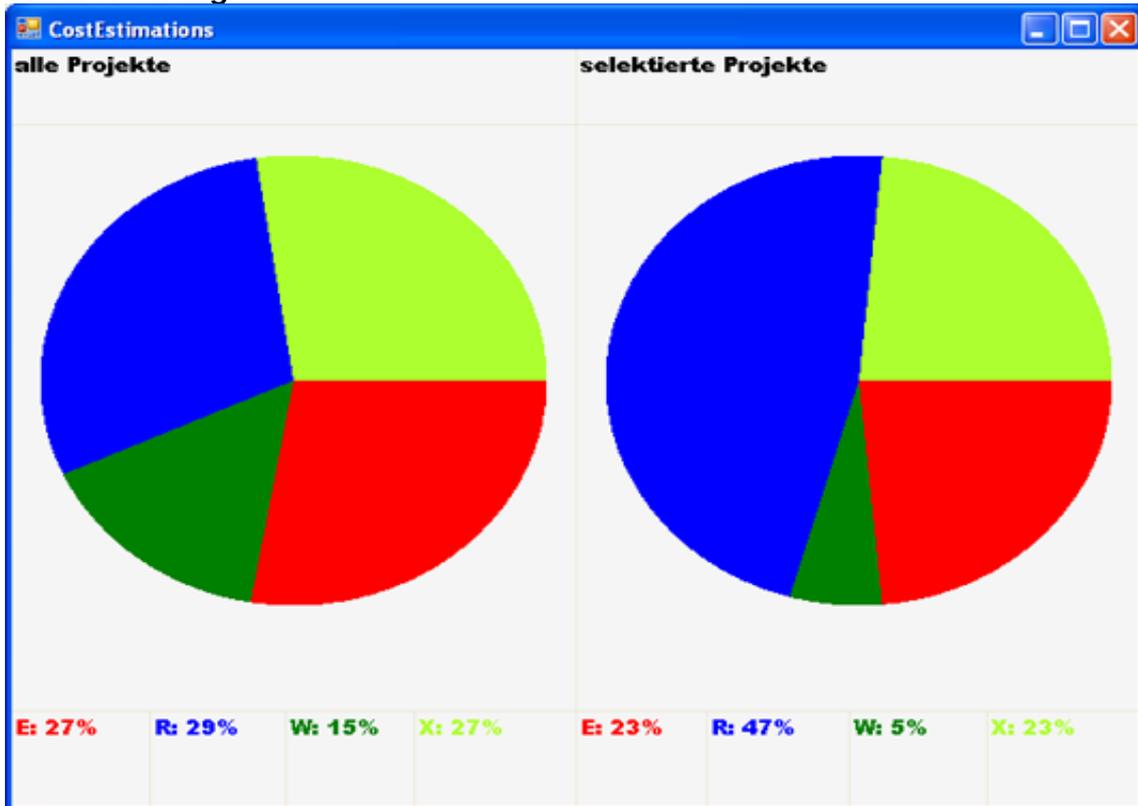


Abbildung 24: Prozentuale Verteilung der vier Basisfunktionskomponenten

Legende

Linkes Fenster: Durchschnittsverteilung der BFC aller Projekte

Rechtes Fenster: Durchschnittverteilung aller BFC der selektierten Projekte

Abkürzungen: E = Entry, R = Read, W = Write, X = Exit

Die BFC Verteilung zeigt an, zu welchem Anteil die Basisfunktionskomponenten in den Projekten vorhanden sind. Die linke Graphik zeigt die prozentuale Verteilung der Basisfunktionskomponenten über alle Projekte an. Die rechte Graphik verdeutlicht die Verteilung über die vom Benutzer selektierten Projekte (siehe Abbildung 24).

Mit der BFC Verteilung lässt sich kontrollieren, ob Projekte richtig gezählt wurden. Gibt es Abweichungen zum Durchschnitt (links), so ist dies ein Indiz für eine Unregelmäßigkeit. Der Analyst beurteilt anhand seiner Erfahrung und anhand des Projekttyps, ob die Abweichung auf eine fehlerhafte Umfangsmessung hindeutet.

Zeitpunkt der Implementierung

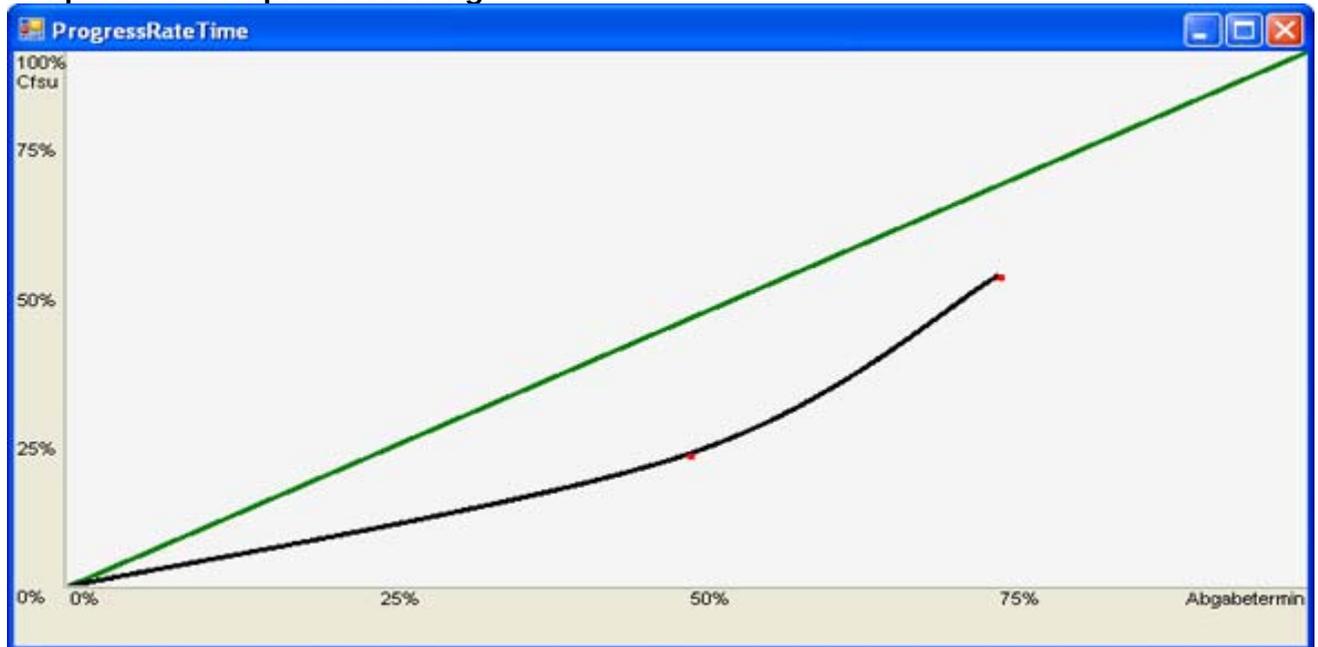


Abbildung 25: Implementierte Cfsu gemessen am Abgabetermin

Legende

Y-Achse: Prozent der bereits implementierten Cfsu

X-Achse: Zur Verfügung stehender Zeitraum

Grüne Linie: Ideallinie

Schwarze Linie: Prozentuale Menge der implementierten Cfsu gemessen am Abgabetermin

Diese Metrik zeigt prozentual an, wann in der zur Verfügung stehenden Zeit wie viel Funktionalität implementiert wurde. Hiermit lässt sich kontrollieren, ob der Projektfortschritt vom Zeitplan abweicht. Befindet sich die schwarze Kurve unterhalb der Ideallinie, so kann der Abgabetermin voraussichtlich nicht eingehalten werden. Es wurde also bisher weniger Funktionalität implementiert als vorgesehen (siehe Abbildung 25).

Produktivitätsrate

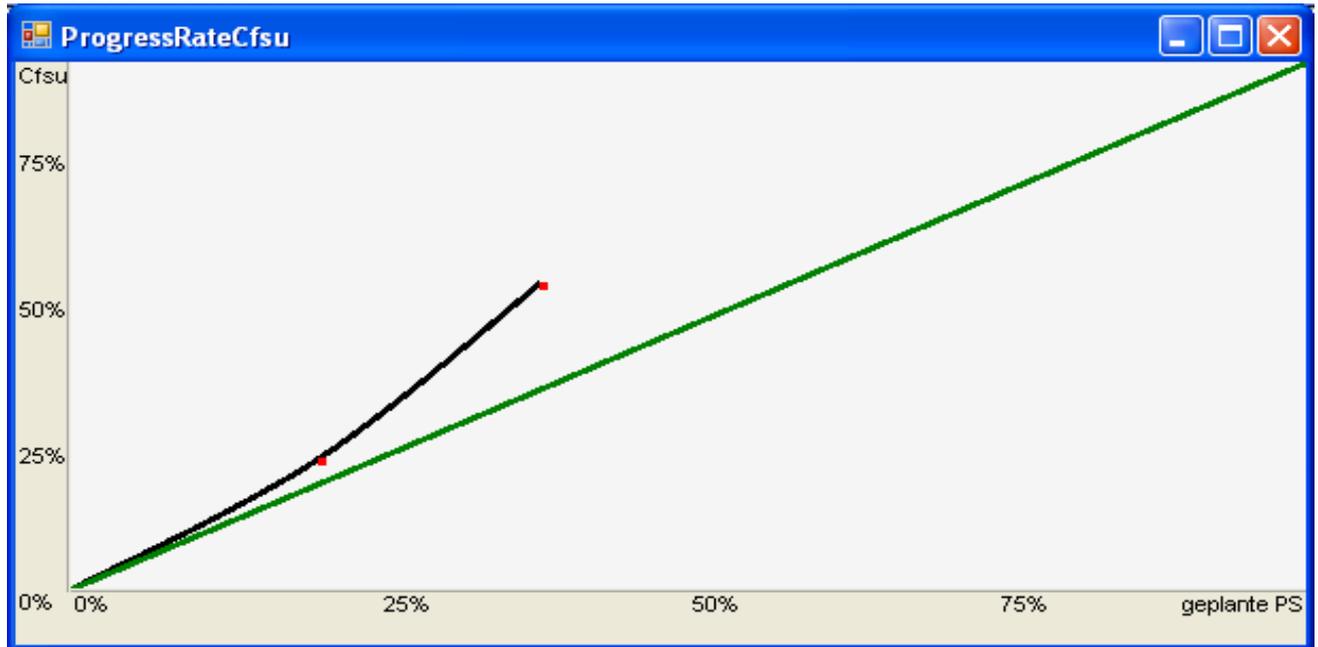


Abbildung 26: Produktivitätsrate

Legende

Y-Achse: Prozentsatz der bereits implementierten CFSU

X-Achse: Prozentsatz der bisher benötigten Personenstunden

Grüne Linie: Ideallinie

Schwarze Linie: real benötigte Cfsu pro Personenstunden

Die Metrik zeigt an, wie viel Funktionalität pro geplante Personenstunde implementiert wurde. Weicht die schwarze Kurve (Realität) nach oben hin von der grünen Ideallinie ab, so wird effektiver implementiert als geschätzt. Der Projektaufwand würde überschätzt. Weicht die Realität (schwarze Kurve) nach unten hin von der Ideallinie (grün) ab, so werden unter Umständen mehr Personenstunden für die Fertigstellung des Projektes benötigt. Der Projektaufwand wurde unterschätzt und verursacht mehr Kosten als geplant (siehe Abbildung 26).

6.5.3 Zukünftige Metriken

Es folgen neue Konzepte für weitere Metriken, die den Analysten und Projektcontroller unterstützen könnten. Generell gilt für alle Metriken: Der Benutzer kann zwischen verschiedenen Teams, Projekten, Teilsystemen und Zeiträumen wählen.

Anzahl der Fehler pro Functionpoint

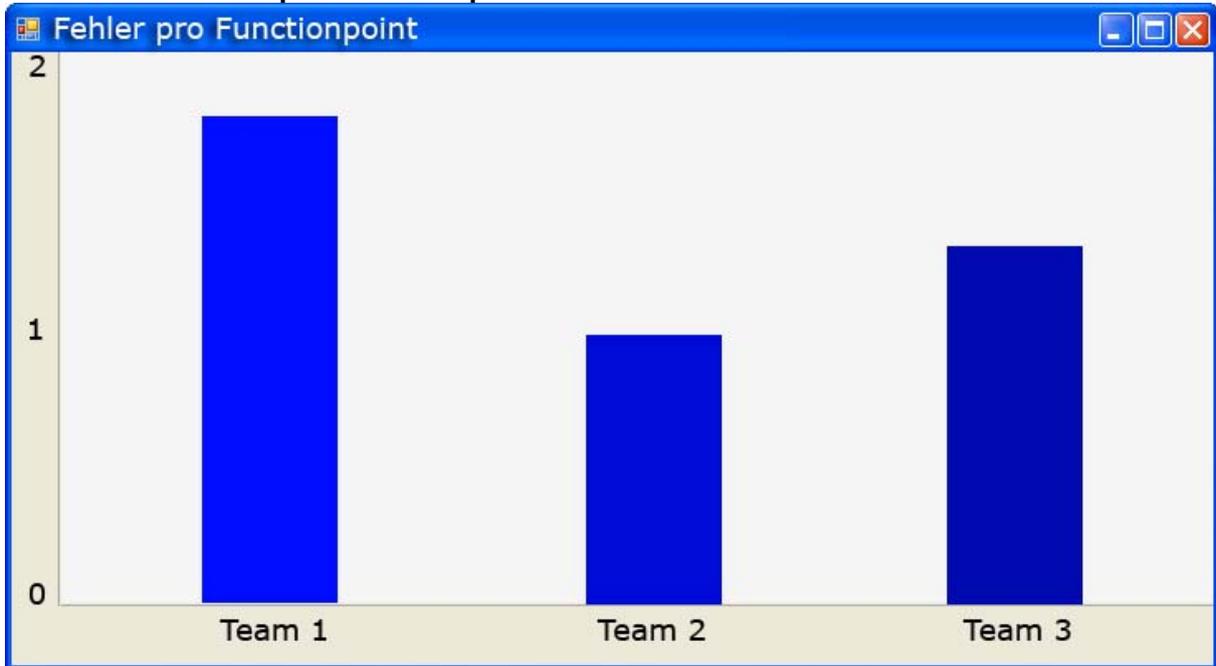


Abbildung 27: Anzahl der Fehler pro Functionpoint

Der Analyst kann zwischen verschiedenen Teams oder Projekten wählen, die er im Graphen gegenüber stellen möchte. Anhand der Graphen kann der Analyst sehen, welche Projekte besonders viele Fehler pro Functionpoint haben. Dementsprechend kann mit der Ursachenanalyse begonnen werden, um die Quelle für die erhöhte Fehlerquote zu finden.

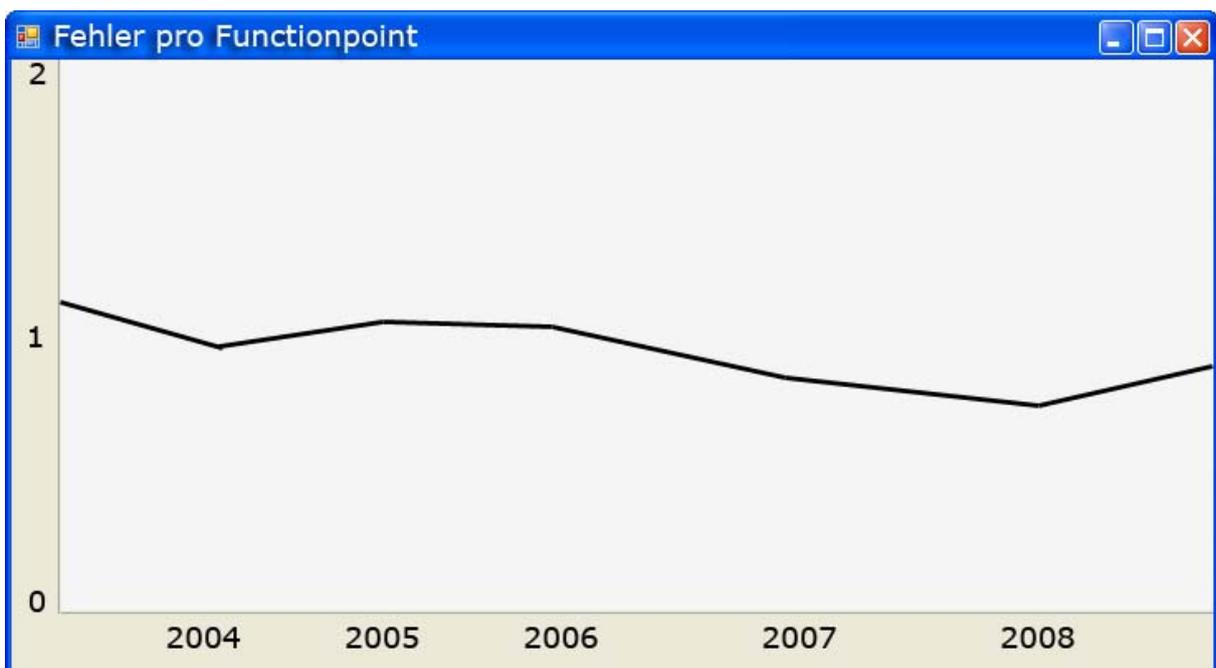


Abbildung 28: Fehler pro Functionpoint

Des Weiteren sollte eine Kurve über den kompletten Zeitraum der Erfassung zeigen, ob sich die Qualität, sprich die Fehler pro FP, über die Jahre verbessert hat.

Übersicht der Arbeitsstundenverteilung eines Mitarbeiters

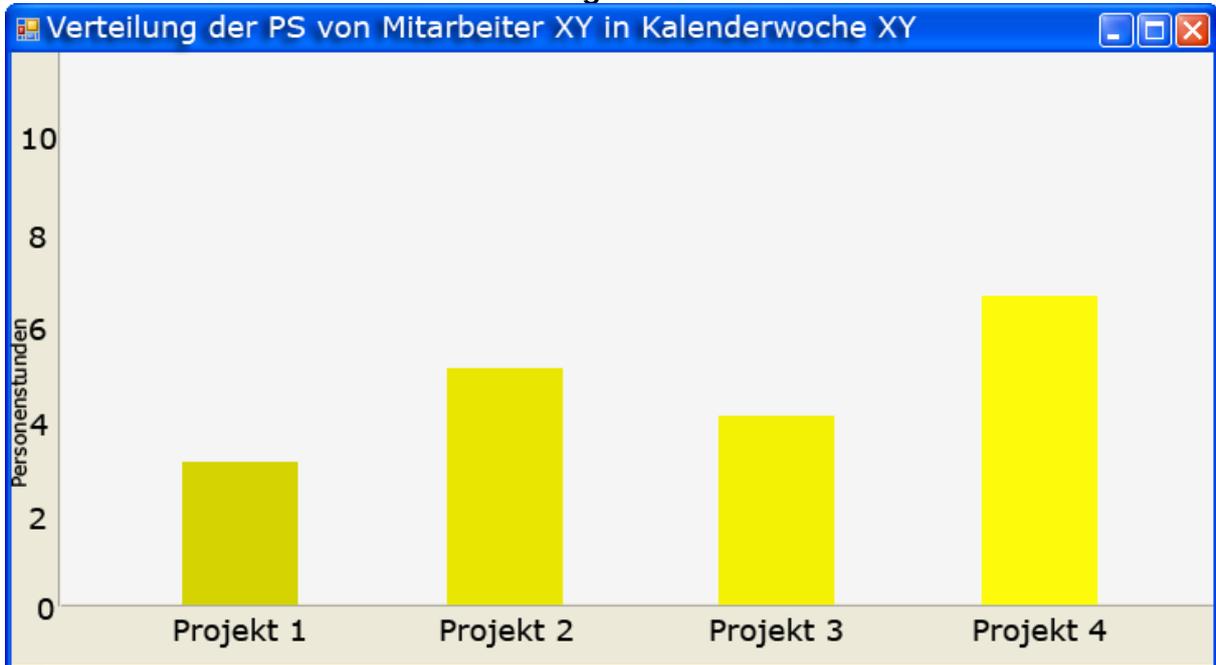


Abbildung 29: Arbeitsstundenverteilung eines Mitarbeiters

Der Projektcontroller kann einsehen, an welchen Projekten einzelne Entwickler über einen bestimmten Zeitraum gearbeitet haben. Der Controller kann überwachen, ob ein Entwickler evtl. zu wenig Personenstunden in ein Projekt mit hoher Priorität investiert.

Verteilung der Personenstunden eines Projektes über die Entwickler

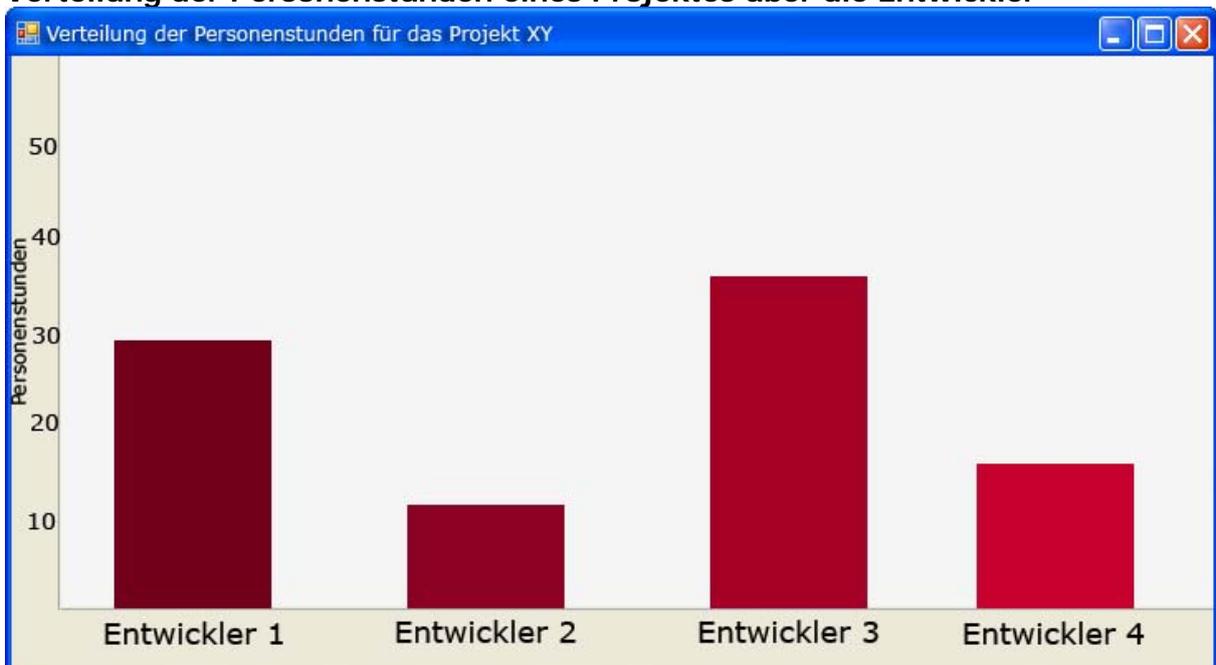


Abbildung 30: Verteilung der Personenstunden eines Projektes

Der Projektcontroller kann hier einsehen, ob die einzelnen Entwickler gleichermaßen an einem Projekt mitgewirkt haben.

Vergleich der Effizienz zwischen den Teams

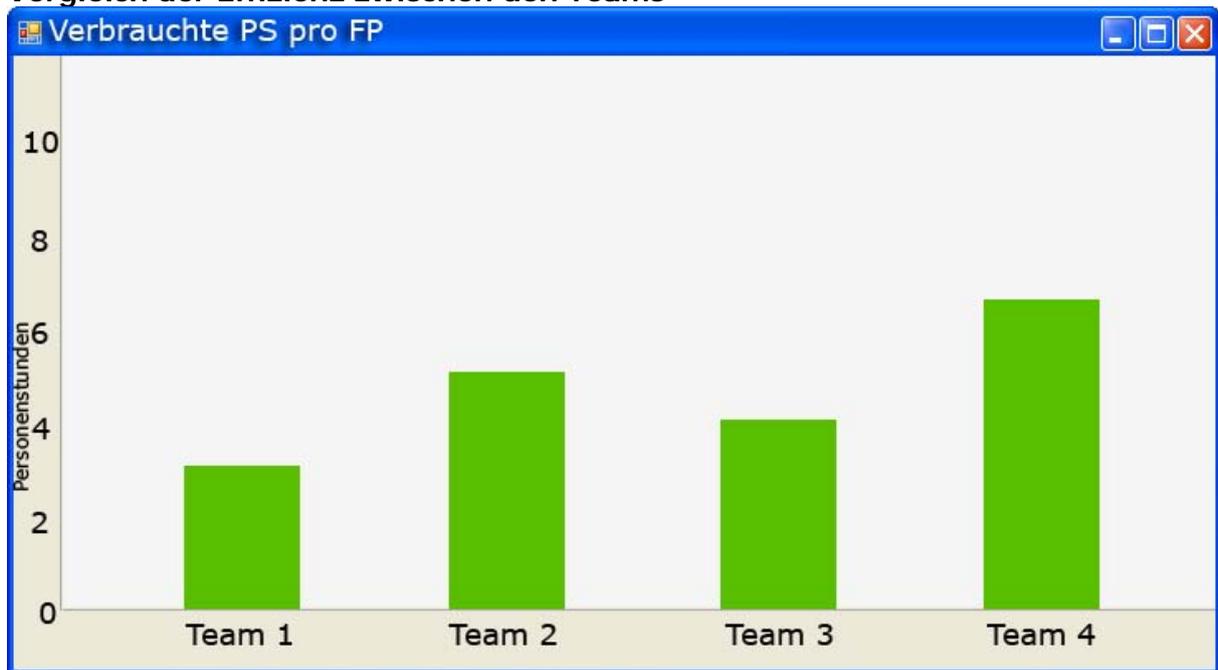


Abbildung 31: Effizienz der Implementierung

Der Analyst kann vergleichen, wie viele Personenstunden die einzelnen Teams im Schnitt für die Implementierung eines Functionpoint benötigen. Es kann so die Effizienz der einzelnen Teams bewertet werden. Kommt es zu großen Abweichungen zwischen den Teams, so muss nach den Ursachen gesucht und diese müssen beseitigt werden.

Requirement Creep

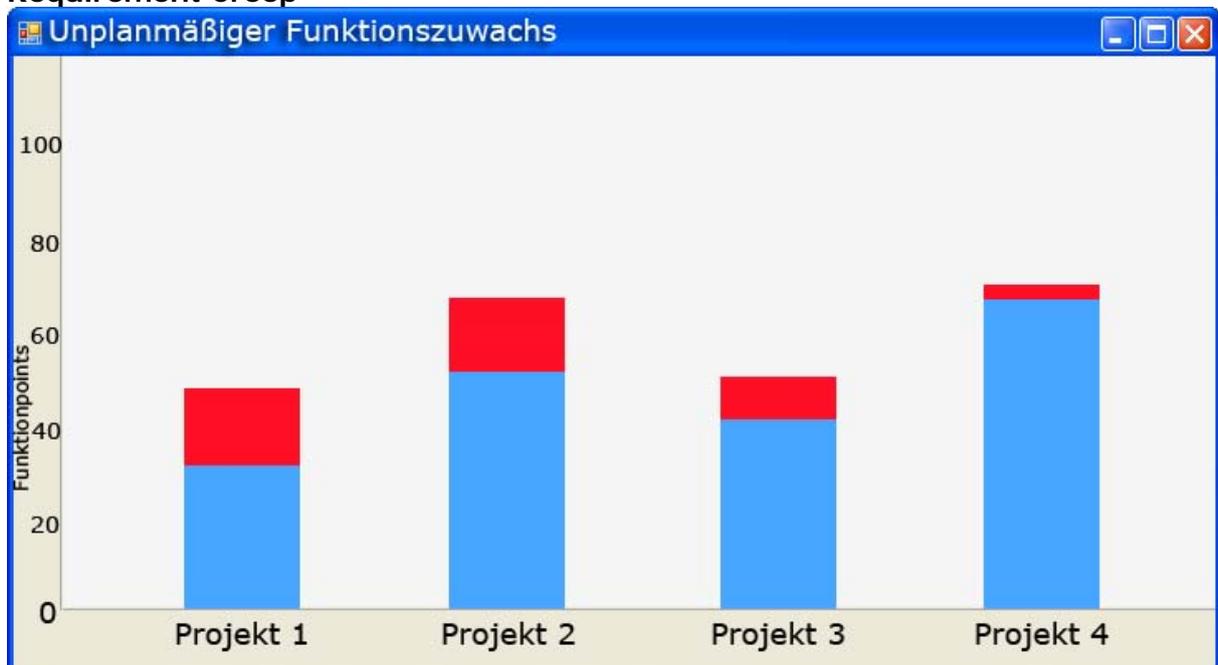


Abbildung 32: Unplanmäßiger Funktionszuwachs

Werden nach Absegnung eines Pflichtenheftes oder Fachkonzeptes noch zusätzlich Funktionalitäten vom Kunden gefordert, so handelt es sich um unplanmäßigen Funktionszuwachs (engl. Requirement Creep). In dieser Metrik kann der Controller einsehen, wie viel unplanmäßigen Funktionszuwachs die einzelnen Projekte haben. Bei

den blauen Balken handelt es sich um den zugesagten Funktionsumfang. Die roten Balken in Abbildung 32 stellen den unplanmäßigen Funktionszuwachs der einzelnen Projekte dar.

Umsatz pro FP

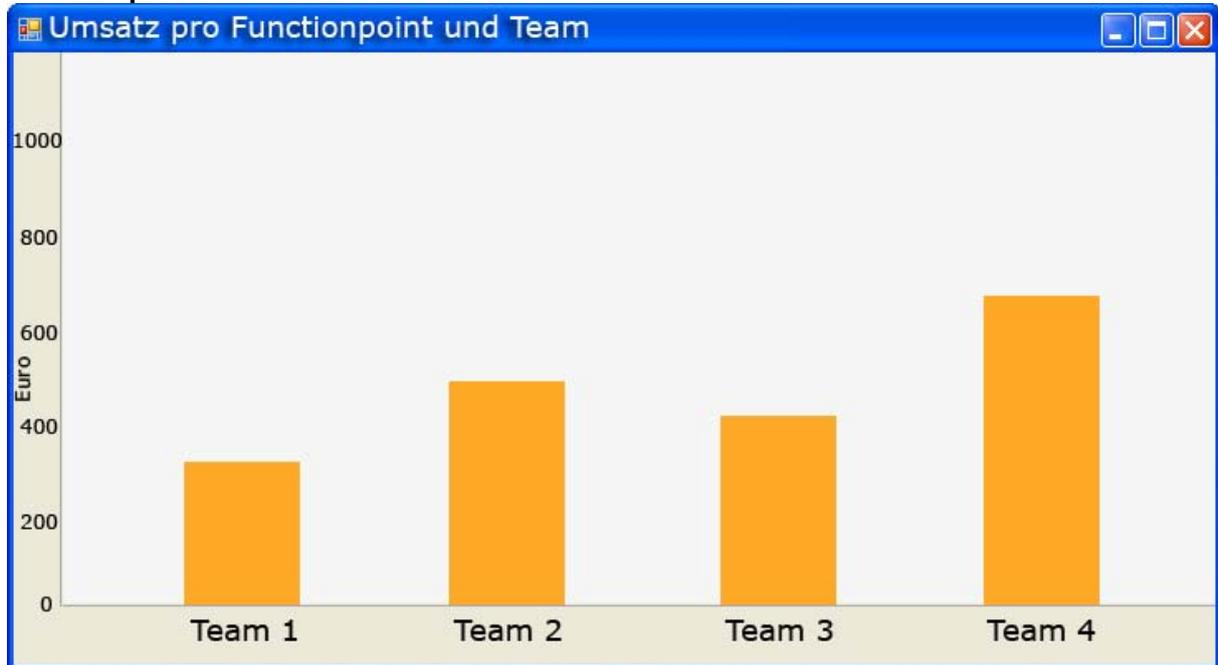


Abbildung 33: Umsatz pro FP

Aus dieser Metrik lässt sich entnehmen, wie viel Umsatz die einzelnen Teams pro Functionpoint generieren. Diese Metrik kann die Ergebnisse anderer Metriken abschwächen: Steht ein Team unter Verdacht, zu wenig Functionpoint zu implementieren, so kann es sein, dass dieses Team gleichermaßen Umsatz wie andere Teams generiert, weil die Quote Umsatz pro Functionpoint höher ist.

Kosten pro FP

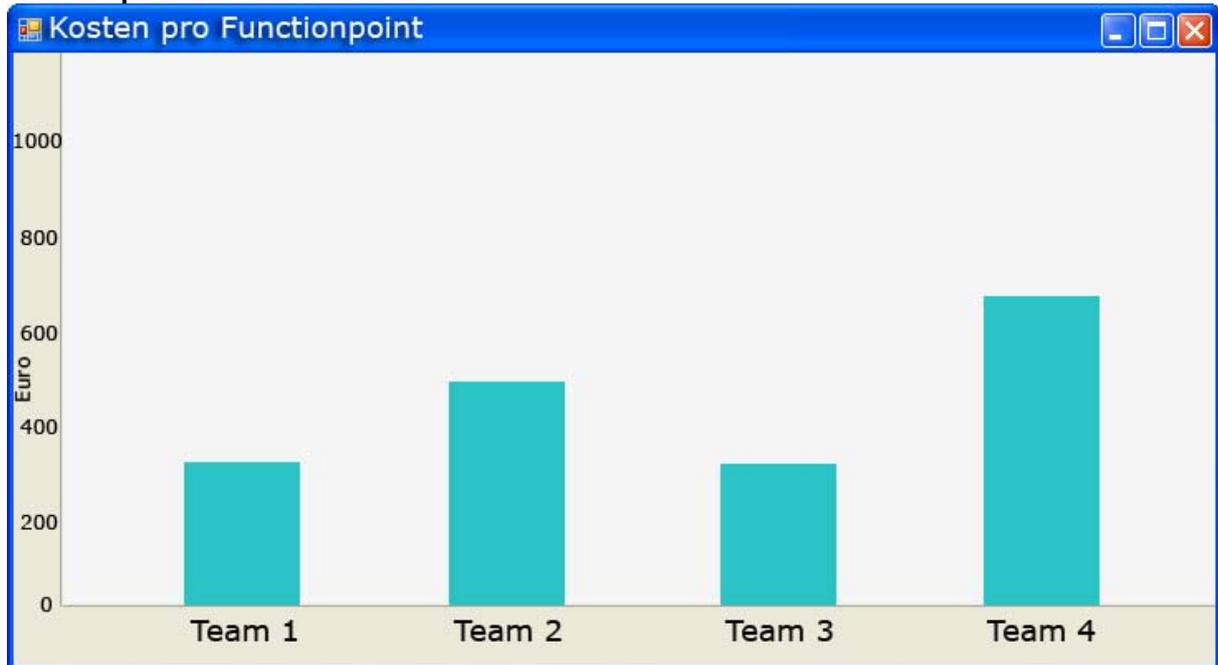


Abbildung 34: Kosten pro Functionpoint

Auch diese Metrik kann andere Metriken abschwächen. Hat ein Team einen schlechten Schnitt in der PS/FP Metrik, so kann es z.B. sein, dass das Team schlechter bezahlt wird als ein Team mit besserem PS/FP Schnitt. Diese Metrik würde Aufschluss darüber geben, ob das Team deshalb auch höhere Kosten verursacht. Es könnte also sein, dass die Kosten pro FP gleich sind, obwohl mehr Personenstunden für einen Funktionalprozess benötigt werden. Team 1 und Team 3 können unterschiedlich viele Personenstunden pro Funktionalprozess benötigen, für das Unternehmen bleiben die Kosten pro FP in diesem Beispiel aber gleich hoch.

6.6 Fazit

Das Tool wurde nach den Vorgaben der Architektur so weit entwickelt, dass die Entwickler ihre Projekte erfassen und der Analyst erste Schätzprognosen erstellen kann.

Der Prototyp wurde getestet und läuft ohne erkennbare Fehler.

7 Die Einführung im Unternehmen

7.1 Eignung der Fachkonzepte für eine Zählung

Von der Präzision der Schätzung hängt die Qualität der Fachkonzepte ab. Ist das Fachkonzept nicht ausreichend beschrieben, so ist keine fundierte Schätzung möglich, da der funktionale Umfang nicht im Fachkonzept komplett beschrieben ist. Was nicht im Fachkonzept steht, kann auch nicht geschätzt werden.

Zur Eignungsprüfung der Fachkonzepte wurde eine Umfrage unter den Leitern der Testteams durchgeführt, welche zeigen sollte, ob Schätzungen mit der aktuellen Qualität der Fachkonzepte grundsätzlich möglich wären.

Geprüft werden soll, ob der komplette funktionale Umfang von Neu- und Weiterentwicklungen beschrieben ist.

Der Autor hat die Erfahrung gemacht, dass Aktivitätsdiagramme ein guter Indikator für die Schätzbarkeit von Projekten sind. Ist es möglich, ein Aktivitätsdiagramm für ein Projekt oder Fachkonzept zu erstellen, so ist es nach eigener Erfahrung auch schätzbar. Dies ist nicht durch Fachliteratur belegt und bedarf einer Feldstudie.

Die Teams wurden befragt, ob für Iterative Projekte (Basisprojekte) ein Aktivitätsdiagramm erstellt werden kann, und wie viele Personentage dies in Anspruch nehmen würde.

Diese Einschätzung ist notwendig, da für Schätzung von Iterativen Projekten erst das Basisprojekt erfasst werden muss.

Der Umfragebogen, zusammengefasst mit Ergebnissen und einer Auswertung:

Umfrage an 3 Teamleiter der intersoft ag über die Qualität von Fachkonzepten

Fragestellung

1. Wie viel Prozent der Neuentwicklungen beschreiben den kompletten funktionalen Umfang der Neuentwicklung?
2. Wie viel Prozent der Weiterentwicklungen beschreiben den kompletten funktionalen Umfang der Weiterentwicklung?
3. Wie viel Aufwand entsteht bei der Erstellung eines Aktivitätsdiagramms für das Basisprojekt?
4. Wie viel Aufwand entsteht bei der Erstellung eines Aktivitätsdiagramms für den Teilbereich der Änderung (gleichgültig ob sich der Ablauf ändert oder besteht bleibt)?

	Frage 1	Frage 2	Frage 3	Frage 4
Team 1	100%	95%	0,5 PT	meist vorhanden
Team 2	80%	90%	ca. 10 PT	bis zu 10 PT
Team 3	100%	Frage anders interpretiert	abhängig von FUBen	Bis zu mehreren Wochen

Frage 2, Team 3: Ich würde schätzen zwischen 5% und 10%. Es wird eben nur die Teile beschrieben, die sich ändern. Der Rest wird fachlich nicht erwähnt.

Auswertung

Neuentwicklungen sind laut Aussage der Teams gut schätzbar. Nahezu der komplette funktionale Umfang der Fachkonzepte ist beschrieben.

Weiterentwicklungen benötigen das mit Functionpoints erfasste Basisprojekt. Ein Aktivitätsdiagramm ist ein guter Indikator für die Schätzbarkeit von Projekten. Für das Erstellen von Aktivitätsdiagrammen werden teilweise Wochen benötigt. Eine Erfassung der Functionpoints der Basisprojekte würde den Aufwand grob verdreifachen. Der Aufwand für die Erfassung der eigentlichen Änderungen wäre sehr gering.

Zusammenfassung

Neuentwicklungen der intersoft AG sind problemlos schätzbar, stellen aber nur 20% der Projekte dar (Quelle MZ).

Weiterentwicklungen sind ohne hohen Zählaufwand der Teamleiter praktisch nicht zu schätzen. Diese stellen aber 80% der Projekte bei der intersoft AG dar. Es könnte versucht werden, das Basisprojekt nur an den Teilbereichen der Veränderungen zu erfassen bzw. nur die Veränderungen. Diese Zählungen wären allerdings zusammenhangslos und würden spätere Schätzungen bei Weiterentwicklungen des Projektes nicht unterstützen. Der Mehraufwand dieser Methode, sollte sie überhaupt anwendbar sein, würde früher oder später den Aufwand einer Basisprojekterfassung übersteigen.

Glossar:

- Neuentwicklungen: Projekte, bei denen Funktionalität zu einem Projekt hinzugefügt wird bzw. ein komplett neues Projekt
- Weiterentwicklungen: Projekte, bei denen die Funktionalität eines alten Projektes verändert wird
- Funktional komplett beschrieben: Der Entwickler kann mit Hilfe des Konzeptes die Implementierung vornehmen, ohne offene Fragen zu haben
- Basisprojekt: Projekt, dessen Funktionalität von einem Weiterentwicklungsprojekt verändert wird

7.2 Vorgehensweise zur Einführung von Aufwandschätzungen

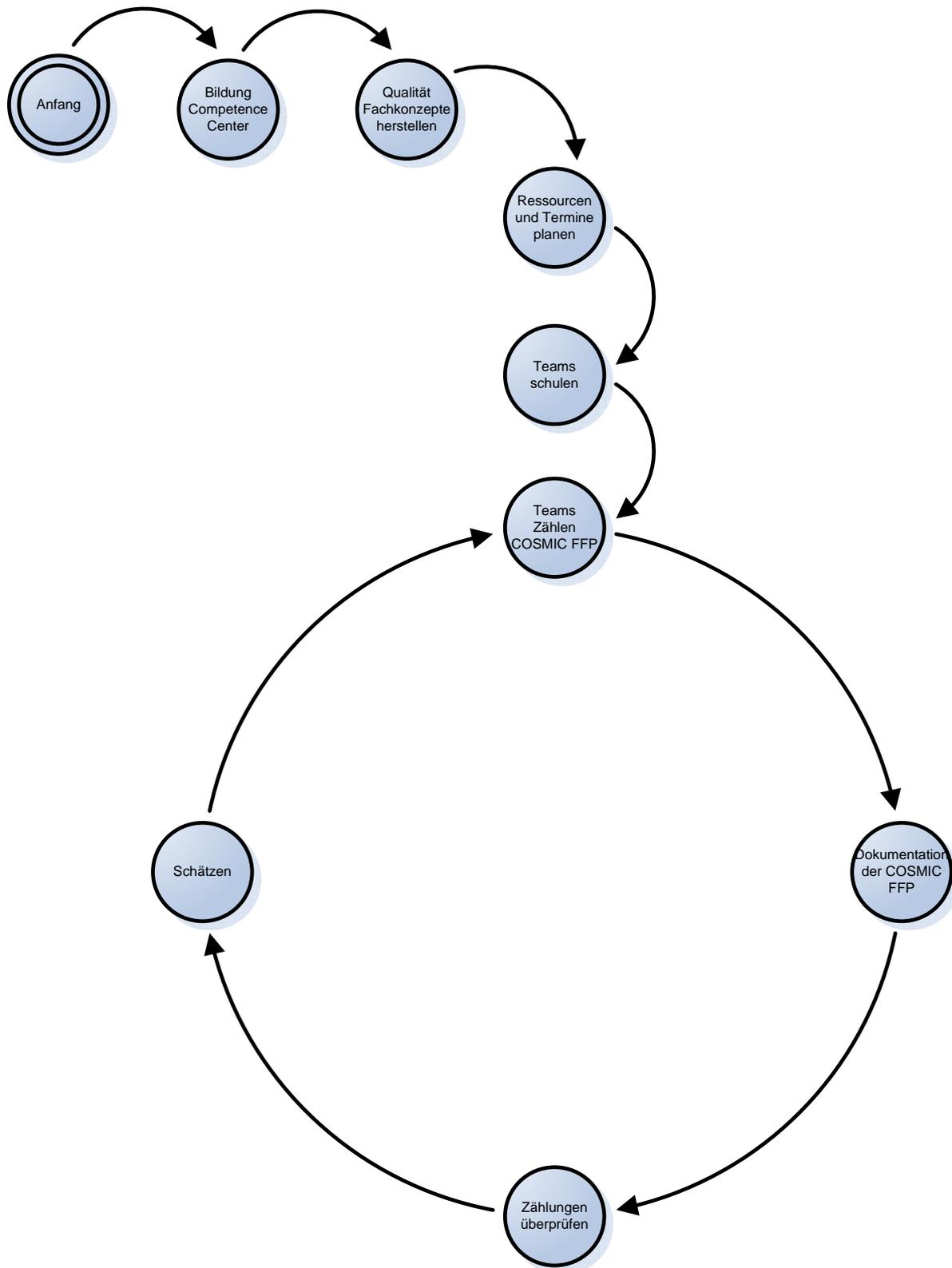


Abbildung 35: Zyklischer Ablauf der Gewinnung von historischen Daten

7.2.1 Bildung eines Competence Centers

Plan:

Das Competence Center ist die zentrale Anlaufstelle für alle Belange zu Aufwandschätzungen der Firma. Im Competence Center werden alle Informationen aus den Abteilungen zentralisiert. Dies führt zu einer einheitlichen Datenbasis für die ganze Firma und erleichtert das Dokumentieren und Analysieren von Schätzungen. Das Competence Center kann regelmäßig firmenweite Publikationen vornehmen, um den einzelnen Abteilungen ein Feedback ihrer Schätzungen zu geben und um Diskussionen anzuregen. Unter anderem wird durch diesen Erfahrungsaustausch die Akzeptanz gefördert (siehe Abbildung 35).

Der Competence Center Koordinator steht den Function-Point-Zählern der Teams bei offenen Fragen jederzeit zur Verfügung.

Zudem bleiben die Schätzungen objektiv, da sie vom Competence Center durchgeführt werden und nicht von den Teams selbst. Absichtliche Fehlschätzungen, welche aus buchungstechnischen Gründen von den Teams verursacht werden könnten, werden somit vermieden.

Des Weiteren werden die Daten zentral gesammelt und sind nicht über die Teams verteilt. Inkonsistenzen durch dezentrale Datenhaltung in einer Erfahrungsdatenbank werden somit minimiert.

Aufwandschätzer sind eine begrenzte Ressource; es kann nicht in jedem Team einen Aufwandschätzer geben.

Umsetzung:

Es wurde ein Competence Center in der betreuenden Firma gebildet. Leiter des Competence Centers ist der Autor dieser Arbeit. Die zentrale Datenbasis aller Zählungen, in Form einer Datenbank befindet sich im Competence Center.

Nach abgeschlossenen Schätzungen gibt es Feedback in Form eines Newsletters an die Teams. Die Zähler wenden sich bei Fragen an das Competence Center.

7.2.2 Bisherige Fachkonzepte auf Schätzbarkeit analysieren

Siehe Kapitel 7.1 Eignung der Fachkonzepte für eine Zählung

7.2.3 Ressourcen und Termine planen

Plan:

Die benötigten Ressourcen müssen geschätzt werden. Da es bisher noch keine Schätzungen in der Firma gegeben hat, fallen Vergleiche schwer. Eine Faustformel besagt, dass die Aufwandschätzung 1% des gesamten Projektaufwandes verursacht. Da der Projektaufwand aber noch geschätzt werden muss, kann diese Formel noch keine Anwendung finden.

Des Weiteren müssen Termine für die Schulung gefunden werden. Die Teams sollen möglichst zusammen an einem Termin geschult werden.

Umsetzung:

Da eine Formel zum Schätzen des Zeitaufwands für Umfangsmessungen fehlt, wurde subjektiv vom Competence Center geschätzt. Grundlage der Schätzung war die eigene Erfahrung mit Umfangsmessungen. Diese wurde verdoppelt, da es die erste Zählung für die Teams wäre, und somit natürlich mehr Zeit in Anspruch nehmen würde.

Insgesamt war die Schätzung aber Spekulation, weil weder bekannt war, wie viele Projekte zu schätzen waren, noch wie groß diese sein würden.

Veranschlagt wurden 1,5 Personentage pro Team.

Nach Schätzung der benötigten Ressourcen mussten diese beim Vorstand beantragt werden. Die Zusage des Vorstands kam aufgrund der Urlaubszeit sehr spät. Zudem standen zu der Zeit alle Teams kurz vor Fertigstellung eines Meilensteins. Danach begann die Urlaubszeit, so dass während dieser Zeit auch keine Schulung möglich war. Insgesamt war es innerhalb eines Zeitraums von zwei Monaten nicht möglich gewesen, einen Termin für eine Schulung zu vereinbaren.

Bei späteren Zählungen stellte sich heraus, dass den Teamleitern für die Zählungen zeitlicher Freiraum geschaffen werden muss. Wird den Teamleitern kein solcher Freiraum gegeben, erhöht die Zählung durch den zusätzlichen Arbeitsaufwand den Druck auf die Teams. Somit sinkt auch die Akzeptanz.

7.2.4 Teams zu den geplanten Terminen schulen

Plan:

Teams schulen und möglichst zeitnah nach der Schulung erste Schätzungen vornehmen, damit das Geschulte präsent bleibt.

Umsetzung:

Die Teams wurden 1,5 Stunden lang geschult. Geschult wurden nur die Zähler, keine Analysten. Die Schulung kam bei den Testteams gut an. Unklarheiten konnten noch während der Schulung geklärt werden. Noch in der gleichen Woche fanden erste Schätzungen statt.

7.2.5 Teams bei den ersten Messungen unterstützen

Plan:

Die ersten Schätzungen der Teams sollen zu verschiedenen Terminen durchgeführt werden. Bei ersten Schätzungen kommen mit hoher Wahrscheinlichkeit viele Fragen auf. Um jedem Team bei Unklarheiten zur Verfügung zu stehen, ist ein zeitversetztes Schätzen von Vorteil.

Umsetzung:

Die Schätzungen der einzelnen Teams wurden so gelegt, dass sich keine Überschneidungen ergaben. Wöchentlich wurden Schätzungen von historischen Projekten vorgenommen, um möglichst schnell einen Datenbestand für Prognosen herzustellen. Die Schätzungen brachten auch Unklarheiten mit sich, die durch das scheinbar umfangreich beschriebene Schätzhandbuch nicht beantwortet werden konnten. Hier gilt es, feste Regeln aus diesen unklaren Situationen zu ermitteln.

7.2.6 Dokumentation der Zählung

Vorüberlegung:

Die Teams dokumentieren ihre Ergebnisse in dem Tool. Ohne Aufbereitung der Daten können sofort Analysen gefahren werden.

Umsetzung:

Die Teams haben die Ergebnisse ihrer Zählungen in dem Tool dokumentiert. Es konnten sofort erste Probeschätzungen erfolgen. Alle Teams kamen mit der Bedienung des Tools gut zurecht. Kleine Unklarheiten konnten sofort geklärt werden.

7.2.7 Feedback der Schulungsteilnehmer

Vorüberlegung:

Die Schulungsteilnehmer geben dem Competence Center Feedback über Probleme beim Schätzen und Verständnisprobleme bei der Schulung.

Umsetzung:

Die Schulungsteilnehmer haben den Inhalt der Schulung gut aufgenommen und verstanden. Nach der Schulung wurden in einer Diskussion verschiedene Punkte noch einmal vertieft. Erwartungsgemäß kamen während der ersten Zählungen weitere Fragen auf, die durch das Competence Center beantwortet werden konnten.

7.2.8 Prüfen der Zählungen

Vorüberlegung:

Die Zählungen werden auf Fehler vom Competence Center geprüft. Mit der Präzision der Zählungen steht und fällt die Genauigkeit der Schätzungen.

Umsetzung:

Die Zählungen der Teams wurden vom Competence Center geprüft. Erwartungsgemäß gab es zu jeder Schätzung verschiedene Korrekturen. Oftmals wurden Funktionalprozesse nicht als solche erkannt, da aus Sicht des Schätzers keine Datenbewegungen vorlagen. Unter anderem wurden Funktionalprozesse oftmals doppelt angelegt, obwohl sie die gleiche Funktionalität haben. Diese Unklarheiten konnten geklärt werden.

Es stellte sich auch heraus, dass Anforderungen keiner genauen Regel des COSMIC-FFP Measurement Manual unterlagen. Für diese Definitionslücken müssen eigene Regeln erstellt werden, damit alle Teams gleich schwer gewichten.

7.2.9 Schätzprognosen erstellen

Vorüberlegung:

Anhand weniger historischer Daten, die bereits in das Tool eingepflegt wurden, können erste vorsichtige Schätzprognosen vorgenommen werden. Für zuverlässige Schätzungen bedarf es weitaus mehr historischer Daten.

Umsetzung:

Vorsichtige Schätzprognosen wurden anhand von den bisherigen wenigen historischen Daten erstellt. Ein ausreichend großer Datenbestand für fundierte Schätzungen wird wahrscheinlich in einem Jahr vorhanden sein.

7.2.10 Schätzabweichung messen und Messfehler

Vorüberlegung:

Schätzabweichung der erstellten Prognosen mittels des realen Aufwands berechnen. Vergleiche zu bisherigen subjektiven Schätzungen ziehen.

Bei allzu starken Abweichungen evtl. nach Messfehlern suchen.

Umsetzung:

Die Schätzabweichungen waren bisher zu groß für zuverlässige Schätzungen. Es stellte sich heraus, dass es unter den Teams aufgrund der Projektart zu erheblichen Unterschieden kam. Dies muss im Laufe der Zeit und mit wachsenden historischen Daten evaluiert werden.

7.2.11 Ergebnisse der Schätzung an die Testteams / ganze Firma publizieren

Vorüberlegung:

Projektleiter sowie Schätzer sollten umgehend über den Erfolg der Messungen informiert werden. Dies mindert bei guten Ergebnissen die Akzeptanzprobleme.

Umsetzung:

Den einzelnen Testteams wurden mündlich erste Bewertungen ihrer Zählungen mitgeteilt. Aufgrund der noch geringen Menge an Zählungen wurde noch kein Newsletter verfasst.

7.2.12 Vorgänge wiederholen

Vorüberlegung:

Diese Vorgänge werden wiederholt, bis eine ausreichende Datenbasis für zuverlässige Schätzungen mit geringen Schätzabweichungen vorhanden ist.

Umsetzung:

Die Schätzungen waren bisher nicht aussagekräftig genug, um eine Prognose für die allgemeine Eignung für die intersoft AG zu treffen. Deswegen wurden die restlichen Teams noch nicht geschult. Der zyklische Vorgang aus Abbildung 35 wird wiederholt, bis ein ausreichender Datenbestand für Schätzungen vorhanden ist. Sind die ersten Schätzungen erfolgreich, so werden die restlichen Entwicklerteams geschult.

8 Ausblick und Fazit

Die Functionpoint Methode

Die COSMIC-FFP bieten eine viel versprechende Methode, um aus fachlicher Sicht den Umfang eines Fachkonzeptes bzw. einer Software zu messen. Die Methode scheint sich unter anderem aufgrund ihrer Objektivität besonders gut für die betreuende Firma zu eignen.

Tools die den Schätzprozess bei intersoft ausreichend unterstützen, sind auf dem Markt noch nicht vertreten. Deswegen wurde ein eigenes Tool entwickelt, welches den Zähl- und Schätzprozess unterstützt.

Die Einführung im Unternehmen

Es wurde ein Plan erstellt, nach welchem die Aufwandschätzung im Unternehmen eingeführt wurde. Es stellte sich heraus, dass die Einführung in ein Unternehmen vom Zeitplan her exakt im Voraus geplant werden muss. Ist dies nicht der Fall, so kann die Einführung sich stark verzögern und dadurch an Priorität verlieren. Ressourcen, wie der Leiter des Competence Centers, sind in dieser Zeit unproduktiv.

Zudem reicht es nicht, Testteams für die Einführung bereit zu stellen. Den Zählern der Testteams muss ein Zeitraum zur Verfügung gestellt werden, in dem sie Zeit haben zu zählen. Functionpoint Zählungen sollten vorhandene Projektarbeit nicht behindern oder verzögern. Die Schätzungen werden dann als notwendiges Übel angesehen, welches nur den bisherigen Projektablauf behindert.

Akzeptanzprobleme wurden im Vorfeld durch Aufzeigen des Nutzens von Aufwandschätzungen gemindert. Die Akzeptanz war bei den vom Competence Center geschulten Testteams weitaus höher als bei den ungeschulten Teams.

Ein bisher ungelöstes Problem besteht in der Motivierung der Entwickler, ihre Stunden in zwei Zeiterfassungssystemen zu dokumentieren. Viele Metriken sind erst dann möglich, wenn die benötigte Zeit für die Fertigstellung eines Teilbereiches direkt am entsprechenden Funktionalprozess oder der Datenbewegung dokumentiert wird. Diese Informationen über das Projekt sind natürlich nur im Tool vorhanden nicht aber im Zeiterfassungssystem der intersoft AG.

Erst diese genaue Zeiterfassung ermöglicht eine Vielzahl von Metriken und Projektcontrolling Möglichkeiten.

Zudem sind in den ersten Zählungen viele Unklarheiten aufgekommen, die durch das COSMIC-FFP Measurement Manual nicht beantwortet werden konnten. Komplexe Dinge wie Software lassen sich nicht durch einige wenige klar definierte Regeln messen.

Weiterentwicklungsmöglichkeiten

In Zukunft gilt es, Aspekte der COSMIC-FFP Methode auszumachen die nicht zu der intersoft AG passen und diese durch eigene Regeln zu ergänzen. Auch ist es denkbar, unvollständig beschriebene Fachkonzepte mit der Widgetpoint Methode zu schätzen, und somit die COSMIC-FFP Methode zu ergänzen.

Nach erfolgreicher Etablierung in der Firma könnte die Aufwandschätzung in das Qualitätsmanagementmodell der Firma integriert werden.

Es wäre ebenso denkbar, Fachkonzepte graphisch zu modellieren, so dass eine automatisierte Zählung ähnlich der Widgetpoints möglich wäre.

Das Tool könnte um eine Schnittstelle erweitert werden, die ermöglicht, Arbeitsstunden in das bestehende Zeiterfassungssystem zu exportieren. Somit würde das Problem der doppelten Zeiterfassung (siehe Anhang Tabelle 'Team') umgangen werden.

Bewertung

Die bisher gewonnenen Ergebnisse reichen quantitativ und qualitativ nicht aus, um zuverlässige Schätzprognosen vorzunehmen. Es müssen weitere Vergleichsdaten gesammelt werden, bis fundierte Schätzprognosen möglich sind. Danach gilt die Testphase als abgeschlossen, und die restlichen Teams können geschult werden.

Ausblick

Insgesamt rechnet der Autor mit einem Zeitraum von ca. einem Jahr, bis ausreichend Vergleichsdaten vorliegen, und die Teams eigenständig den Umfang von Software bestimmen können.

Es müssen einheitliche Regeln für unklare Zählsituationen definiert werden, damit alle Teams gleich schwer gewichtet werden. Sollten die Teams ihre Personenstunden im Tool dokumentieren, sollen die entsprechenden Metriken nachgepflegt werden.

Abkürzungsverzeichnis

BFC	Basisfunktionskomponente
CFSU	COSMIC Functional Size Unit
COSMIC	Common Software Measurement International Consortium
COSMIC-FFP	COSMIC Full Functionpoint
FP	Functionpoint
FFP	Full Functionpoint
FUB	Funktionsumfangbeschreibung
FUR	Functional User Requests – siehe FUB
IFPUG	International Function Point Users Group
ISBSG	International Software Benchmarking Standards Group
ISO	International Organisation for Standardisation
LOC	Lines Of Code
MIS	Management Information system
MKII FPM	Mark II Function Point Methode

Literaturverzeichnis

- [BOEHM] Barry W. Boehm: A Spiral Model of Software Development and Enhancement, <http://www.sce.carleton.ca/faculty/ajila/4106-5006/Spiral%20Model%20Boehm.pdf>, 10.09.2007
- [BUNDSCHUH04] Manfred Bundschuh / Axel Fabry: Aufwandschätzung von IT-Projekten, 2.Aufl., mitp –Verlag, Bonn 2004
- [COSMIC03] Common Software Measurement International Consortium: Measurement Manual (THE COSMIC IMPLEMENTATION GUIDE FOR ISO/IEC 19761: 2003), <http://www.geolog.etsmtl.ca/cosmic-ffp/manual.html>, 05.05.2007
- [COSMIC05] Common Software Measurement International Consortium: GUIDELINE FOR SIZING BUSINESS APPLICATION SOFTWARE USING COSMIC-FFP, <http://www.geolog.etsmtl.ca/cosmic-ffp/manual.html>, 05.05.2007
- [COSMICAD] Common Software Measurement International Consortium: Advantages of the COSMIC method, <http://www.cosmicon.com/advantagecs.asp> , 10.06.2007
- [DENERT92] Ernst Denert: Software-Engineering. Methodische Projektabwicklung, Springer, Berlin 1992
- [GARMUS00] David Garmus / David Herron: Function Point Analysis – Measurement Practices for Successful Software Projects, Addison-Wesley, Amsterdam 2000
- [IFPUGWEB] <http://www.ifpug.org>
- [ISBSG] ISBSG: Practical Project Estimation 2nd edition, <http://www.isbsg.org/Isbsg.Nsf/weben/Estimation%20Toolkit> 19.09.2007
- [HUERTEN05] Robert Hürten: Function-Point Analysis, Theorie und Praxis, Expert-Verlag, 2005
- [KRASEMANN01] Dr. Hartmut Krasemann: Erfolgreich mit Objektorientierung, 2. Aufl., Oldenbourg Verlag, München 2001
- [POEGENS05] Benjamin Poegens / Bertram Bock: Function-Point-Analyse – Ein Praxishandbuch, dpunkt.verlag GmbH, Heidelberg 2005
- [POHLSCH06] Dominik Pohlschneider: Aufwandschätzung auf Basis funktionaler Größenmessung – Konzeption eines Metrikprogramms bei der intersoft AG, Fachhochschule Flensburg, 04.09.2006
- [VOGELE] Frank Vogelegang: COSMIC Full Function Points The next generation of functional sizing, <http://www.lrgl.uqam.ca/publications/pdf/111p.pdf>, 19.09.2007
- [STANDISH95] Standish Group International: The Chaos Report 1995, http://www.standishgroup.com/sample_research/chaos_1994_1.php, 15.07.2007

Tabellenverzeichnis

Tabelle 1: Legende der Bewertung von Schätzverfahren	4
Tabelle 2: Bewertung - Funktionelles Einsatzgebiet	5
Tabelle 3: Bewertung - Projektart	5
Tabelle 4: Bewertung - Standardisierung.....	5
Tabelle 5: Bewertung - Geographische Verbreitung	5
Tabelle 6: Bewertung - Weiterentwicklung der Methoden	6
Tabelle 7: Bewertung - ISBSG Unterstützung	6
Tabelle 8: Bewertung - Aufwand	6
Tabelle 9: Bewertung - Toolunterstützung	6
Tabelle 10: Bewertung - Leichtigkeit.....	7
Tabelle 11: Bewertung - Robustheit.....	7
Tabelle 12: Bewertung - Transparenz	7
Tabelle 13: Bewertung - Konstanz	7
Tabelle 14: Bewertung - Objektivität	8
Tabelle 15: Zusammenfassung der Bewertung.....	8
Tabelle 16: Bewertung von Tools - Aufwand	10
Tabelle 17: Bewertung von Tools - Toolunterstützung	10
Tabelle 18: Bewertung von Tools - Leichtigkeit.....	10
Tabelle 19: Bewertung von Tools - Projektsteuerung.....	11
Tabelle 20: Bewertung von Tools - Schätzgenauigkeit	11
Tabelle 21: Bewertung von Tools - Transparenz	11
Tabelle 22: Bewertung von Tools - Objektivität	11
Tabelle 23: Bewertung von Tools - Flexibilität	12
Tabelle 24: Zusammenfassung der Bewertung von Tools.....	12
Tabelle 25: Die verschiedenen Kategorien von Widgets (vgl. [KRASEMANN01] S. 311).	26
Tabelle 26: Legende der Zählung von Widgets.....	29
Tabelle 27: Ergebnisse der Zählung von Widgets	29
Tabelle 28: Gegenüberstellung Widgetpoints - Cosmic-FFP	30
Tabelle 29: Legende der Bewertung von COSMIC-FPP Tools.....	34
Tabelle 30: Bewertung des Tools MeterIT-Cosmic	34
Tabelle 31: Bewertung des Tools Experience Pro 3.1	35
Tabelle 32: Bewertung des Tools KnowledgePLAN	35
Tabelle 33: Bewertung des Tools Siesta.....	36
Tabelle 34: Bewertung des Tools bei Eigenentwicklung	36
Tabelle 35: Zusammenfassung der Bewertung von COSMIC-FPP Tools	37
Tabelle 36: Datenbanktabelle 'Projekt'.....	88
Tabelle 37: Datenbanktabelle 'Benutzer'.....	89
Tabelle 38: Datenbanktabelle 'Team'.....	90
Tabelle 39: Datenbanktabelle 'Personenstunden'.....	91
Tabelle 40: Datenbanktabelle 'Funktionalprozess' (FP).....	92
Tabelle 41: Datenbanktabelle 'Meilenstein'.....	93
Tabelle 42: Datenbanktabelle 'Status'	93
Tabelle 43: Datenbanktabelle 'Veraenderungstyp'	94
Tabelle 44: Datenbanktabelle Basisfunktionskomponente (BFC)	96
Tabelle 45: Datenbanktabelle Typ.....	96

Abbildungsverzeichnis

Abbildung 1: Ablauf einer Schätzung	3
Abbildung 2: Verfahren zur Berechnung des Aufwands.....	9
Abbildung 3: Datenbewegungen und Schichten (vgl. [COSMIC] S.19)	15
Abbildung 4: Die 'COSMIC-FFP Mapping' Phase (vgl. [COSMIC] S. 30)	16
Abbildung 5: Allgemeines Softwaremodell zum Messen der funktionalen Größe (vgl. [COSMIC] S. 22)	17
Abbildung 6: Die vier Datenbewegungen im Frontend / Backend	18
Abbildung 7: Die Mapping-Phase bei der intersoft AG(vgl. Abb. 2)	21
Abbildung 8: Zustandsdiagramm des GeVo TAB der intersoft AG	22
Abbildung 9: Unterteilung des GeVo TAB in Funktionalprozesse.....	23
Abbildung 10: Die Datenbewegungen der LösCHFunktion des GeVo TAB.....	24
Abbildung 11: Anzahl und Typ der benötigten Basisfunktionskomponenten	25
Abbildung 12: Graphische Oberfläche des Geschäftsvorfalles TAB	27
Abbildung 13: Vorkommende Widgets farblich markiert	28
Abbildung 14: Gegenüberstellung der Zählung von Widgetpoints und COSMIC-FFP	31
Abbildung 15: Fachliche Architektur	40
Abbildung 16: technische Architektur	41
Abbildung 17: Spiralmodell nach Barry W. Boehm [BOEHM]	44
Abbildung 18: Datenbankdiagramm des Tools	45
Abbildung 19: Businessobjekt 'Projekt'	46
Abbildung 20: Businessobjekt Funktionalprozess (FP).....	47
Abbildung 21: Businessobjekt Basisfunktionskomponente (BFC).....	47
Abbildung 22: Eingabemaske zur Erfassung der FP.....	48
Abbildung 23: Historische Projektwerte mit Projektkurve	50
Abbildung 24: Prozentuale Verteilung der vier Basisfunktionskomponenten.....	51
Abbildung 25: Implementierte Cfsu gemessen am Abgabetermin	52
Abbildung 26: Produktivitätsrate	53
Abbildung 27: Anzahl der Fehler pro Functionpoint	54
Abbildung 28: Fehler pro Functionpoint	54
Abbildung 29: Arbeitsstundenverteilung eines Mitarbeiters	55
Abbildung 30: Verteilung der Personenstunden eines Projektes	55
Abbildung 31: Effizienz der Implementierung	56
Abbildung 32: Unplanmäßiger Funktionszuwachs	56
Abbildung 33: Umsatz pro FP	57
Abbildung 34: Kosten pro Functionpoint	58
Abbildung 35: Zyklischer Ablauf der Gewinnung von historischen Daten	61
Abbildung 36: Datenbanktabelle 'Projekt'	87
Abbildung 37: Datenbanktabelle 'Benutzer'	89
Abbildung 38: Datenbanktabelle 'Team'	90
Abbildung 39: Datenbanktabelle 'Personenstunden'	91
Abbildung 40: Datenbanktabelle 'Funktionalprozess' (FP)	92
Abbildung 41: Datenbanktabelle 'Meilenstein'	93
Abbildung 42: Datenbanktabelle 'Status'	93
Abbildung 43: Datenbanktabelle 'Veränderungstyp'	94
Abbildung 44: Datenbanktabelle Basisfunktionskomponente (BFC)	95
Abbildung 45: Datenbanktabelle Typ	96
Abbildung 46: Beziehung FP - FP	97
Abbildung 47: Beziehung Personenstunden - FP	97
Abbildung 48: Beziehung Projekt - Projekt	98
Abbildung 49: Beziehung Projekt - Benutzer	99
Abbildung 50: Beziehung Projekt - Team	100
Abbildung 51: Beziehung Projekt - Meilenstein	101
Abbildung 52: Beziehung Projekt - Status	102
Abbildung 53: Beziehung BFC - Veränderungsstyp	103

Abbildung 54: Beziehung BFC - Typ	103
Abbildung 55: Beziehung Projekt - FP.....	104
Abbildung 56: Beziehung BFC - FP	105
Abbildung 57: Beziehung BFC - BFC.....	105

Anhang

Präsentationsunterlagen der Schulung zum COSMIC-FFP Zähler

intersoft:



➤ Functionpoints

Veranstaltung: Einführung von Full Function Points bei der intersoft AG
Referent: Jan Kurniawan
Datum: 17.09.2007

© 2007 intersoft AG

intersoft:

Motivation

Problematik der intersoft AG

- Unzuverlässigkeit und oftmals Ungenauigkeit bei bisherigen Schätzungen
- Schätzerfahrung hängt an wenigen Personen
- Dezentralisierte Informationen über den Aufwand abgeschlossener Projekte
- Nichtberücksichtigung und fehlender Austausch von Erfahrung aus bisherigen Schätzungen
- Subjektivität und Nichtreproduzierbarkeit bisheriger Schätzungen

Daraus resultiert

- Starke Abweichung der Schätzung von der Realität
- Fehlende Maßeinheit für die Softwaregröße
- Der Umfang eines Projektes kann einem fachfremden Kunden nur schwer veranschaulicht werden
- Planung von Kosten, Umfang und Projektlaufzeit ist nicht fundiert
- Überstunden
- schlechte Qualität
- Nachträgliches Abmelden von Funktionalität
- Nachlieferungen

© 2007 intersoft AG

Konkret

- Die Schätzabweichung bei intersoft liegt bei 80%

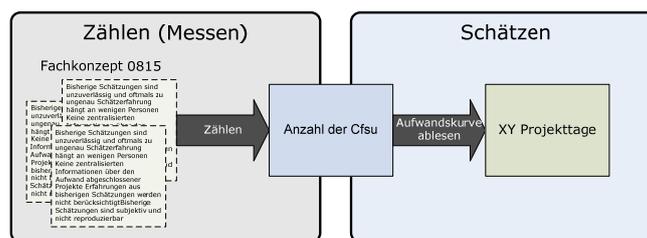
Lösung

- Funktionsumfang der Software mit einem Verfahren reproduzierbar zählen und den Aufwand ermitteln
- Zentrale Datenhaltung für Zählungen und Schätzungen

© 2007 intersoft AG

Womit zählen?

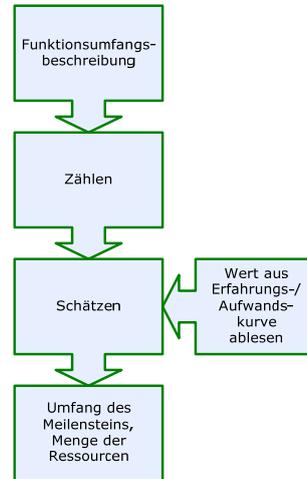
- D. Pohlschneider ermittelte das passende Verfahren für die intersoft AG: Die Cosmic-FFP (The **C**osmic **S**oftware **M**easurement **I**nternational **C**onsortium - **F**ull **F**unction **P**oints)
 - Keine fachlichen Schätzerfahrung oder Vergleiche zu alten Projekten benötigt
 - Objektiv, da frei von subjektiven Gewichtungen
 - CFFP bedürfen keiner Schätzerfahrung und sind gegenüber anderen Function Point Methoden, reproduzierbar



© 2007 intersoft AG

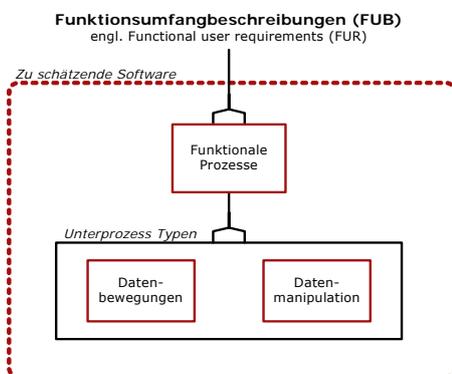
Zählen und Schätzen

- Zählen und Schätzen sind die elementaren Funktionen von Function Points
 - Gezählt wird der funktionale Umfang der Software
 - Aufgrund der Ergebnisse der Zählung sowie anderer Faktoren, werden Projektdauer und Ressourcen bestimmt



© 2007 intersoft AG

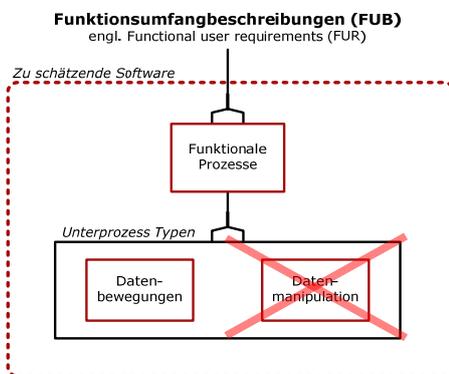
Was wird gezählt?



- Zerlegung der Funktionsumfangbeschreibung in kleinstmögliche Funktionale Prozesse (FP)
- Ein Funktionalprozess ist eine zusammenhängende und unabhängig ausführbare Kombination von Datenbewegungen
- Kleinste sinnvolle und in sich abgeschlossene Aktivität

© 2007 intersoft AG

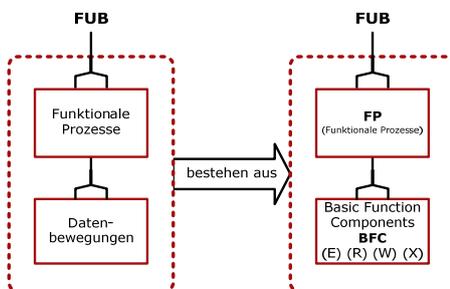
Was wird gezählt?



- Es werden ausschließlich Datenbewegungen gezählt, keine Form der Datenmanipulation
- Jeder Funktionalprozess wird nur ein mal gezählt, egal wie oft er Verwendung findet

© 2007 intersoft AG

Was wird gezählt?



- Datenbewegungen (**BFC**) gliedern sich in wiederum in vier Typen:
 - Entry(E)
 - Read(R)
 - Write(W)
 - Exit(X)
- Alle Typen sind gleich schwer gewichtet

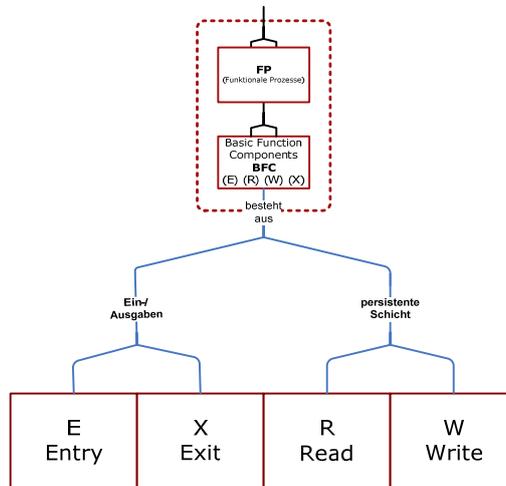
© 2007 intersoft AG

Vier Datenbewegungen im Detail

- **Entry:** Datenbewegung, welche Daten einer Datengruppe über die Anwendungsgrenze zum funktionalen Prozess bewegt
- **Write:** bewegt Daten einer Datengruppe in einen persistenten Speicher
- **Read:** liest, analog zum Write, Daten einer Datengruppe aus einem persistenten Speicher
- **Exit:** bewegt Daten einer Datengruppe über die Anwendungsgrenze zum Benutzer

© 2007 intersoft AG

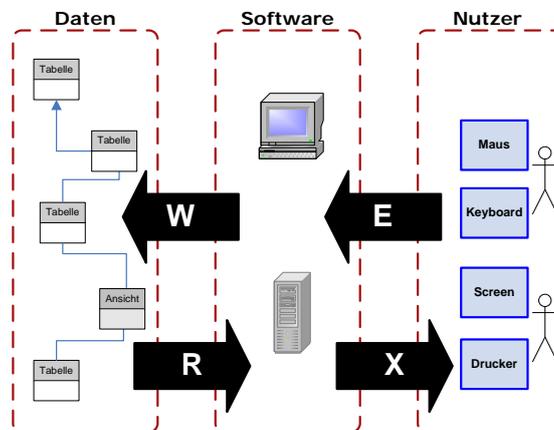
Vier Datenbewegungen im Detail



© 2007 intersoft AG

Vier Datenbewegungen im Detail

Basisfunktionskomponenten BFC
(Basic Function Component)



© 2007 intersoft AG

Definition von Datengruppen

- Eine Datengruppe besteht aus einer eindeutigen, nicht leeren, unsortierten, nicht redundanten Menge von Datenattributen
- Alle enthaltenen Daten einer Datengruppe beziehen sich auf das selbe Interessenobjekt (Entity / object of interest)
- Datengruppen können mehrere Zustände haben: Persistent und Transient
- Bei lifestream können Datengruppen praktisch mit Tabellen in der Datenbank gleichgesetzt werden
- Wiederholte Datenbewegungen eines Typs (BFC), welche die selbe Datengruppe betreffen, werden nicht mehrfach gezählt
- Ein Funktionalprozess muss mindestens zwei Datenbewegungen haben: ENTRY – WRITE oder ENTRY EXIT

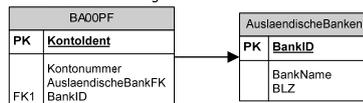
© 2007 intersoft AG

Definition Datengruppe

Beispiel: mit **einem** Read (R) werden Kontonummer, *AuslaendischeBank* und *AuslaendischeBankleitzahl* aus BA00PF gelesen.

BA00PF	
PK	<u>Kontolident</u>
	AuslaendischeBank AuslaendischeBankleitzahl Kontonummer

Da die Tabelle BA00PF gegen die dritte Normalform verstößt Müssen **zwei** Read (R) gezählt werden, da normalerweise aus zwei Tabellen gelesen werden müsste.

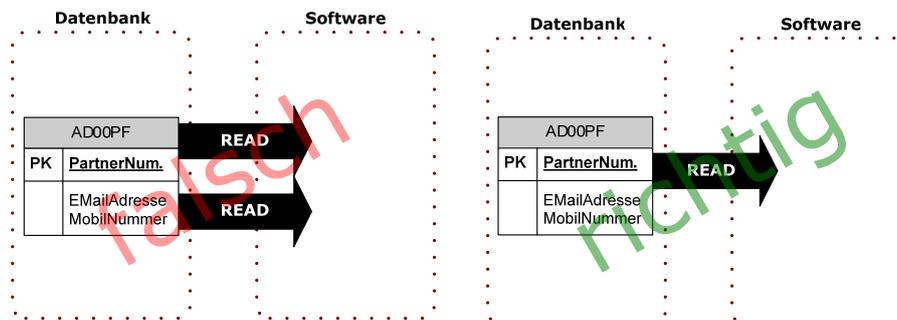


- Pro Datengruppe bewegte Daten innerhalb eines FP wird ein Cfsu berechnet
- Eine Datengruppe entspricht einer Entität (Tabelle) in der 3. Normalform
- Redundante Daten einer Tabelle werden also entsprechend der Anzahl der Tabellen in der 3. Normalform gezählt

© 2007 intersoft AG

Definition Datengruppe

- Für jede Bewegung von Daten einer Datengruppe wird eine Cfsu berechnet
- Wiederholte Datenbewegungen eines Typs (BFC – E,W,R,X) welche die selbe Datengruppe betreffen, werden nicht mehrfach gezählt

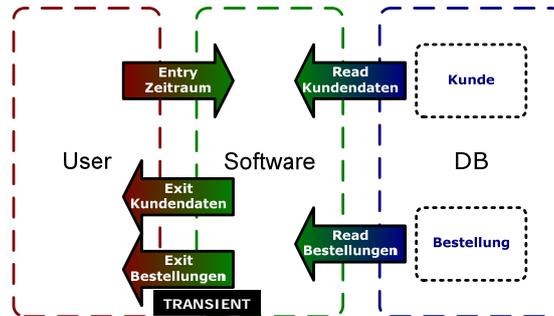


© 2007 intersoft AG

Beispiel einer Zählung anhand eines Funktionsprozess

Funktionalprozess Beschreibung:

- Es sollen alle Kunden mit der Anzahl ihrer getätigten Bestellungen innerhalb eines Zeitraums ausgegeben werden
- ‚Kunde‘ und ‚Bestellung‘ sind jeweils eine Tabelle in der DB

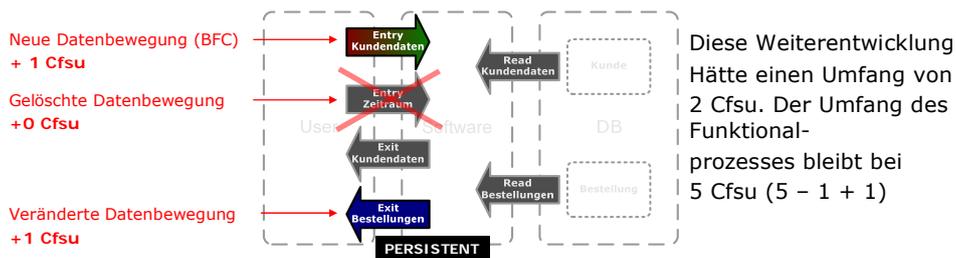


- Es ergeben sich 5 Cfsu (1E + 2R + 2X)

© 2007 intersoft AG

Weiterentwicklungsprojekte

- Der Aufwand einer Weiterentwicklung wird durch die Anzahl der hinzugefügten oder veränderten Datenbewegungen bestimmt
- Durch löschen einer Datenbewegung entsteht kein Aufwand
- Beispiel:
 - Statt eines Zeitraums, werden jetzt Kundendaten selektiert
 - Statt der Anzahl der Bestellungen werden nun alle Bestellungen ausgegeben



Diese Weiterentwicklung hätte einen Umfang von 2 Cfsu. Der Umfang des Funktionalprozesses bleibt bei 5 Cfsu (5 - 1 + 1)

© 2007 intersoft AG

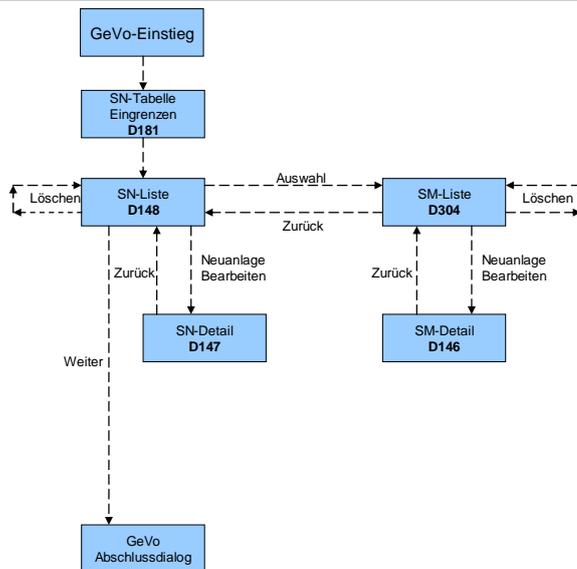
Zählung eines GeVo

Anforderungen an den GeVo TAB (Quelle: Doku-Pool)

- Der GeVo Tab ist dafür vorgesehen, um Einträge aus den Schlüsselstabellen (SN und SM) auszulesen und deren Details darzustellen, zu Bearbeiten, zu Löschen oder Neuanlagen zu tätigen
- Der GeVo Tab setzt sich aus fünf Dialogen zusammen. Der Eingrenzdialog (D181 (1)),in welchem die gewünschte SN-Tabelle(n) angegeben wird
- Dem Listendialog mit den Einträgen innerhalb der SN-Tabelle (D148(2)). Dann der Detaildialog mit allen Details der einzelnen Listeneinträge aus der SN-Listentabelle (D147(3))
- Als viertes der Listendialog mit den zur SN gehörigen Werten aus der entsprechenden SM-Tabelle (D304(4)) und als letztes der Detaildialog, mit den Details zu jedem SM-Listeneintrag (D146(5))
- Wichtig ist: Der GeVo Tab verfügt über eine MAV (Mehrfach-Aktions-Verarbeitung). Dies bedeutet, dass in den Listendialogen (D148(6)/D304(7)) via Checkbox mehrere Werte gleichzeitig selektiert werden können, dessen Detaildialoge dann der Reihe nach bearbeitet werden können**

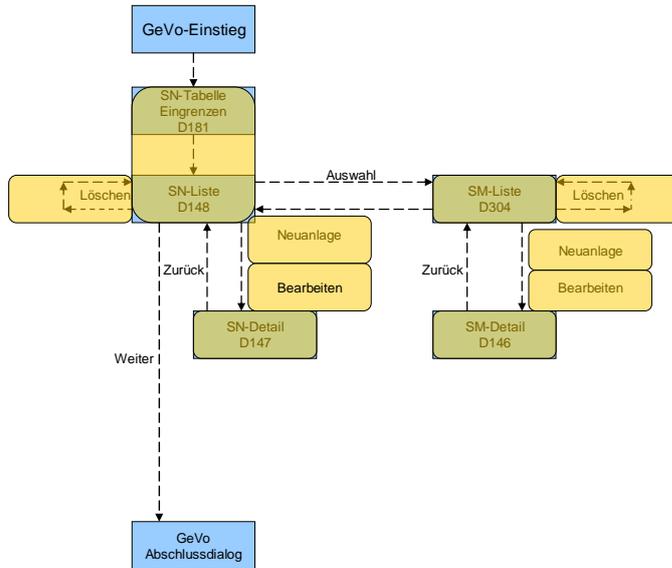
© 2007 intersoft AG

Diagramm des GeVo TAB (Quelle: Doku-Pool)



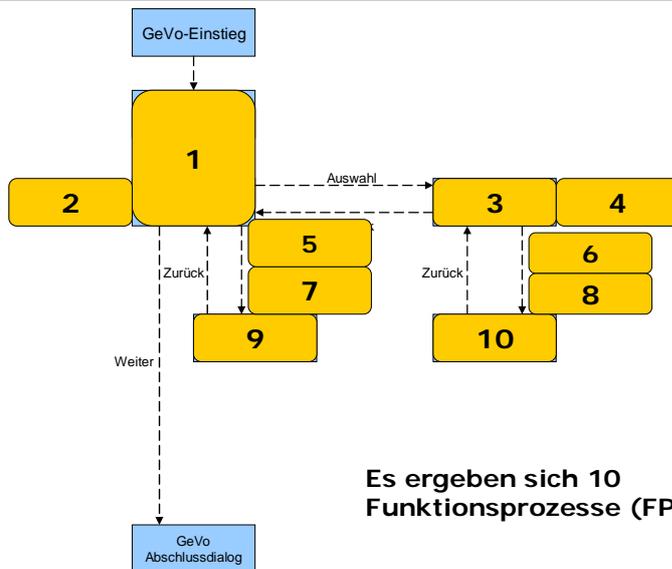
© 2007 intersoft AG

Unterteilung in Funktionsprozesse (FP)



© 2007 intersoft AG

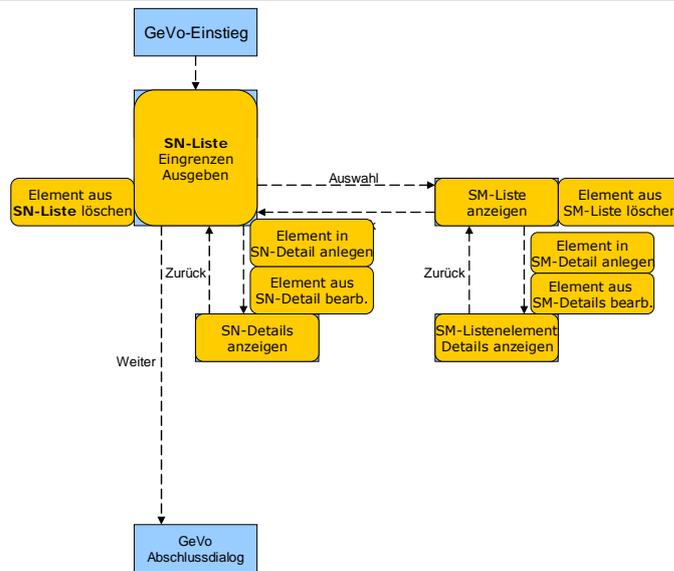
Unterteilung in Funktionsprozesse (FP)



Es ergeben sich 10 Funktionsprozesse (FP)

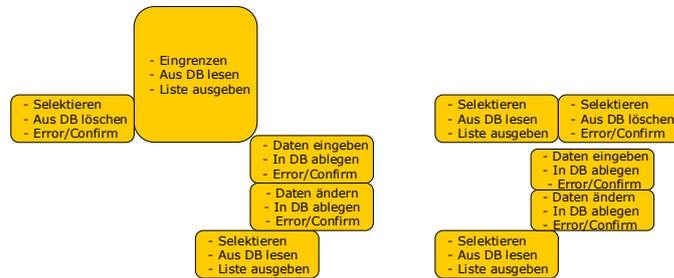
© 2007 intersoft AG

Unterteilung in Funktionsprozesse (FP)



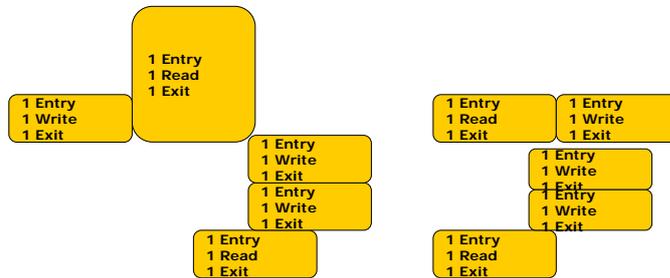
© 2007 intersoft AG

Definieren der Datenbewegungen



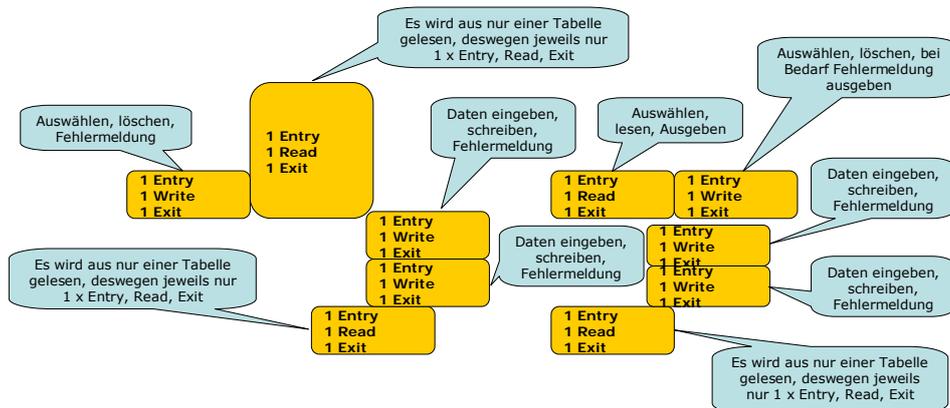
© 2007 intersoft AG

Berechnung der benötigten Basic Functional Components (BFC)

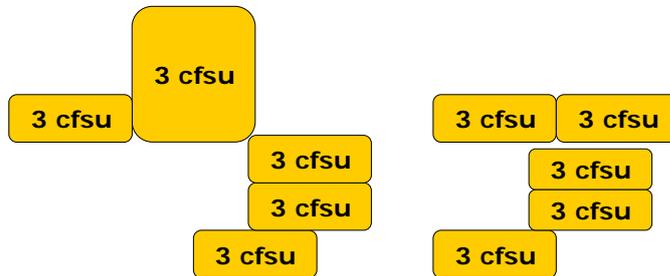


© 2007 intersoft AG

Berechnung der benötigten Basic Functional Components (BFC)



© 2007 intersoft AG

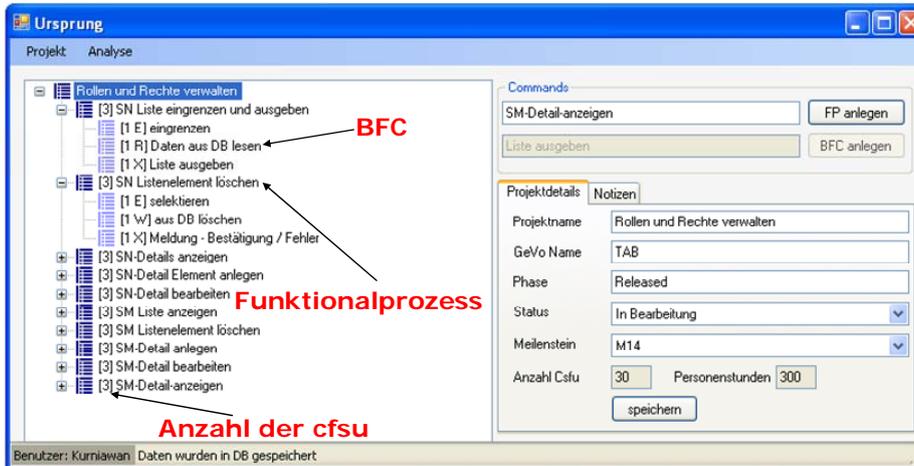


© 2007 intersoft AG

**Die Anwendung besteht
aus 10 FP und 30 cfsu**

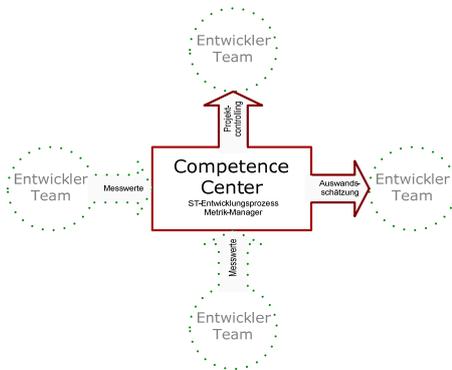
© 2007 intersoft AG

Gemessene Daten im Tool dokumentieren



© 2007 intersoft AG

Analyse und Prognose



- Zählungen werden vom Competence Center ausgewertet
- Aufwand wird an Aufwandskurve abgelesen
- Projektcontrolling überwacht Abweichungen im laufenden Entwicklungsprozess

© 2007 intersoft AG

Datenbanktabellen und Beziehungen

Tabelle: 'Projekt'

Beschreibung: In der Tabelle werden die allgemeinen, das Projekt betreffenden Daten gespeichert. Für jedes Neu- / Aufbauprojekt gibt es einen Eintrag in der Projekttabelle. Das Projekt dient im Tool gleichzeitig als oberster Knoten des Baumes. Unter dem Projekt befinden sich die Funktionalprozesse. Die zwei Arten von Projekten werden über das Feld AufbauProjektFK definiert: Ist das Feld leer, so handelt es sich um ein 'Neuprojekt'. Enthält das Feld einen Fremdschlüssel, so handelt es sich um ein Weiterentwicklungsprojekt, welches auf dem Projekt mit dem Primärschlüssel basiert. Unter anderem ist somit eine Hierarchie von Aufbauprojekten in der Datenbank abgebildet.

Abbildung als Objekt: Alle Daten der Tabelle 'Projekt', werden in die Klasse 'Projekt' der Anwendung geladen.

Projekt			
Spaltenname	Datentyp	NULL zulassen	
ProjektID	int	<input type="checkbox"/>	
AufbauPro...	int	<input checked="" type="checkbox"/>	
StatusFK	int	<input type="checkbox"/>	
Projektname	varchar(50)	<input type="checkbox"/>	
GeVoName	varchar(50)	<input checked="" type="checkbox"/>	
Phase	varchar(50)	<input checked="" type="checkbox"/>	
Notizen	varchar(300)	<input checked="" type="checkbox"/>	
BenutzerFK	int	<input checked="" type="checkbox"/>	
Meilenstei...	int	<input type="checkbox"/>	
entwicklun...	datetime	<input checked="" type="checkbox"/>	
voraussich...	datetime	<input checked="" type="checkbox"/>	
tatsaechlic...	datetime	<input checked="" type="checkbox"/>	
geschaetz...	int	<input checked="" type="checkbox"/>	
TeamFK	int	<input checked="" type="checkbox"/>	
VerfasserFK	int	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Abbildung 36: Datenbanktabelle 'Projekt'

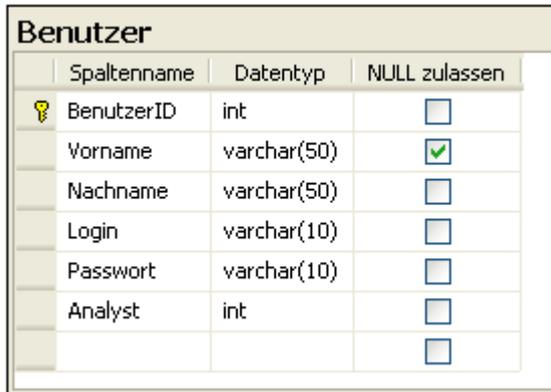
Feldname	Datentyp	Beschreibung
ProjektID	int	Primärschlüssel der Tabelle.
AufbauProjektFK	int	Ist ein Aufbauprojekt, wenn es einen Fremdschlüssel enthält. Dieser gibt das Basisprojekt an. Ist ein Neuprojekt wenn das Feld den Wert NULL enthält.
StatusFK	int	Fremdschlüssel für Status Tabelle. Der Status gibt den aktuellen Stand der FP-Erfassung an.
Projektname	Varchar (50)	Name des Projektes.
GeVoName	Varchar	Name des Geschäftsvorfalles.

	(50)	
Phase	Varchar (50)	Phase in der sich das Projekt aktuell befindet.
Notizen	Varchar (300)	Ein Notizfeld für alle Informationen, die durch die anderen Felder nicht definiert wurden.
BenutzerFK	int	Fremdschlüssel für die Benutzer Tabelle. Ist hier ein Schlüssel eingetragen, so wird das Projekt von der entsprechenden Person gerade benutzt. Nach Beenden der Nutzung wird das Feld geleert. Ein Nutzer kann durch die Programmlogik kein Projekt öffnen, das benutzt wird.
MeilensteinFK	int	Meilenstein, in dem das Projekt entwickelt wird.
entwicklungsStartDatum	datetime	Datum, an dem mit der Implementierung angefangen wird.
voraussichtlichesEnde	datetime	Datum, für das geplante Ende der Implementierung.
tatsaechlichesEnde	datetime	Datum, an dem die Implementierung abgeschlossen wurde.
geschaetzteCfsu	int	Geschätzte Anzahl von Personenstunden, die das Projekt voraussichtlich benötigt. Redundanter Wert, der sich aus Einträgen der Personenstundentabelle zusammensetzt. Um Datenbankzugriffe zu minimieren, wurde dieser Wert angelegt.
TeamFK	int	Fremdschlüssel zur Team Tabelle. Team, welches das Projekt realisiert.
VerfasserFK	int	Fremdschlüssel zur Benutzer Tabelle. Gibt an, welche Person dieses Projekt im Tool erfassen.

Tabelle 36: Datenbanktabelle 'Projekt'

Tabelle: 'Benutzer'

Beschreibung: Die Tabelle beinhaltet Benutzerdaten für Personen, die Zugriff auf das Tool haben. Die Tabelle definiert auch, wer Zugriff auf das Analysetool hat und wer nur Functionpoints erfassen darf. Meldet sich der Benutzer am System an, so wird geprüft, ob das entsprechende Login und Passwort vorhanden sind. Ist dies der Fall, wird der Nutzer am System angemeldet. Für die Pflege der Benutzerdaten besteht keine Anwendung. Diese können unkompliziert über das graphische Interface des Microsoft SQL Server Management Studio gepflegt werden.



Spaltenname	Datentyp	NULL zulassen
BenutzerID	int	<input type="checkbox"/>
Vorname	varchar(50)	<input checked="" type="checkbox"/>
Nachname	varchar(50)	<input type="checkbox"/>
Login	varchar(10)	<input type="checkbox"/>
Passwort	varchar(10)	<input type="checkbox"/>
Analyst	int	<input type="checkbox"/>

Abbildung 37: Datenbanktabelle 'Benutzer'

Feldname	Datentyp	Beschreibung
BenutzerID	int	Primärschlüssel der Tabelle.
Vorname	varchar(50)	Vorname des Benutzers.
Nachname	varchar(50)	Nachname des Benutzers.
Login	varchar(10)	Login des Benutzers.
Passwort	varchar(10)	Passwort des Benutzers in unverschlüsselter Form.
Analyst	int	Boolescher Wert (0 = false; 1 = true) der definiert, welcher Benutzer Analyst ist. 1 = Analyst.

Tabelle 37: Datenbanktabelle 'Benutzer'

Tabelle: 'Team'

Beschreibung: Enthält eine kurze Beschreibung der Teams. Für die Pflege der Teaminformationen besteht keine Anwendung. Diese können unkompliziert über das graphische Interface des Microsoft SQL Server Management Studio gepflegt werden.

Team			
	Spaltenname	Datentyp	NULL zula...
🔑	TeamID	int	<input type="checkbox"/>
	Kurzbezeichnung	varchar(50)	<input type="checkbox"/>
	Bezeichnung	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Abbildung 38: Datenbanktabelle 'Team'

Feldname	Datentyp	Beschreibung
TeamID	int	Primärschlüssel der Tabelle.
Kurzbezeichnung	varchar(50)	Kurzbezeichnung des Teams.
Bezeichnung	varchar(50)	Beschreibung des Teams.

Tabelle 38: Datenbanktabelle 'Team'

Tabelle: Personenstunden

Beschreibung: Tabelle enthält jeweils einen Personenstundeneintrag. Die Entwickler dokumentieren hiermit, wie viele Personenstunden sie für einen Funktionalprozess benötigt haben. Die ursprüngliche Idee war, die entsprechenden Personenstunden aus der internen Zeiterfassung der betreuenden Firma zu laden. Die Daten stellten sich aber als zu wenig detailliert heraus, und dies würde ein Projektcontrolling unmöglich machen.

Abbildung als Objekt: Alle Daten der Tabelle 'Personenstunden' werden bei Bedarf in eine Instanz der Klasse 'Personenstunden' der Anwendung geladen.

Personenstunden			
	Spaltenname	Datentyp	NULL zula...
	PersonenstundenID	int	<input type="checkbox"/>
	FUBFK	int	<input type="checkbox"/>
	Datum	datetime	<input type="checkbox"/>
	BenutzerFK	int	<input type="checkbox"/>
	Stunden	int	<input type="checkbox"/>
			<input type="checkbox"/>

Abbildung 39: Datenbanktabelle 'Personenstunden'

Feldname	Datentyp	Beschreibung
PersonenstundenID	int	Primärschlüssel der Tabelle.
FUBFK	int	Fremdschlüssel zur Funktionalprozess Tabelle. Gibt an zu welchem Funktionalprozess dieser Eintrag gehört.
Datum	datetime	Tag an dem die Personenstunden geleistet wurden.
BenutzerFK	int	Fremdschlüssel zur Benutzer Tabelle. Gibt an, welcher Nutzer die Personenstunden geleistet hat.
Stunden	int	Anzahl der geleisteten Personenstunden.

Tabelle 39: Datenbanktabelle 'Personenstunden'

Tabelle: FP

Beschreibung: Die Tabelle beinhaltet alle Daten, die zu einem Funktionalprozess gehören. Ein Funktionalprozess befindet sich in der Anwendung immer unter einem Projektknoten. Deshalb hat die Tabelle einen Fremdschlüssel zu dem zugehörigen Projekt. Für die richtige Anordnung im Baum dient der Fremdschlüssel Parentsibling (oberes Geschwister). Dieser gibt an, welcher Funktionalprozess sich in der Baumstruktur über dem Funktionalprozess befindet. Diese Anordnung dient nur der Übersichtlichkeit.

Abbildung als Objekt: Alle Daten der Tabelle 'FUB', werden bei Bedarf in eine Instanz der Klasse 'FUB' der Anwendung geladen.

FP			
	Spaltenname	Datentyp	NULL zula...
	FUBID	int	<input type="checkbox"/>
	ProjektFK	int	<input type="checkbox"/>
	FUBName	varchar(50)	<input type="checkbox"/>
	Notizen	varchar(300)	<input checked="" type="checkbox"/>
	ParentSiblingFK	int	<input checked="" type="checkbox"/>
	Aufwand	int	<input type="checkbox"/>
	ImplementierungsDatum	datetime	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Abbildung 40: Datenbanktabelle 'Funktionalprozess' (FP)

Feldname	Datentyp	Beschreibung
FUBID	int	Primärschlüssel der Tabelle.
ProjektFK	int	Fremdschlüssel zur Projekt-Tabelle. Zeigt an zu welchem Projekt dieser Funktionalprozess gehört.
FUBName	varchar(50)	Name des Funktionalprozess.
Notizen	varchar(300)	Ein Notizfeld für alle Informationen, die durch die anderen Felder nicht definiert wurden.
ParentSiblingFK	int	Fremdschlüssel zum Funktionalprozess, der auf der gleichen Ebene des gleichen Projekts liegt. Dies dient nur zur Übersichtlichkeit; die Anordnung der FP spielt für die Zählung keine Rolle.
Aufwand	int	Summe aller Personenstunden aus der Tabelle Personenstunden, die zu diesem FP gehören.
ImplementierungsDatum	datetime	Datum, an dem dieser FP implementiert wurde, bzw. fertig gestellt.

Tabelle 40: Datenbanktabelle 'Funktionalprozess' (FP)

Tabelle: 'Meilenstein'

Beschreibung: Die Tabelle enthält alle Meilensteine, denen Projekte zugeordnet werden können. Meilensteine sind festgelegte Zeiträume, in denen Projekte realisiert werden sollen. Es wird nur die Bezeichnung des Meilensteins gespeichert, nicht aber der Zeitraum.

Abbildung als Objekt: Alle Daten der Tabelle 'Meilenstein', werden bei Bedarf in eine Instanz der Klasse 'Meilenstein' der Anwendung geladen.

Spaltenname	Datentyp	NULL zulassen
MeilensteinID	int	<input type="checkbox"/>
MeilensteinName	varchar(50)	<input type="checkbox"/>
Jahr	int	<input type="checkbox"/>

Abbildung 41: Datenbanktabelle 'Meilenstein'

Feldname	Datentyp	Beschreibung
MeilensteinID	int	Primärschlüssel der Tabelle.
MeilensteinName	varchar(50)	Name des Meilensteins.
Jahr	int	Jahr in dem der Meilenstein liegt.

Tabelle 41: Datenbanktabelle 'Meilenstein'

Tabelle: 'Status'

Beschreibung: Tabelle enthält den aktuellen Status der Erfassung, bzw. Analyse eines Projektes. Es wird eine einfache Kurzbezeichnung gespeichert.

Abbildung als Objekt: Alle Daten der Tabelle 'Status' werden bei Bedarf in eine Instanz der Klasse 'Status' der Anwendung geladen.

Spaltenname	Datentyp	NULL zulassen
StatusID	int	<input type="checkbox"/>
Name	varchar(50)	<input type="checkbox"/>

Abbildung 42: Datenbanktabelle 'Status'

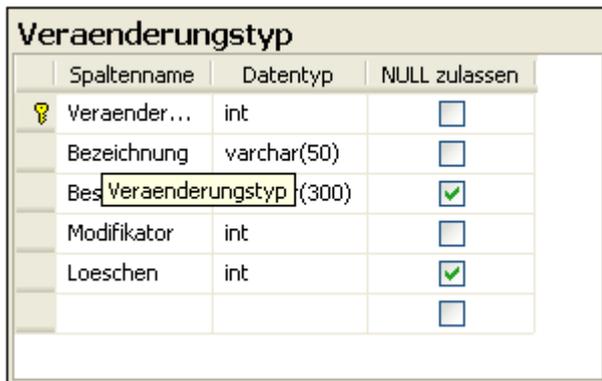
Feldname	Datentyp	Beschreibung
StatusID	int	Primärschlüssel der Tabelle.
Name	varchar(50)	Kurzbezeichnung des Status.

Tabelle 42: Datenbanktabelle 'Status'

Tabelle: 'Veraenderungstyp'

Beschreibung: Tabelle enthält die verschiedenen Veränderungstypen einer Basisfunktionskomponente. Ein Veränderungstyp gibt einem Weiterentwicklungsprojekt an, ob die Basisfunktionskomponente verändert, gelöscht oder im initialen Ursprung ist.

Abbildung als Objekt: Alle Daten der Tabelle 'Status', werden in ein Dropdownmenü der Klasse 'Mainwindow' geladen.



Spaltenname	Datentyp	NULL zulassen
Veraender...	int	<input type="checkbox"/>
Bezeichnung	varchar(50)	<input type="checkbox"/>
Beschreibung	varchar(300)	<input checked="" type="checkbox"/>
Modifikator	int	<input type="checkbox"/>
Loeschen	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Abbildung 43: Datenbanktabelle 'Veraenderungstyp'

Feldname	Datentyp	Beschreibung
VeraenderungstypID	Int	Primärschlüssel der Tabelle.
Bezeichnung	varchar(50)	Kurzbezeichnung des Veränderungstyps.
Beschreibung	varchar(300)	Beschreibung des Veränderungstyps
Modifikator	Int	Wert der zu den Cfsu hinzuaddiert wird.
Loeschen	Int	Gibt an, ob bei dieser Art des Veränderungstyps der BFC gelöscht werden soll.

Tabelle 43: Datenbanktabelle 'Veraenderungstyp'

Tabelle: 'Basisfunktionskomponente'

Beschreibung: Die Tabelle beinhaltet alle Daten die zu einer Basisfunktionskomponente gehören. Eine Basisfunktionskomponente gehört immer zu einem Funktionalprozess und hat deswegen auch einen Fremdschlüssel (FUBFK), der auf die Tabelle verweist. Sollte die BFC zu einem Aufbauprojekt hinzugefügt werden, so handelt es sich um eine BFC die von den anderen unterschieden werden muss. Hierfür wird das Feld Aufbauprojekt / Neufunktion mit einem Booleschen Wert (0 = false; 1 = true) belegt. Für die richtige Anordnung im Baum dient der Fremdschlüssel Parentsibling (oberes Geschwister). Dieser gibt an, welche Basisfunktionskomponente sich in der Baumstruktur über der Basisfunktionskomponente befindet.

Abbildung als Objekt: Alle Daten der Tabelle 'BFC' werden bei Bedarf in eine Instanz der Klasse 'BFC' der Anwendung geladen.

BFC			
	Spaltenname	Datentyp	NULL zulassen
	BFCID	int	<input type="checkbox"/>
	FUBFK	int	<input type="checkbox"/>
	BFCTyp	int	<input type="checkbox"/>
	BFCName	varchar(50)	<input type="checkbox"/>
	Csfu	int	<input type="checkbox"/>
	Notizen	varchar(300)	<input checked="" type="checkbox"/>
	ParentSiblingFK	int	<input checked="" type="checkbox"/>
	VeraenderungstypFK	int	<input checked="" type="checkbox"/>
	AufbauprojektNeufunk...	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Abbildung 44: Datenbanktabelle Basisfunktionskomponente (BFC)

Feldname	Datentyp	Beschreibung
BFCID	int	Primärschlüssel der Tabelle.
FUBFK	int	Fremdschlüssel für den übergeordneten Funktionalprozess.
BFCTyp	int	Fremdschlüssel für den Typ der Basisfunktionskomponente.
BFCName	Varchar (50)	Name der Basisfunktionskomponente.
Csfu	int	Anzahl der Csfu die diese Basisfunktionskomponente umfasst. Default ist 1 Csfu.
Notizen	Varchar (300)	Ein Notizfeld für alle Informationen, die durch die anderen Felder nicht definiert wurden.
ParentSiblingFK	int	Fremdschlüssel zur Basisfunktionskomponente, die auf der gleichen Ebene des gleichen Funktionalprozesses liegen. Dies dient nur zur Übersichtlichkeit; die Anordnung der Basisfunktionskomponenten spielt für die Zählung keine Rolle.
VeraenderungstypFK	int	Nur bei Aufbauprojekten relevant. Wird bei Aufbauprojekten die Basisfunktionskomponente verändert, so

		gibt dieses Feld an welcher Art die Veränderung ist.
AufbauprojektNeufunktion	int	Neu erstellter BFC eines Aufbauprojektes? 0 = false; 1 = true

Tabelle 44: Datenbanktabelle Basisfunktionskomponente (BFC)

Tabelle: Typ

Beschreibung: Die Tabelle enthält die vier verschiedenen Typen, welche eine Basisfunktionskomponente annehmen kann (E, W, R, X).

Abbildung als Objekt: Alle Daten der Tabelle 'Typ' werden in Radiobuttons der Klasse 'Mainwindow' geladen.

Spaltenname	Datentyp	NULL zulassen
TypID	int	<input type="checkbox"/>
TypName	varchar(50)	<input type="checkbox"/>
Abkürzung	varchar(1)	<input type="checkbox"/>
Beschreibung	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Abbildung 45: Datenbanktabelle Typ

Feldname	Datentyp	Beschreibung
TypID	int	Primärschlüssel der Tabelle.
TypName	varchar(50)	Name des Typs.
Abkürzung	varchar(1)	Abkürzung des Typs.
Beschreibung	varchar(50)	Beschreibung des Typs.

Tabelle 45: Datenbanktabelle Typ

Beziehung: FK_FUB_SIBLING

Beschreibung: Dient zur Organisation der Blätter eines Baumes in einer Ebene mit dem gleichen Vater.

Primärschlüsseltabelle: FUB

Feld: FUBID

Fremdschlüsseltabelle: FUB

Feld: ParentSiblingFK

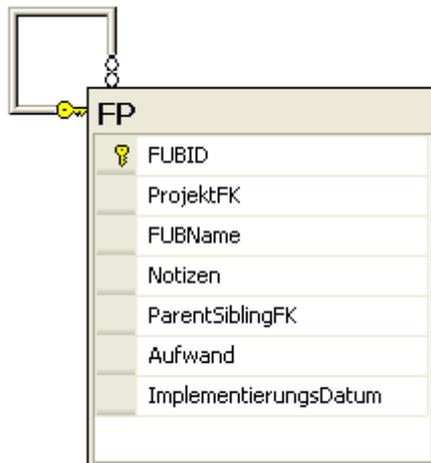


Abbildung 46: Beziehung FP - FP

Beziehung: FK_Personenstunden_FUB

Beschreibung: Ordnet einem Funktionalprozess Personenstundeneinträgen zu. Ein Funktionalprozess kann mehrere Personenstundeneinträge haben, ein Personenstundeneintrag aber immer nur zu einem Funktionalprozess gehören (1 zu n).

Primärschlüsseltabelle: FUB

Feld: FUBID

Fremdschlüsseltabelle: Personenstunden

Feld: FUBFK

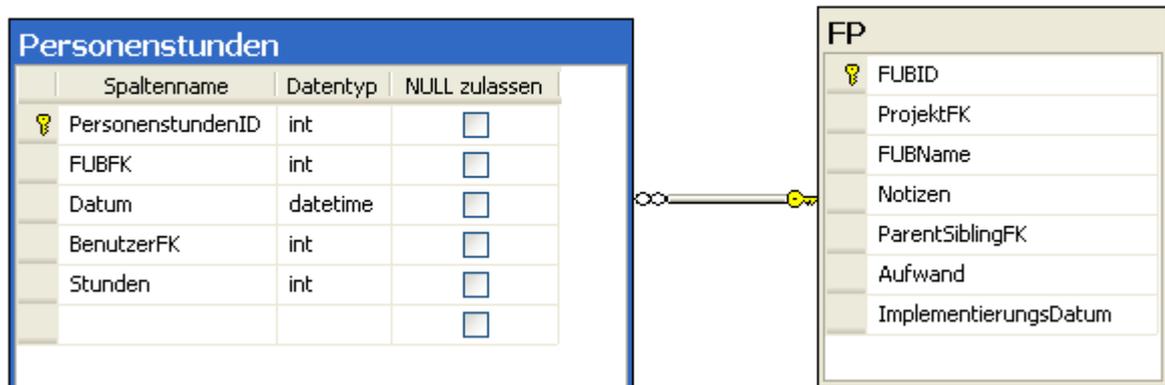


Abbildung 47: Beziehung Personenstunden - FP

INSERT- und UPDATE-Spezifikation

Beim Löschen eines FP werden alle zugehörigen Personenstunden von der Datenbank gelöscht.

Beziehung: FK_Projekt_AufbauProjekt

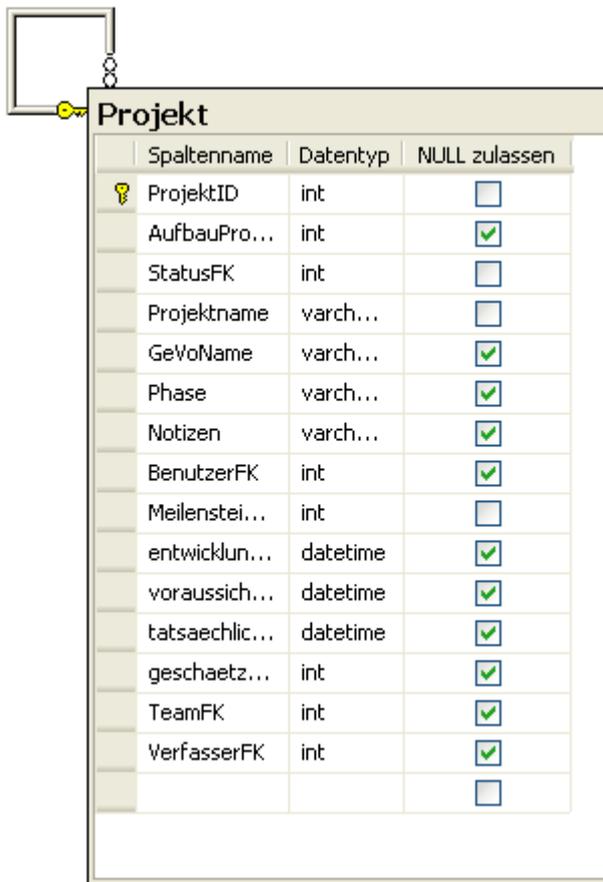
Beschreibung: Die zwei Arten von Projekten werden über das Feld AufbauProjektFK definiert. Referenziert ein Projekt ein anderes, so handelt es sich um ein Weiterentwicklungsprojekt.

Primärschlüsseltabelle: Projekt

Feld: ProjektID

Fremdschlüsseltabelle: Projekt

Feld: AufbauProjektFK



The image shows a screenshot of a database table definition for a table named 'Projekt'. The table has the following columns and properties:

Spaltenname	Datentyp	NULL zulassen
ProjektID	int	<input type="checkbox"/>
AufbauPro...	int	<input checked="" type="checkbox"/>
StatusFK	int	<input type="checkbox"/>
Projektname	varch...	<input type="checkbox"/>
GeVoName	varch...	<input checked="" type="checkbox"/>
Phase	varch...	<input checked="" type="checkbox"/>
Notizen	varch...	<input checked="" type="checkbox"/>
BenutzerFK	int	<input checked="" type="checkbox"/>
Meilenstei...	int	<input type="checkbox"/>
entwicklun...	datetime	<input checked="" type="checkbox"/>
voraussich...	datetime	<input checked="" type="checkbox"/>
tatsaechlic...	datetime	<input checked="" type="checkbox"/>
geschaetz...	int	<input checked="" type="checkbox"/>
TeamFK	int	<input checked="" type="checkbox"/>
VerfasserFK	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Abbildung 48: Beziehung Projekt - Projekt

Beziehung: FK_Projekt_Benutzer

Beschreibung: Ordnet einem Projekt einen Benutzer zu, welches dieser gerade bearbeitet. Ein Projekt kann nur von einer Person bearbeitet werden. Beendet diese Person die Arbeit am Projekt, so wird die Referenz gelöscht.

Primärschlüsseltabelle: Benutzer

Feld: BenutzerID

Fremdschlüsseltabelle: Projekt

Feld: BenutzerFK

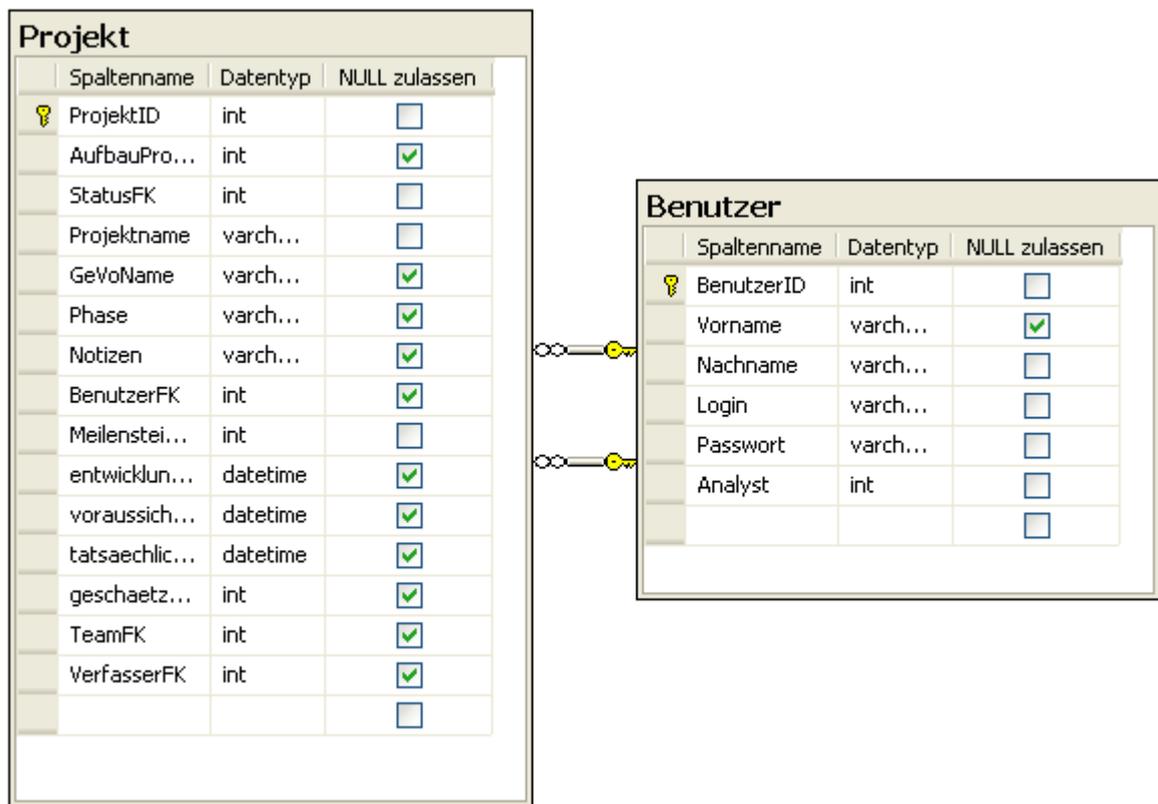


Abbildung 49: Beziehung Projekt - Benutzer

Beziehung: FK_Projekt_Team

Beschreibung: Ordnet ein Team einem Projekt zu. Ein Projekt kann nur zu einem Team gehören, ein Team aber zu mehreren BFC.

- Primärschlüsseltabelle: Team
- Feld: TeamID
- Fremdschlüsseltabelle: Projekt
- Feld: TeamFK

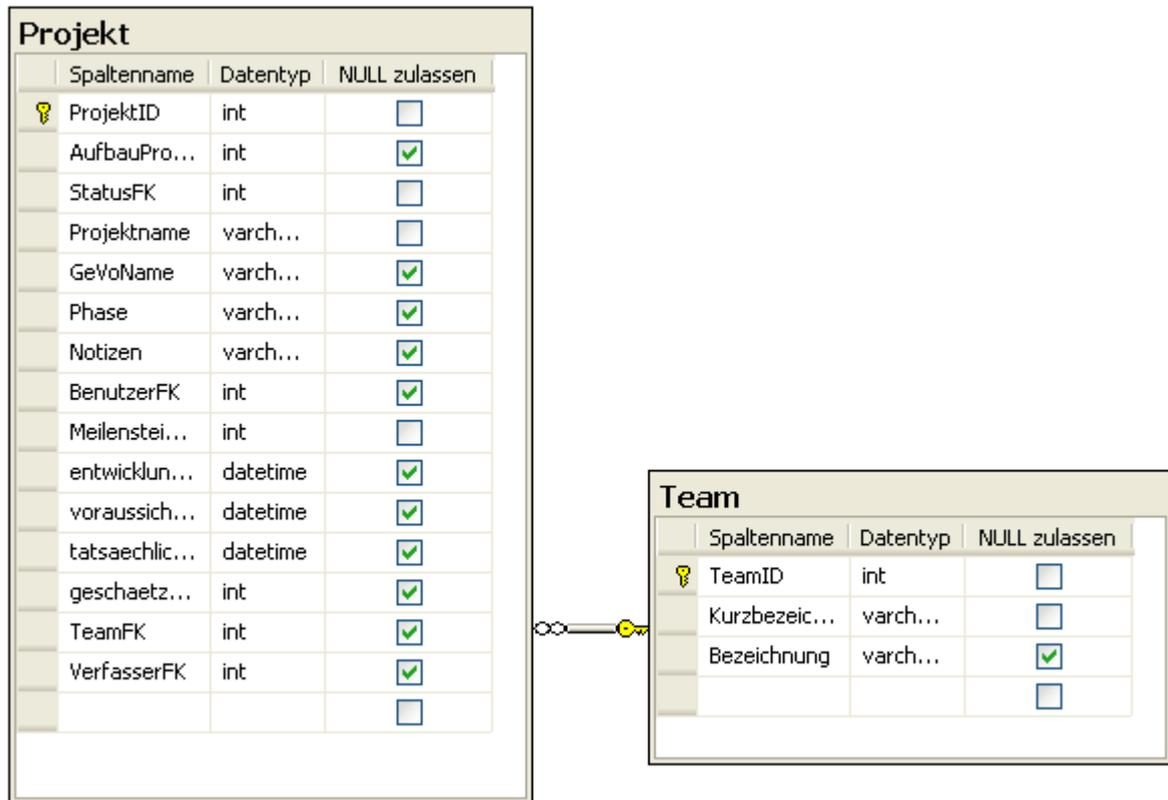


Abbildung 50: Beziehung Projekt - Team

Beziehung: FK_Projekt_Meilenstein

Beschreibung: Ordnet einen Meilenstein einem Projekt zu. Ein Projekt kann nur zu einem Meilenstein gehören, ein Meilenstein aber zu mehreren Projekten.

Primärschlüsseltabelle: Meilenstein

Feld: MeilensteinID

Fremdschlüsseltabelle: Projekt

Feld: MeilensteinFK

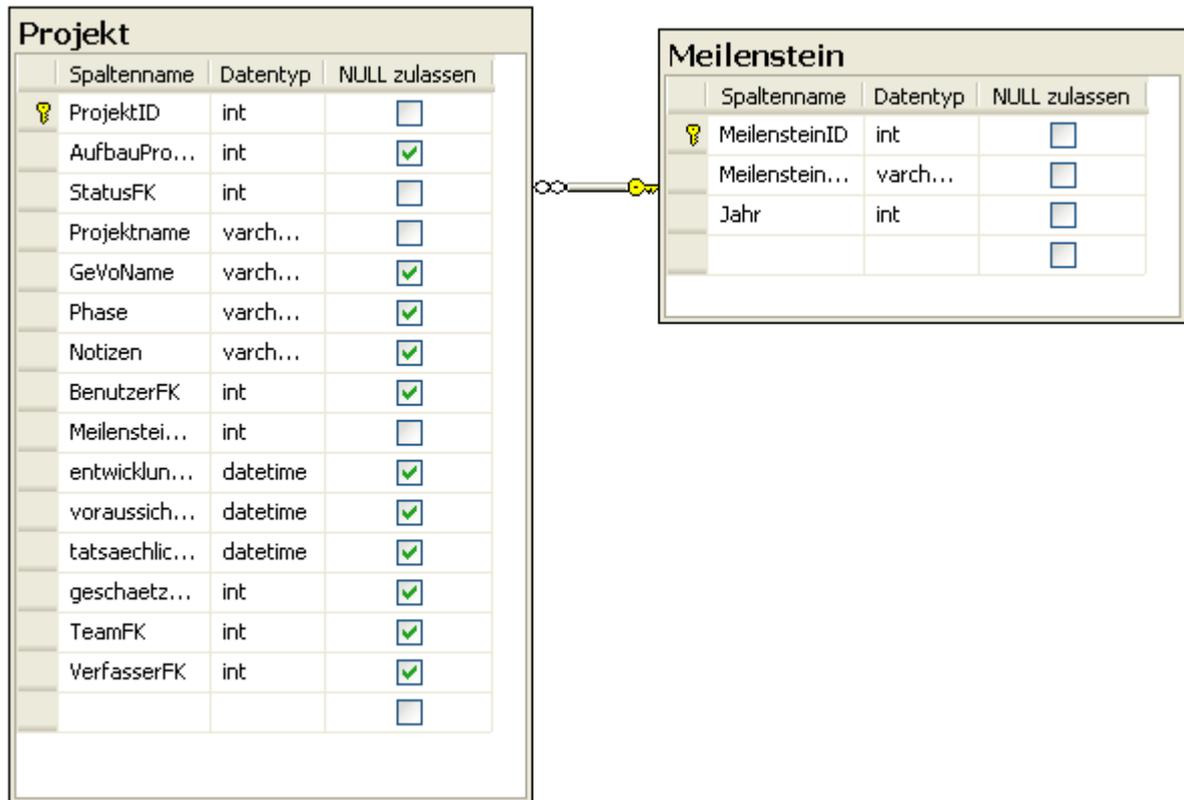


Abbildung 51: Beziehung Projekt – Meilenstein

Beziehung: FK_Projekt_Status

Beschreibung: Ordnet einen Status einem Projekt zu. Ein Projekt kann nur zu einem Status gehören, ein Status aber zu mehreren Projekten.

Primärschlüsseltabelle: Status
 Feld: StatusID
 Fremdschlüsseltabelle: Projekt
 Feld: StatusFK

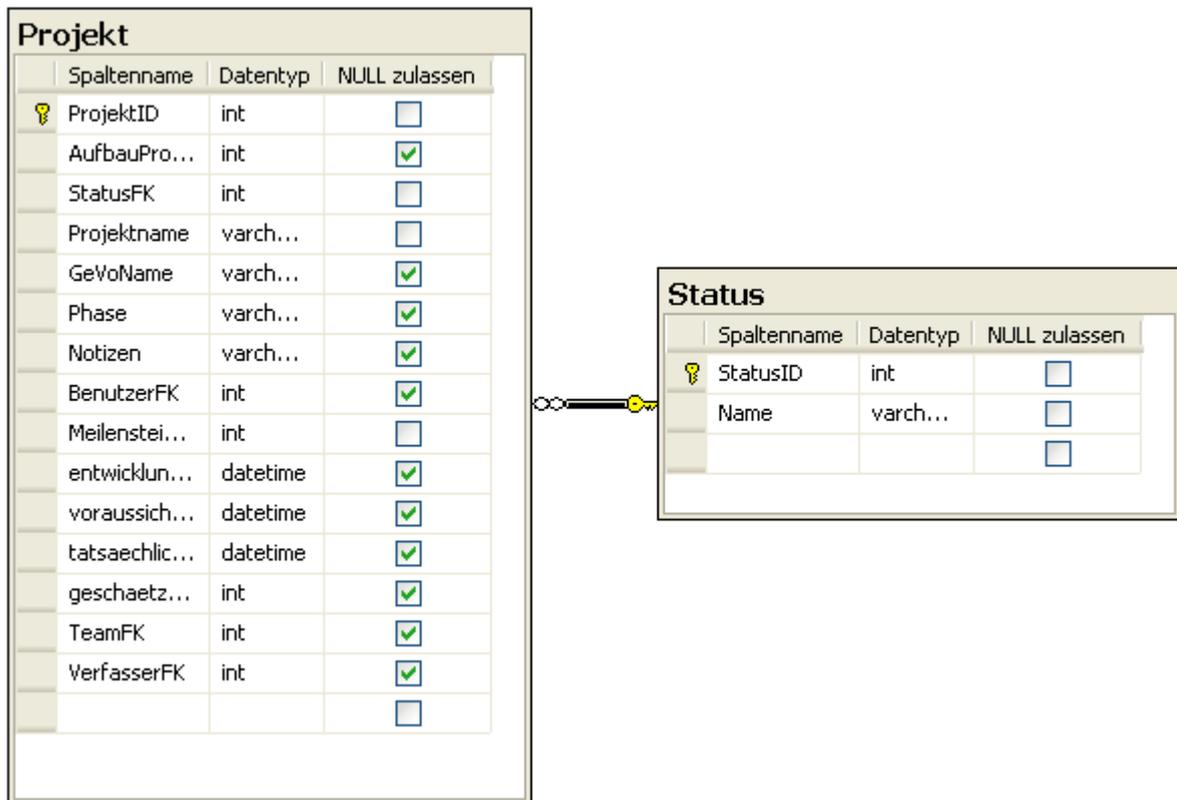


Abbildung 52: Beziehung Projekt - Status

Beziehung: FK_BFC_Veraenderungstyp

Beschreibung: Ordnet einen Veränderungstyp einer BFC zu. Eine BFC kann nur zu einem Veränderungstyp gehören, ein Veränderungstyp aber zu mehreren Projekten.

Primärschlüsseltabelle: Veraenderungstyp

Feld: VeraenderungstypID

Fremdschlüsseltabelle: BFC

Feld: VeraenderungstypFK

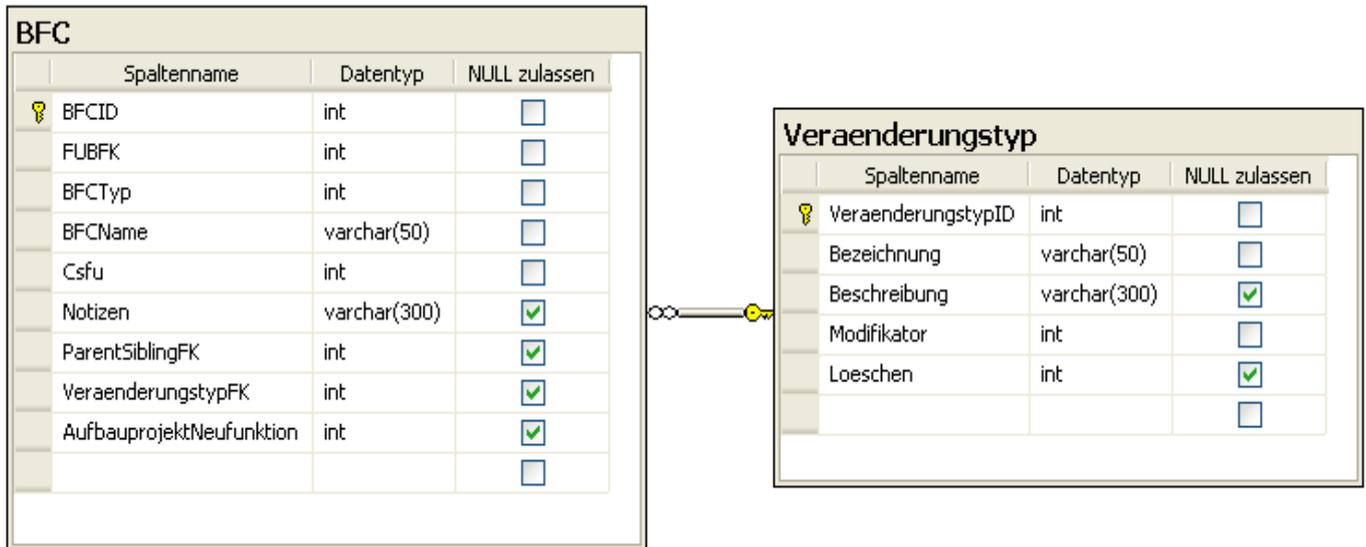


Abbildung 53: Beziehung BFC - Veraenderungstyp

Beziehung: FK_BFC_Typ

Beschreibung: Ordnet einer BFC einen Typ zu. Eine BFC kann nur zu einem Typ gehören, ein Typ aber zu mehreren BFC.

Primärschlüsseltabelle: Typ
 Feld: TypID
 Fremdschlüsseltabelle: BFC
 Feld: BFCTyp

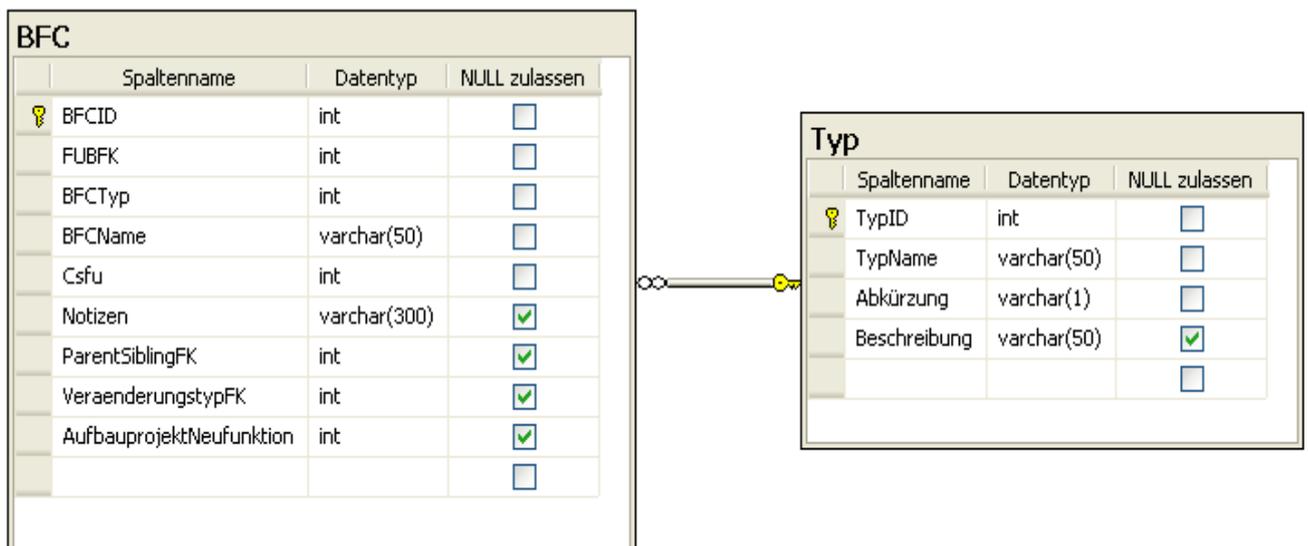


Abbildung 54: Beziehung BFC - Typ

Beziehung: FK_FUB_Projekt

Beschreibung: Ordnet ein Funktionalprozess einem Projekt zu. Ein Projekt kann mehrere Funktionalprozesse haben, ein Funktionalprozess aber immer nur zu einem Projekt gehören (1 zu n).

Primärschlüsseltabelle: Projekt

Feld: ProjektID

Fremdschlüsseltabelle: FP

Feld: ProjektFK

INSERT- und UPDATE-Spezifikation

Beim Löschen eines Projektes werden alle zugehörigen Funktionalprozesse von der Datenbank gelöscht.

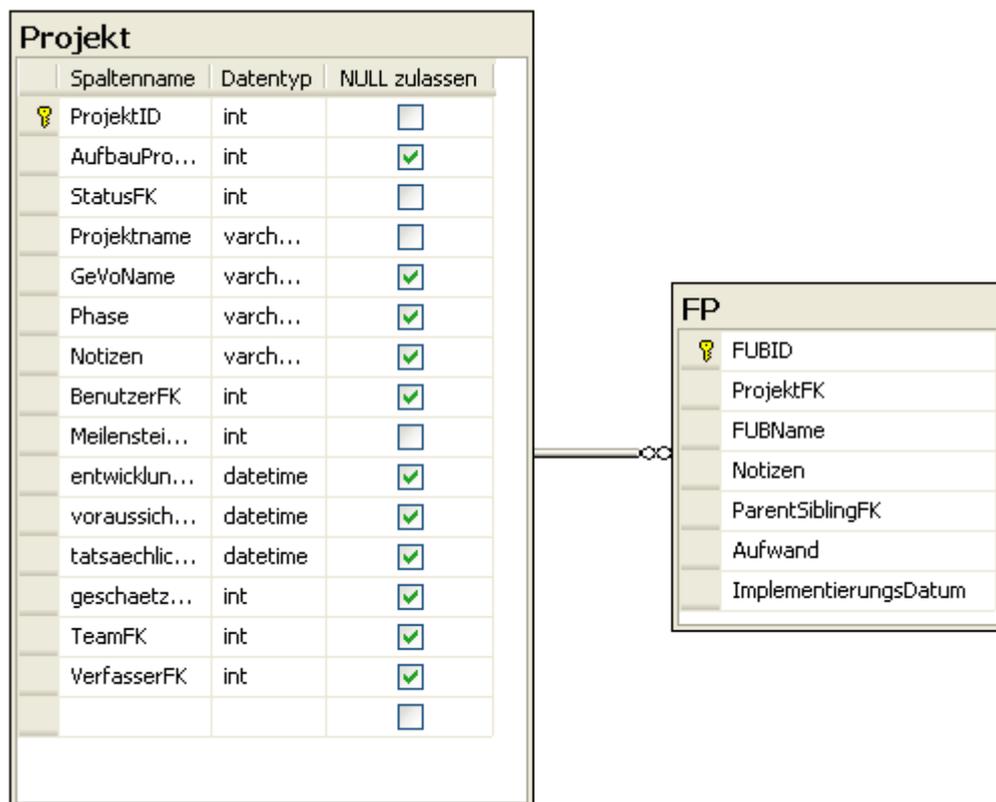


Abbildung 55: Beziehung Projekt - FP

Beziehung: FK_BFC_FUB

Beschreibung: Ordnet eine Basisfunktionskomponente einem Funktionalprozess zu. Ein Funktionalprozess kann mehrere Basisfunktionskomponenten haben, eine Basisfunktionskomponente aber immer nur zu einem Funktionalprozess gehören (1 zu n).

Primärschlüsseltabelle: FUB

Feld: FUBID

Fremdschlüsseltabelle: BFC

Feld: FUBFK

INSERT- und UPDATE-Spezifikation

Beim Löschen eines Funktionalprozesses werden alle zugehörigen Basisfunktionskomponenten von der Datenbank gelöscht.

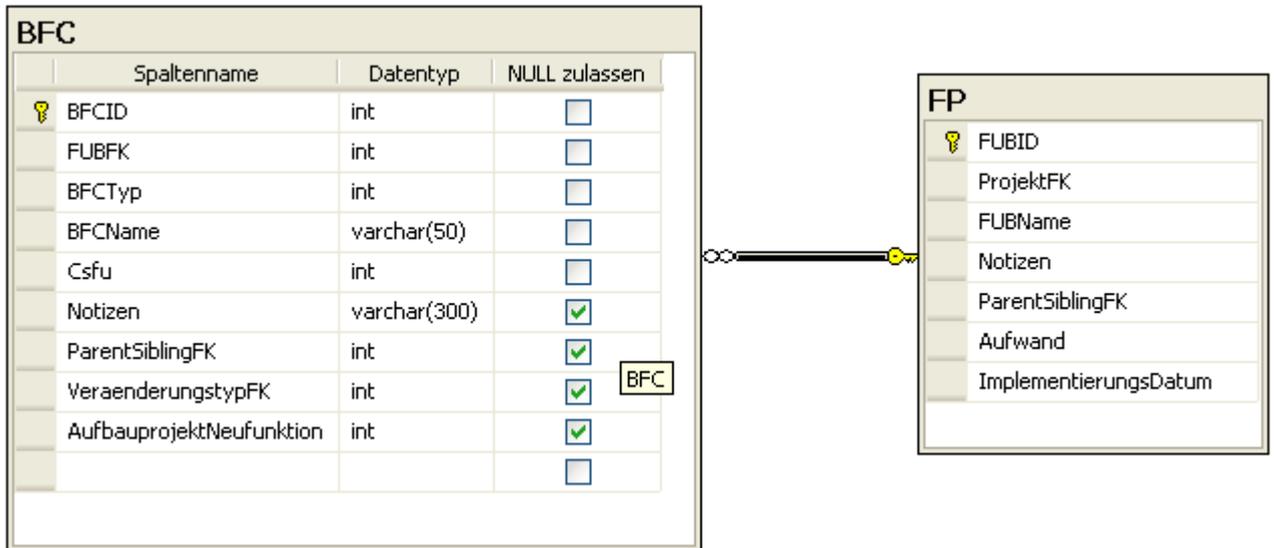


Abbildung 56: Beziehung BFC - FP

Beziehung: FK_BFC_Sibling

Beschreibung: Dient zur Organisation der Blätter einer Ebene mit dem gleichen Vater.

Primärschlüsseltabelle: BFC

Feld: BFCID

Fremdschlüsseltabelle: BFC

Feld: ParentSiblingFK

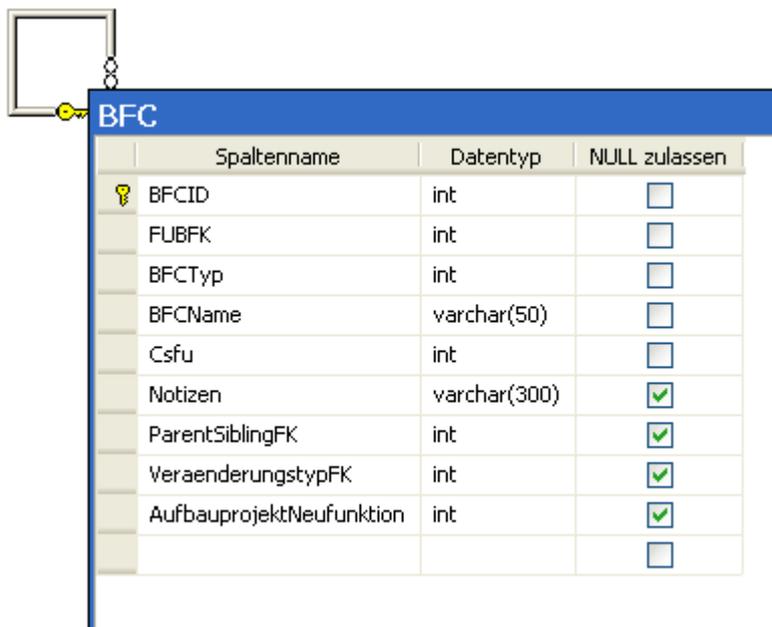
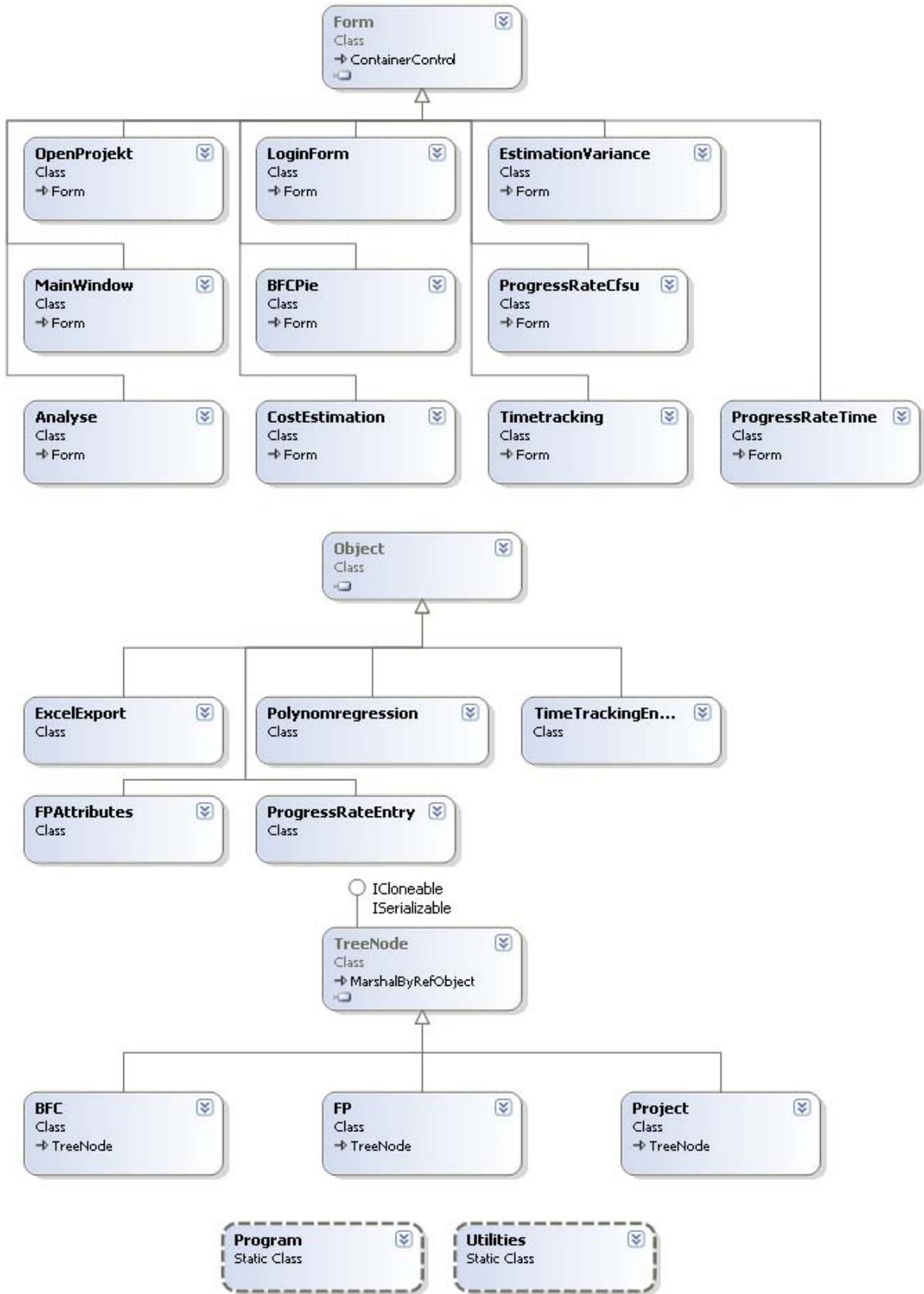


Abbildung 57: Beziehung BFC - BFC

Klassendiagramm



Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 14. November 2007
Ort, Datum

Unterschrift