

Institut für Rechnergestützte Ingenieursysteme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Diplomarbeit Nr. 3719

Aufwandsschätzung bei Geschäftsprozessmodellerstellung

Aleksandar Milutinovic

Studiengang:	Informatik
Prüfer/in:	Univ-Prof. Hon-Prof. Dieter Roller
Betreuer/in:	Dipl.-Inf. Felix Baumann
Beginn am:	23. März 2015
Beendet am:	02. Oktober 2015
CR-Nummer:	D.2.8, H.4.1, J.1

Kurzfassung

Geschäftsprozessmanagement-Projekte im Allgemeinen und Modellierungsprojekte im Speziellen besitzen noch keine angewandte und wissenschaftlich gestützte und untersuchte Methode zur Aufwandsschätzung. Insbesondere gibt es für die Phase der Modellerstellung innerhalb des Geschäftsprozessmanagement-Lebenszyklus keine Aufwandsschätzungen, während der Bereich der Implementierung von Geschäftsprozessmodellen erste Ansätze zu Aufwandschätzungen in der Literatur aufzeigt.

Diese Arbeit gibt einen Überblick über Schätzmethoden, den aktuellen Stand der Literatur aus angrenzenden und entfernten Forschungsfeldern und entwickelt basierend darauf eine Methode namens BPM COCOMO zur Aufwandsschätzung bei der Erstellung von Geschäftsprozessmodellen. Es wird eine Adaption von COCOMO vorgenommen und eine weitere Untersuchung und Validierung dieses Modells vorgeschlagen.

Keywords: geschäftsprozessmodellierung, aufwandsschätzung, gpm, bpm, bpmn, function point, cocomo, modellierung, aufwand, kosten

Abstract

Up to this point business process management projects in general and business process modeling projects in particular could not rely on a practical and scientifically validated method to estimate cost and effort.

Especially for the process step of model development, so far no cost and effort estimate approach has been developed within the business process management life cycle. However, regarding the implementation of business process models, initial solutions are discussed in the literature.

The thesis at hand provides an overview on the existing methods to estimate cost and effort and the state of the literature published to date from closely related fields of research as well as from less connected areas. Based on that, a method is being developed to estimate cost and effort for the development of business process models: BPM COCOMO. After an adaptation of the method, different ideas for further analysis and validation of the method are proposed.

Keywords: Business process modelling, cost estimation, effort estimation, gpm, bpm, bpmn, function point, cocomo, modelling, effort, cost

Inhaltsverzeichnis

1	Einleitung	9
1.1	Motivation	10
1.2	Business Process Management (BPM)	10
1.2.1	Definition	11
1.2.2	BPM-Lifecycle	12
1.3	Aufgabenstellung der Arbeit	13
1.4	Modellierungssprachen	15
1.4.1	WS-BPEL	16
1.4.2	Business Process Model and Notation (BPMN) 2.0	17
1.4.3	CMMN und DMN	18
2	Forschungsstand	21
2.1	Aktueller Forschungsstand	21
2.2	Entfernte Forschungsfelder	22
2.3	Angrenzende Forschungsfelder	24
2.4	Übertragbarkeit der Ergebnisse	26
3	Voraussetzungen zur Schätzung	27
3.1	Einsatz von BPM in Unternehmen	27
3.2	Prozessdokumentation	29
3.2.1	Vorhandene Prozessdokumentation	30
3.2.2	Prozessdokumentation erstellen	31
4	Überblick Schätzmethoden	35
4.1	Schätzmethoden aus der Softwareentwicklung	35
4.1.1	Function Point (FP)	37
4.1.2	Use Case Points (UCP)	39
4.1.3	Constructive Cost Model (COCOMO)	43
4.2	Auswahl der Faktoren zur Schätzung	47
4.3	Weitere Faktoren und Einflussgrößen	51
5	Einführung BPM Schätzmethode	53
5.1	Auswahl der Methode	53
5.2	BPM Function-Point	54
5.3	BPM COCOMO	57
5.3.1	Bestimmen der Eingangsgrößen	57
5.3.2	Ermitteln der Lines of Code (LOC)	59
5.3.3	Selektion der Faktoren	61

5.3.4	Adaption der Faktoren	66
5.3.5	Zusätzliche Faktoren	73
5.3.6	Gesamtüberblick Adaption	75
5.3.7	Gesamtüberblick Berechnung	75
5.4	Web-basiertes Berechnungstool	79
5.5	Vorschlag zur Evaluation der Methode	81
6	Zusammenfassung und Ausblick	83
6.1	Fazit Forschungsstand	83
6.2	Fazit Schätzverfahren	84
6.3	Ausblick	85
	Literatur	89

Abbildungsverzeichnis

1.1	BPM-Lifecycle adaptiert aus [Weske 2012]	13
1.2	Darstellung der Prozesslandschaft aus [Freund und Rücker 2012]	14
1.3	BPM-Lifecycle - Fokussierung auf Phasen adaptiert aus [Weske 2012]	15
1.4	Kurzübersicht Elemente BPMN 2.0 [Weske 2012]	18
1.5	Beispielmodell DMN - eigene Darstellung	20
2.1	PDCA-Zyklus adaptiert aus [<i>PDCA Cycle</i>]	24
3.1	camundaBPM Kreislauf – Quelle: [Freund und Rücker 2012]	30
3.2	Streuung der Schätzung in Abhängigkeit von Projektphase – Quelle: [Frohnhoff 2009]	32
4.1	Übersicht Schätzmethoden für SE, Quelle: [Khatibi und Jawawi 2011]	36
4.2	Übersicht Skalenfaktoren COCOMO II – Quelle: [Frohnhoff 2009]	45
4.3	Übersicht Kostentreiber COCOMO II – Quelle: [Frohnhoff 2009]	46
4.4	Beispielprozess Abendessen – Symmetrische Modellierung mit Gateways – Quelle: [Freund und Rücker 2012]	48
4.5	Beispielprozess Abendessen – Modellierung mit mehr Gateways – Quelle: [Freund und Rücker 2012]	49
5.1	Ablauf der der Schätzung mit BPM COCOMO	57
5.2	Übersicht der Komplexitätsmetriken für BPM adaptiert aus [Gruhn und Laue 2006])	59
5.3	Adaptionstabelle BPM COCOMO mit Eingangsgrößen, Skalenfaktoren und Kostentreibern	76
5.4	Übersicht Skalenfaktoren BPM COCOMO (Adaption aus [Frohnhoff 2009])	77
5.5	Übersicht Kostentreiber BPM COCOMO (Adaption aus [Frohnhoff 2009])	78
5.6	Screenshot BPM COCOMO Tool - eigene Anfertigung	80

Tabellenverzeichnis

1.1	Vergleich zwischen BPMN, CMMN und DMN, Quelle: [Kemsley 2015]	19
-----	---	----

2.1	SE Lebenszyklenphasen, Quelle: [Khatibi und Jawawi 2011]	25
2.2	Mapping von BPM und SE Lebenszyklenphasen, Quelle: [Baklizky u. a. 2013]	26
4.1	Function Point Kategorien adaptiert aus [Khatibi und Jawawi 2011]	38
4.2	TCF in Function Point – Übersetzt aus [Frohnhoff 2009; Khatibi und Jawawi 2011]	39
4.3	Gewichtung Use Cases - deutsche Übersetzung aus [Frohnhoff 2009]	40
4.4	Zuordnung Actor Gewichte - adaptiert aus [Frohnhoff 2009]	41
4.5	Einflussgrößen des TCF übersetzt aus [Frohnhoff 2009]	42
4.6	Einflussgrößen des EF – übersetzt aus [Frohnhoff 2009]	42
4.7	Übersicht Berechnungsfaktoren und Projektarten [Barry W. Boehm 1981; Frohnhoff 2009]	43
5.1	Vergleichstabelle Function-Point und BPM	55
5.2	Beschreibung der FP Kategorien für BPM	55
5.3	Einflussfaktoren FP adaptiert aus [Wieczorrek und Mertens 2007]	56
5.4	Adaptionstabelle UFP für BPM	59
5.5	Ermittlung von Codezeilen für BPMN Objekte	60
5.6	Ermittlung von Codezeilen für BPEL Objekte	61
5.7	Umrechnungstabelle BMP zu LOC	61
5.8	Übersicht Komplexitätsmetriken. Adaptiert aus [Gruhn und Laue 2006]	70
5.9	Skalentreiber „Levels of Process Documentation“ (Adaption aus [Frohnhoff 2009])	74
5.10	Kostentreiber "Process Confidentiality" (Adaption aus [Frohnhoff 2009])	75

Verzeichnis der Listings

1.1	BPEL Beispielprozess aus [BPEL 2.0 Specification]	17
5.1	BPMN XML-Beschreibung für das Starterereignis [BPMN 2.0 Specification]	60
5.2	BPMN XML-Beschreibung für eine Aktivität [BPMN 2.0 Specification]	61

Kapitel 1

Einleitung

Diese Arbeit beschäftigt sich mit dem Entwurf eines angepassten Schätzmodells für die Geschäftsprozessmodellerstellung. Die Vorgehensweise hierbei ist zunächst den aktuellen Forschungsstand zu untersuchen und basierend auf den Untersuchungsergebnissen unter Einbeziehung von entfernten und angrenzenden Forschungsfeldern ein Schätzmodell zu entwickeln. Abschließend werden Vorschläge zur Evaluierung der Schätzmethode postuliert.

Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 1 – Einleitung: In diesem Kapitel wird eine Einführung in das Thema Geschäftsprozessmanagement gegeben, die zum besseren Verständnis von Geschäftsprozessmodellen dienen soll. Ebenfalls werden Nutzen und Ziele der Prozessausrichtung im Unternehmen und der Modellierung selbst sowie ausgewählte Modellierungssprachen erläutert.

Kapitel 2 – Forschungsstand: Hier wird der aktuelle Forschungsstand zu Schätzverfahren für die Modellerstellung beschrieben. Dabei werden angrenzende und entfernte Forschungsfelder betrachtet und hinsichtlich ihrer Anwendbarkeit auf das Thema hin untersucht.

Kapitel 3 – Voraussetzungen zur Schätzung: Dieses Kapitel beschreibt die Voraussetzungen für die Anfertigung einer Schätzung zur Geschäftsprozessmodellerstellung.

Kapitel 4 – Überblick Schätzmethoden: Aus den untersuchten Forschungsfeldern werden in diesem Kapitel ausgewählte Schätzmethoden detaillierter beschrieben und eine Übertragbarkeit sowie eine Adaption auf die Fragestellung hergeleitet.

Kapitel 5 – Einführung BPM Schätzmethode: Aus den bisherigen Erkenntnissen wird hier eine neue Schätzmethode entwickelt und detailliert beschrieben.

Kapitel 6 – Zusammenfassung und Ausblick: Dieses Kapitel fasst die Ergebnisse der Arbeit zusammen und stellt Anknüpfungspunkte vor. Insbesondere wird hier auf die Evaluation der Schätzmethode eingegangen.

1.1 Motivation

Die Erstellung und Bearbeitung von Geschäftsprozessen erfolgt in der Industrie teilweise unsystematisch. Durch sich dynamisch verändernde Märkte und den damit einhergehenden Wandel in Unternehmen, ist Geschäftsprozessmanagement zu einem sehr wichtigen Thema in Unternehmen geworden [Freund und Rücker 2012; Horan 2011; Müller 2014; Posluschny 2012].

Die Modellierung dieser Prozesse und die damit einhergehende informationstechnologische (IT) Unterstützung bei der Ausführung, Überwachung und insbesondere bei der Verbesserung jener ist bereits stark verbreitet [Freund und Rücker 2012; Müller 2014].

Trotz der Existenz von Metriken, welche die Qualität und Fehlerhaftigkeit solcher Modelle bewerten, halten diese Qualitätsattribute nur langsam Einzug in die entsprechenden Werkzeuge und in das Handeln der Anwender. Eng verwandt mit diesen Metriken ist die Frage nach Aufwands- und Kostenschätzung zur Erstellung, Wartung und Anpassung von Geschäftsprozessmodellen [Gruhn und Laue 2006; Kluza und Nalepa 2012].

Im Gegensatz zu anderen verwandten Feldern, wie beispielsweise der Softwareentwicklung, fehlt im Bereich des Business Process Management (BPM) eine akzeptierte Vorgehensweise zur Schätzung des Aufwands für die Modellerstellung von Geschäftsprozessen. Da allerdings Geschäftsprozesse ein erfolgskritischer Faktor für Unternehmen sind und die Produktqualität aus der Prozessqualität resultiert - „Process equals Product“ [Leymann 2011], ist deren Modellierung und Dokumentation ein unverzichtbarer Bestandteil der Unternehmensführung.

Wie bereits erwähnt, soll diese Arbeit den aktuellen Forschungsstand eingehend betrachten und mit Hilfe dieser Untersuchung einen Ansatz finden, den Aufwand für die Modellierung abzuschätzen.

1.2 Business Process Management (BPM)

Business Process Management (BPM), auf Deutsch als Geschäftsprozessmanagement (GPM) übersetzt, ist ein seit Jahren aktuelles und oft untersuchtes Thema¹. BPM findet immer mehr Einzug in die Forschung und in Organisationen, die es mittlerweile als eine Notwendigkeit und etablierte Aufgabe betrachten [Gadatsch 2013].

Dieser Abschnitt vermittelt die Grundlagen zu BPM und den damit verbundenen Begrifflichkeiten.

¹<http://www.kurze-prozesse.de/2014/02/26/bpm-quintessenz-wertet-35-studien-aus/>

1.2.1 Definition

„Ein Unternehmen funktioniert besser mit BPM“ [Freund und Rücker 2012].

Unternehmen müssen in der heutigen, schnelllebigen Zeit besonders schnell und flexibel auf Umwelteinflüsse reagieren können. Um dabei die Wirtschaftlichkeit zu erhalten, müssen Abläufe, und damit das Unternehmen selbst, schnell und effizient sein. Dabei ist Kostendruck ist der stärkste Treiber, um Abläufe und Prozesse weiter zu verschlanken und zu optimieren. Ebenfalls zwingt Kostendruck das Unternehmen weitere Einsparungen zu erzielen und die Effizienz zu erhöhen [Schmelzer und Sesselmann 2008]. Hierbei wird BPM als Mittel zur prozessorientierten Unternehmensgestaltung eingesetzt. BPM befasst sich mit dem Dokumentieren, Gestalten und Verbessern von Geschäftsprozessen und deren IT-technischer Unterstützung [Müller 2014].

Ein Geschäftsprozess ist die Abfolge von zusammenhängenden Arbeitsschritten, die mit einem bestimmten Input eine bestimmte Leistung erbringen [Posluschny 2012]. Geschäftsprozesse beinhalten die Kenntnis über Kundengruppen, Anwendungen, Erfolgsfaktoren, Stärken & Schwächen, Wettbewerbsstrategie und Unternehmensziele. Mit Hilfe dieser Kenntnisse ist es möglich, die Wertschöpfung signifikant zu erhöhen. [Posluschny 2012]. Ein Geschäftsprozess beginnt und endet beim Kunden [Gadatsch 2013; Schmelzer und Sesselmann 2008].

Nicht nur Kunden stellen Anforderungen an Geschäftsprozessmanagement, sondern auch die Ebene der Geschäftsführung, die Leistungsparameter (KPIs) erfragt, um die eigene Effizienz und Effektivität zu bewerten und daraus weitere Optimierungsnotwendigkeiten etc. abzuleiten [Schmelzer und Sesselmann 2008].

BPM schlägt die Brücke zwischen Betriebswirtschaft und IT und hilft das Unternehmen mit IT-Unterstützung zu verbessern.

Der in dieser Arbeit verwendete Begriff der Organisation steht stellvertretend für Institutionen und Unternehmen. Dabei sind für diese Arbeit im Bereich der Unternehmen mittelständische Unternehmen (KMU) aufgrund ihrer Größe² relevant, da dort eine Vielzahl von BPM-Projekten durchgeführt werden [Freund und Rücker 2012].

²Mittlere Unternehmen bis 249 Beschäftigte, Jahresumsatz bis 50 Mill. EUR [Müller 2014].

1.2.2 BPM-Lifecycle

Der Ablauf von BPM wird durch den BPM-Lifecycle [Weske 2012] dargestellt. Dieser „Lebenszyklus“ verdeutlicht die Vorgehensweise von BPM und stellt dabei die jeweiligen Phasen dar. Die einzelnen Phasen des BPM-Lifecycle lauten:

- ◇ Analyse
- ◇ Modellierung
- ◇ Implementierung
- ◇ Ausführung
- ◇ Überwachung
- ◇ Optimierung

Der Zyklus beginnt mit der Analysephase (in anderer Literatur auch Designphase). In dieser Phase werden existierende Prozesse untersucht, sowie neue Prozesse zur Wertschöpfung identifiziert. Dabei werden KPIs analysiert und Anforderungen für die Beschreibung oder Verbesserung des Prozesses ermittelt. Hierbei entstehen Artefakte wie ein IST-Prozessmodell und ein SOLL-Prozessmodell. Die Bezeichnung des IST-Prozessmodells nach [Freund und Rücker 2012] irreführend, da diese nicht dem in der Modellierungsphase erstellten Modell entspricht. Vielmehr handelt es sich hierbei um eine Prozessdokumentation (die jedoch gleichzeitig auch das Prozessmodell sein kann, aber nicht sein muss).

In der Modellierungsphase wird das zu erstellende Prozessmodell selbst modelliert. Dies erfolgt in einer Modellierungssprache (vgl. Kapitel 1.4) und mit entsprechenden Werkzeugen. Dabei fließen die Erkenntnisse aus der Analysephase mit ein.

Die Implementierungsphase ist hier als zusätzliche Ebene eingeführt, um die Modellierung von der Implementierung abzutrennen, da dies nicht im selben Schritt erfolgt. Die Grenze zwischen den beiden Phasen ist je nach Einsatz von Modellierungssprache- und Werkzeug stark verschwommen [Freund und Rücker 2012]. Jedoch ist sie notwendig, um eine technische Verfeinerung des Prozessmodells vorzunehmen, damit dieses im nächsten Schritt auf einem Workflowmanagementsystem [Allweyer 2014] ausgeführt werden kann.

Da BPM einem kontinuierlichen Verbesserungsprozess unterliegt [Freund und Rücker 2012], erfolgt auch hier stets eine Überwachung des Prozesses, in die Daten aus der Ausführung einfließen. Diese Daten werden analysiert und dabei mögliche Probleme oder Schwachstellen identifiziert, die dann in der Optimierungsphase bearbeitet werden. Die daraus gewonnenen Erkenntnisse fließen wieder in die Analysephase ein, wo veränderte Anforderungen an das Prozessmodell ermittelt werden.

Dadurch ist der Kreislauf in sich geschlossen.

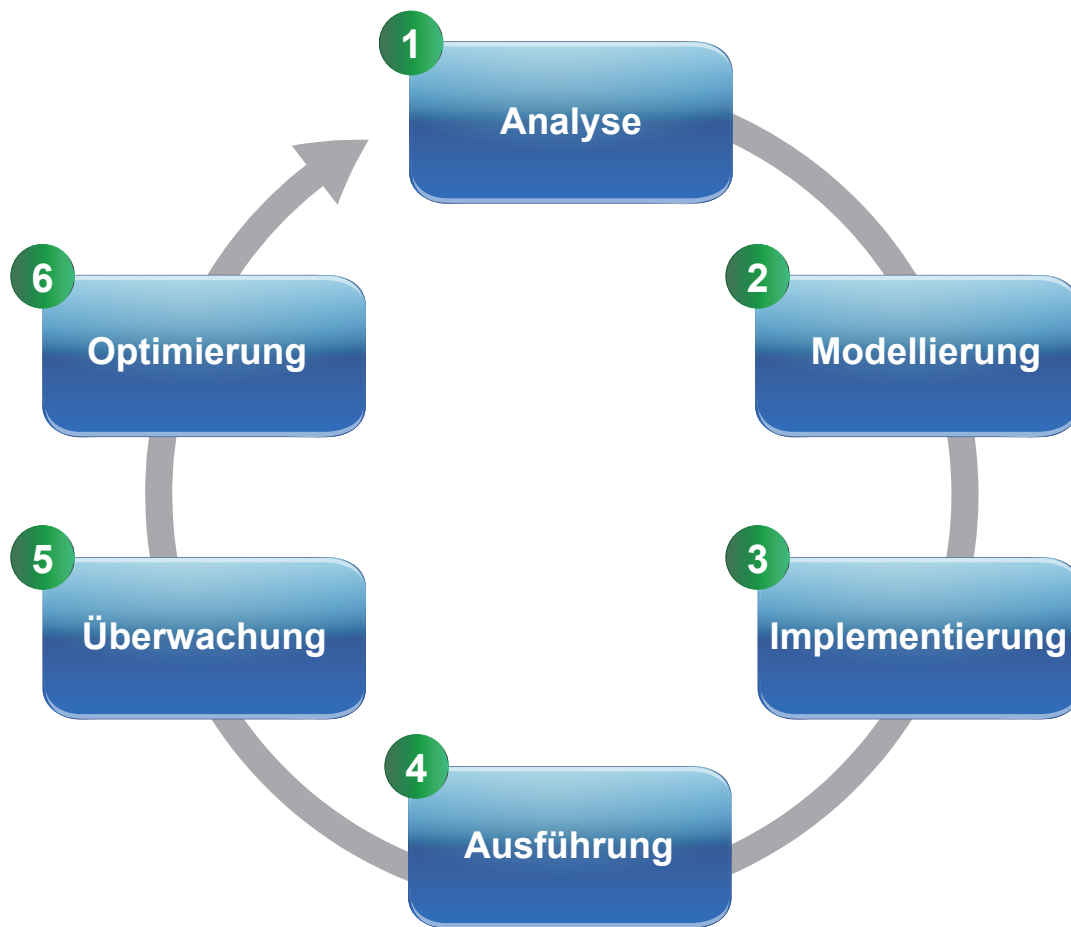


Abbildung 1.1: BPM-Lifecycle adaptiert aus [Weske 2012]

1.3 Aufgabenstellung der Arbeit

Ziel der vorliegenden Arbeit ist die Entwicklung eines Schätzmodells, welches den Aufwand zur Erstellung eines Geschäftsprozessmodells in einer graphischen Modellierungssprache abschätzt. Dabei soll auch zwischen vorhandener und fehlender (inkorrekt oder unvollständiger) Prozessdokumentation unterschieden werden. Schließlich wird ein Verfahren empfohlen, welches diese Methode validieren kann, da die erforderlichen Daten hierzu während der Bearbeitung der Arbeit nicht bereitgestellt wurden und somit keine eigene Validation durchgeführt werden konnte.

Wie Abbildung 1.2 aufzeigt lässt sich die „Prozesslandschaft“ in unterschiedliche Ebenen aufteilen. Zum einen lassen sich die vier Ebenen in der Mitte zwischen fachlicher und technischer Ebene aufteilen, wobei dort dann wieder verfeinert aufgeteilt wird. Für diese Arbeit sind die Ebene 1 – Strategisches Prozessmodell – und Ebene 2 – Operatives Prozessmodell – relevant.

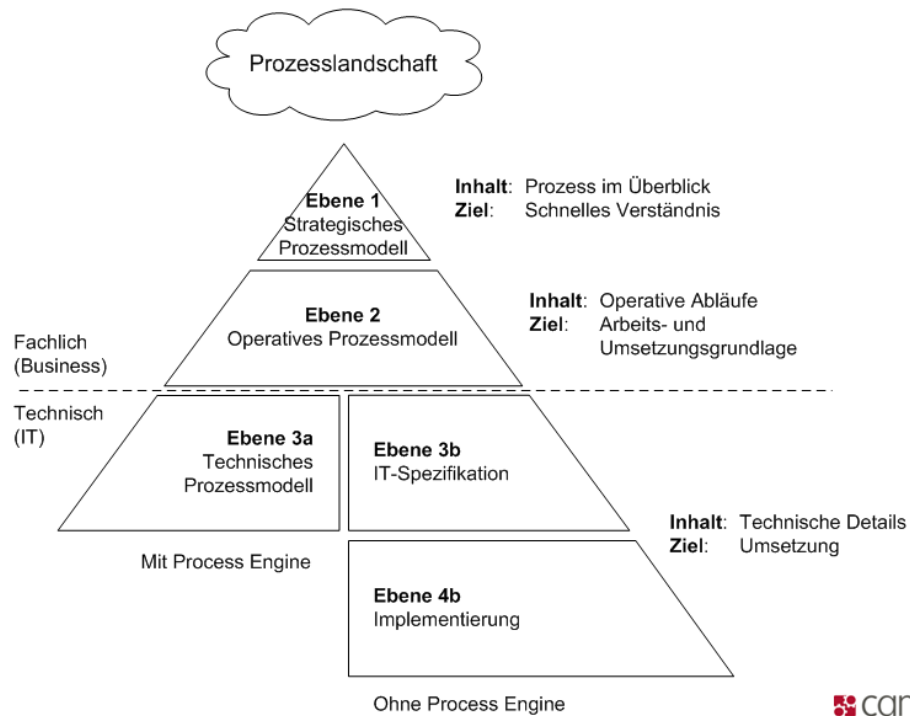


Abbildung 1.2: Darstellung der Prozesslandschaft aus [Freund und Rücker 2012]

In Abbildung 1.3 werden die betrachteten Phasen innerhalb des BPM-Lifecycle hervorgehoben und verdeutlicht.

Diese Arbeit soll kein Modell zur Schätzung eines Implementierungsaufwandes eines BPM Prozessmodells liefern. Dazu sind bereits weiterführende Ansätze in der Literatur vorhanden (vgl. Kapitel 2). Ebenfalls wird nur ein Ausblick auf Kosten für die Operation und Wartung gegeben.

Was die Arbeit nicht bieten kann, ist eine Anleitung zum Management von BPM-Projekten. Notwendige Arbeitsanweisungen und -abläufe werden jedoch im Hinblick auf deren Notwendigkeit zur Erstellung einer Schätzung des Aufwands für die Modellierung thematisch angeschnitten.

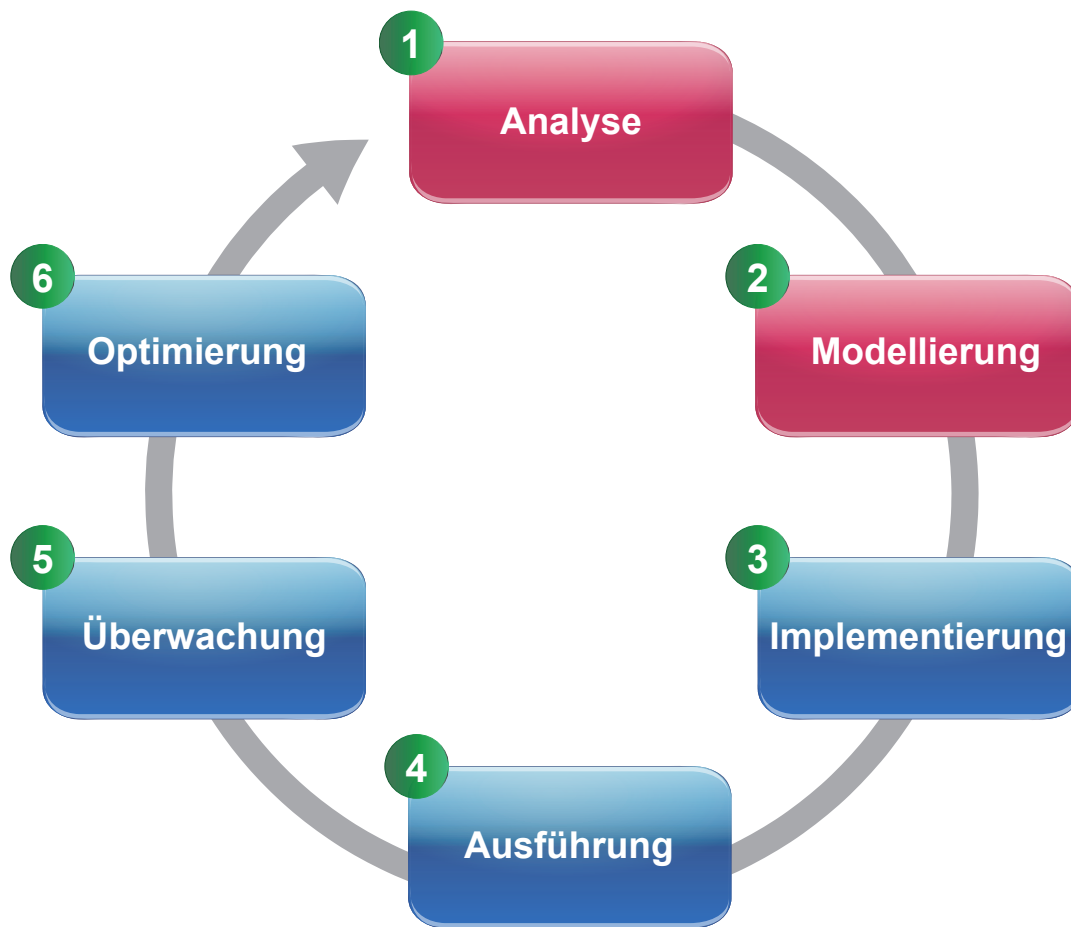


Abbildung 1.3: BPM-Lifecycle - Fokussierung auf Phasen adaptiert aus [Weske 2012]

1.4 Modellierungssprachen

Der Modellierer kann auf eine Vielzahl von Sprachen zurückgreifen, um das Prozessmodell zu beschreiben. So sind neben der Ereignisgesteuerten Prozesskette (EPK) auch Business Process Model And Notation 2.0 (BPMN)³ und Web Services Business Process and Execution Language (WS-BPEL)⁴ weit verbreitet [Müller 2014]. Aus Gründen der Übersichtlichkeit ist mit der Abkürzung BPMN die Version 2.0 gemeint. WS-BPEL wird ebenfalls zu BPEL abgekürzt. Es wird bewusst auf die Beschreibung der EPK verzichtet, da sich BPMN zunehmend verbreitet und in Unternehmen etabliert [Müller 2014]. Ebenfalls hat sich der Reifegrad von BPMN seit 2009 kontinuierlich verbessert [Freund und Rücker 2012].

³<http://www.omg.org/spec/BPMN/2.0/>

⁴<https://www.oasis-open.org/committees/wsbpel/>

Da in dieser Arbeit keine sprachspezifische Schätzmethode entwickelt wird, lässt sich diese auf eine Vielzahl von Modellierungssprachen anwenden, die einen Geschäftsprozess mittels Aktivitäten beschreiben.

Nachfolgend werden die Sprachen BPEL, BPMN, sowie die sich noch in einer Beta-Version befindlichen Sprachen Decision Model and Notation (DMN)⁵ und Case Management Model and Notation (CMMN)⁶ beschrieben. Diese Sprachen haben gemeinsam, dass sich das erstellte Prozessmodell nach einer detaillierteren Implementierung auch ausführen lassen kann. Der hierbei beobachtete Effekt, dass aus einer visuellen Modellerstellung sich ein ausführbares Programm erstellen lässt, hat insbesondere bei den CEOs und nicht IT-affinen Stakeholdern⁷ Eindruck hinterlassen [Freund und Rücker 2012]. Allerdings kann auch BPMN nicht ohne weitere Implementierungsphase (vgl. Kapitel 1.2.2) direkt ausgeführt werden. Den Ansatz eines „Zero-Coding BPM“ klingt zwar interessant, ist aber ohne die Unterstützung der Entwickler und IT-Abteilung schwer realisierbar⁸.

1.4.1 WS-BPEL

Durch die fehlende graphische Unterstützung einer Modellierung und der stark technischen Ausrichtung konnte sich die Sprache WS-BPEL bisher nicht durchsetzen [Freund und Rücker 2012; Lessen, Lübke und Nitzsche 2011]. Dennoch ist es möglich, einen aus einem BPMN Model „funktionierenden“ BPEL Code zu generieren. Einige Firmen haben nach wie vor BPEL im Einsatz und im Rahmen von Service Orchestrated Architecture (SOA) ist BPEL auch weiterhin ein Thema [Lessen, Lübke und Nitzsche 2011].

BPEL basiert auf XML⁹ und wird zur Beschreibung von Geschäftsprozessen verwendet. BPEL ist ein industrieller Standard von OASIS¹⁰. Aufgrund der schwierigen Handhabung und der nicht vollständigen Unterstützung von graphischen Modellierungswerkzeugen wie dem Eclipse BPEL Designer¹¹, ist BPEL in der Handhabung schwieriger und konnte sich daher nicht durchsetzen.

Ein Prozess in BPEL ist durch mehrere Eigenschaften definiert:

- ◇ Aktivitäten, die verschiedene Dienste im Unternehmen darstellen (*Activities*)
- ◇ Kontrollflüsse, die den Ablauf der Aktivitäten steuern (*Control-Flows*)
- ◇ Vordefinition der Struktur von notwendigen Nachrichten (*Message-Flows*)

⁵<http://www.omg.org/spec/DMN/1.0/Beta2/>

⁶<http://www.omg.org/spec/CMMN/1.0/>

⁷Stakeholder wird als Umschreibung für die an BPM beteiligten Personen verwendet.

⁸<http://www.bpm-guide.de/2014/06/12/neues-whitepaper-der-mythos-zero-coding-bpm/>

⁹<http://www.w3.org/XML>

¹⁰<http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

¹¹<https://eclipse.org/bpel/>

Listing 1.1 BPEL Beispielprozess aus [BPEL 2.0 Specification]

```

1 <process name="Processname"
2   xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable">
3   <extensions>...</extensions>
4   <partnerLinks>
5     <!-- Note: At least one role must be specified. -->...
6   </partnerLinks>
7   <messageExchanges>...</messageExchanges>
8   <variables>...</variables>
9   <correlationSets>...</correlationSets>
10  <faultHandlers>...</faultHandlers>
11  <eventHandlers>...</eventHandlers>
12  activity (1...N)
13 </process>

```

Aus diesen Definitionen wird ein ausführbarer Prozess gebildet, der nur zusammen der WSDL Datei (Web Services Description Language) ausgeführt werden kann. Diese enthält eine genaue Beschreibung von Schnittstellen, aufzurufenden Diensten und den ein- und ausgehenden Nachrichten. Somit kann ein BPEL Prozess nur gemeinsam mit dieser umfangreichen WSDL-Datei ausgeführt werden. Listing 1 zeigt einen BPEL-Beispielprozess ohne Schnittstellendefinition. Dabei werden zu Beginn des <process> die Schnittstellen und Variablen deklariert. Anschließend folgen Aktivitäten¹², die in der Reihenfolge des beschriebenen Kontrollflusses ausgeführt werden.

1.4.2 BPMN 2.0

Die Business Process Modeling Notation wurde 2001 von Stephen A. White (IBM¹³-Mitarbeiter) zur graphischen Darstellung von Geschäftsprozessen entwickelt und 2004 von der Business Process Management Initiative (BPMI¹⁴) veröffentlicht. Inzwischen ist die Version BPMN 2.0 von der OMG als Standard verabschiedet worden. Die BPMN ist eine leicht verständliche graphische Modellierungssprache für Geschäftsprozesse. Sie richtet sich an alle Beteiligten im BPM, also sowohl an fachliche, als auch an technische Mitarbeiter [Gadatsch 2013; Müller 2014]. Die wichtigsten Symbole werden in Abbildung 1.4 dargestellt:

- ◇ Rechtecke entsprechen Aufgaben (Activities)
- ◇ Kreise beschreiben Ereignisse, wie beispielsweise den Start und das Ende des Prozesses (Events)
- ◇ Rauten entsprechen Entscheidungen (Gateways)

¹²<http://elib.uni-stuttgart.de/opus/volltexte/2014/9491/>

¹³<http://www.ibm.com>

¹⁴<http://www.bpmi.org>

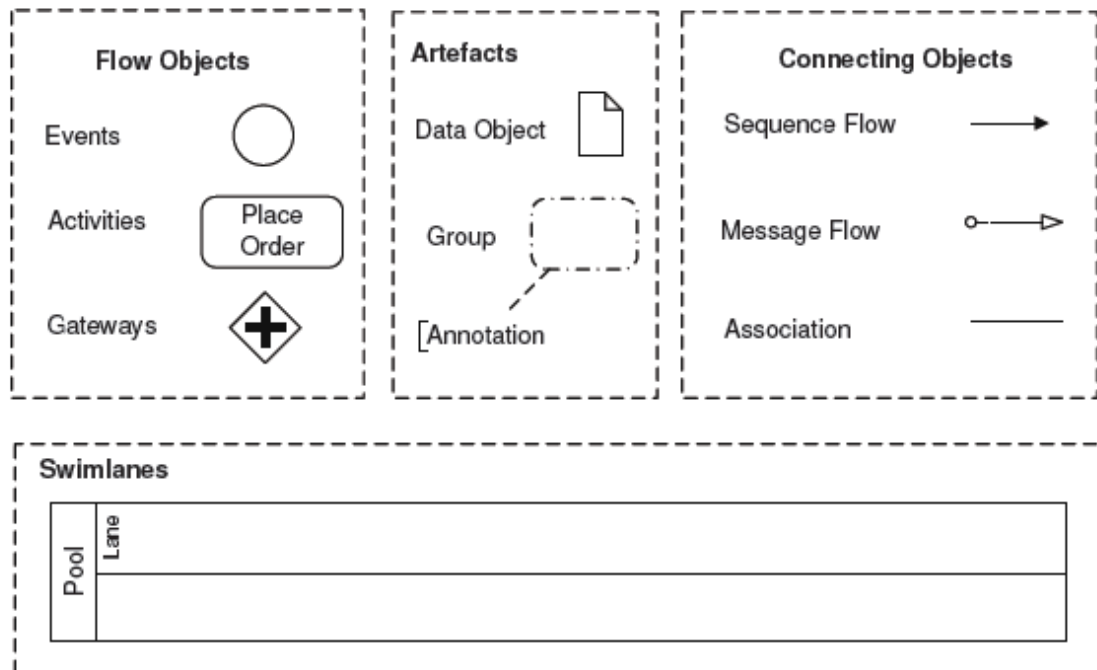


Abbildung 1.4: Kurzübersicht Elemente BPMN 2.0 [Weske 2012]

- ◇ Pfeile beschreiben den Kontroll- und Nachrichtenfluss
- ◇ Dokumentensymbol beschreibt Datenobjekte (als Teil der Artifacts)
- ◇ Prozessbeteiligte und Verantwortungsbereiche werden als Pools und Lanes beschrieben

Mit Hilfe dieser Symbole können Realitätsausschnitte des Geschäftsprozesses intuitiv auf einer fachlich-konzeptionellen Perspektive abgebildet werden [Gadatsch 2013].

1.4.3 CMMN und DMN

In Abgrenzung zu BPMN hat die OMG diese beiden Sprachen veröffentlicht. Case Management Model And Notation (CMMN) beschreibt hierbei „Fälle“, Decision Model And Notation (DMN) beschreibt Entscheidungen im Gegensatz zu Prozessen bei BPMN. Das Entstehen neuer Sprachen folgt aus der Tatsache, dass das Feld der Modellierung zunehmend erforscht wird und mehr Beachtung findet.

CMMN und DMN wurden ebenfalls wie BPMN 2.0 von der OMG veröffentlicht. Dabei sind die jeweiligen Sprachen für einen jeweils anderen Anwendungsfall gedacht:

BPMN	CMMN	DMN
Prozesse	Fälle (Cases)	Entscheidungen (Decisions)
Aktivitäten	Ereignisse	Regeln
Ablaufgesteuert	Kontextbezogen	Angewandt
Daten	Informationen	Wissen
Prozedural	Deklarativ	Funktional
Token	Event Condition Action	First Order Logic

Tabelle 1.1: Vergleich zwischen BPMN, CMMN und DMN, Quelle: [Kemsley 2015]

Tabelle 1.1 vergleicht die Eigenschaften der Sprachen und beschreibt dabei den Anwendungsfall:

- ◇ In BPMN werden Prozesse, auch Geschäftsprozesse, modelliert, die eingegebene Daten durch Aktivitäten verändern und dabei eine Ausgabe erzeugen (vgl. auch Kap. 1.2).
- ◇ In CMMN werden Fälle betrachtet, die durch im Kontext zueinander stehende Ereignisse beschrieben werden. Diese Ereignisse werden durch ein Triggerereignis ausgelöst. Dabei werden Informationen verarbeitet. Das Prozessmodell ist deklarativ.
- ◇ In DMN werden Entscheidungen modelliert, die durch Regeln verknüpft sind. Diese Regeln werden durch Wissen (übersetzt aus Knowledge) erstellt und die Abfolge ist dabei funktional. [Gadatsch 2013] definiert dabei Wissen als die Summe von Informationen und der Fähigkeit, diese zu verwenden.

Abbildung 1.5 zeigt exemplarisch einen möglichen Entscheidungsablauf mit DMN beim Autokauf als Beispiel mit dem Ziel, den Unterschied zu BPMN zu verdeutlichen.

Dabei werden Informationen (Input Data) zur Entscheidungsfindung in den abgerundeten „Kästchen“ dargestellt, während die Entscheidung durch das Viereck symbolisiert wird. Neben den Daten, gibt es auch bindende Vorschriften, die bei der Entscheidung eingehalten werden müssen. Im vorliegenden Beispiel ist dies die Größe der Garage, denn in diese muss das gewählte Fahrzeugmodell hineinpassen.

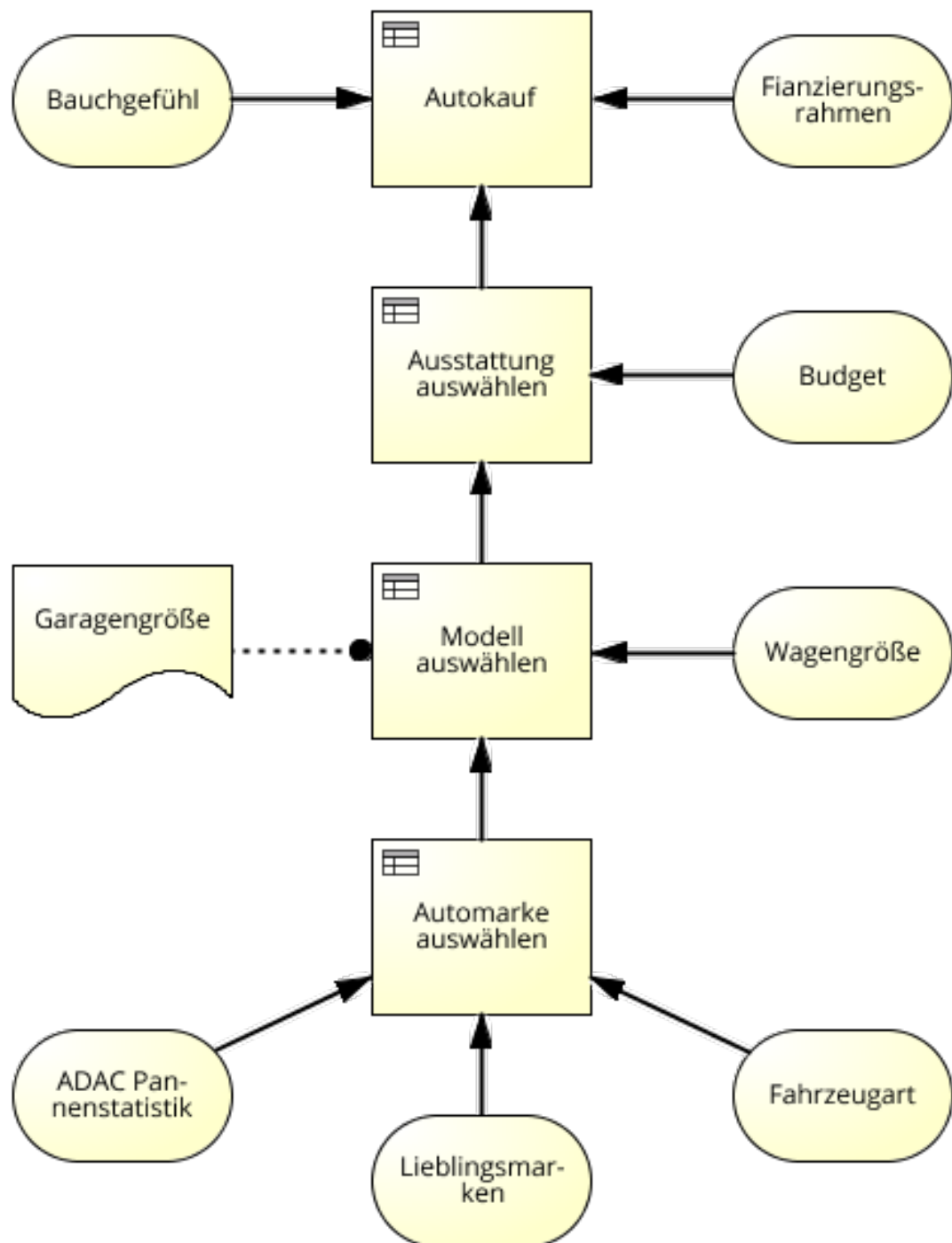


Abbildung 1.5: Beispielmodell DMN - eigene Darstellung

Kapitel 2

Forschungsstand

Die vorliegende Arbeit führt eine Untersuchung zum aktuellen Forschungsstand hinsichtlich der Aufwandschätzung bei BPM durch. Dieses Kapitel beschreibt die Vorgehensweise bei der Literaturrecherche und erläutert die dabei gewonnenen Erkenntnisse. Bei der Recherche werden der aktuelle Forschungsstand, sowie entfernte und angrenzende Forschungsfelder untersucht. Basierend auf den gewonnenen Erkenntnissen und der Selektion der relevanten Forschungsarbeiten folgt in Kapitel 5 die Entwicklung einer neuen Schätzmethode für die Geschäftsprozessmodellierung (GPM).

2.1 Aktueller Forschungsstand

Das Thema Business Process Management (BPM) wird derzeit viel diskutiert und stets weiter entwickelt und erforscht: Die stetig wachsende Anzahl an Mitgliedern im BPM-Netzwerk¹, das beinahe tägliche Erscheinen von neuen Blogeinträgen², das Entstehen von neuen Sprachen und Standards in Anlehnung an BPMN 2.0, wie beispielsweise CMMN und DMN, sowie die kontinuierliche Weiterentwicklung und zunehmende Verbreitung von BPMN 2.0 [Freund und Rücker 2012] deuten auf die zunehmende Etablierung von BPM in Unternehmen hin [Gadatsch 2013].

Umfragen wie die jährlich durchgeführte globale CIO-Studie der Firma IBM³ sowie die Umfrage zur Situationsanalyse von BPM in Deutschland im Rahmen der Studienarbeit von Sabrina Müller an der Universität Stuttgart im Jahr 2014 [Müller 2014] geben weitere Anhaltspunkte dafür, dass das Thema BPM wissenschaftlich untersucht und weiter aufgearbeitet wird.

Im Rahmen der durchgeführten Recherche wurden dabei Hochschulbibliotheken, die deutsche Nationalbibliothek⁴, Fachzeitschriften und Fachjournale, Google-Scholar⁵, sowie Blogeinträge zum Thema BPM als Zugangsquellen für Literatur verwendet.

¹<https://network.camunda.org/>

²<http://www.column2.com>, <http://www.bpm-plus.de>, <http://www.bpmn-buch.de>

³www.ibm.com

⁴<http://www.dnb.de>

⁵<http://scholar.google.de>

Nach der durchgeführten Recherche im Rahmen dieser Arbeit stellt es sich so dar, dass das Themenfeld der Aufwandsschätzung für die Modellierung von Geschäftsprozessen zwar erwähnt wird, aber noch keine wissenschaftliche Auseinandersetzung mit dem Thema stattgefunden hat. So haben sich einige Beratungsfirmen mit dem Thema auseinander gesetzt, allerdings nicht auf wissenschaftlicher Ebene⁶. Daher existiert noch keine empfohlene und standardisierte Vorgehensweise, wie der Aufwand für den Modellerstellungsprozess bei BPM geschätzt werden kann.

Neben Erfahrungswerten in Blogs, gibt es auch veröffentlichte Paper mit Vorschlägen zur Schätzung des Aufwandes. Allerdings sind diese Schätzmodelle für die Modellierungsphase nicht anwendbar, da die Eingabegrößen für diese Schätzungen erst nach der Modellierungsphase vorliegen und die Schätzung daher frühestens für die Implementierungsphase verwendet werden kann [Baklizky u. a. 2013; Kluza und Nalepa 2012; Marin und Quinteros 2014; Mishra und Kumar 2014; Rolon u. a. 2009].

Eine umfassende Schätzung für BPM-Projekte durch Zusammenfassung der gesamten Phasen des BPM-Lifecycle oder durch das Zusammenfassen von Modellierungsphase und Implementierungsphase beschreiben die Autoren [Bankhofer und Nissen 2013; Çulha und Doğru 2014; Thiemich und Puhlmann 2013]. Dabei wird keine direkte Berechnung für die Modellierungsphase vorgestellt, vielmehr kann diese, wenn überhaupt, anteilig aus den Gesamtkosten ermittelt werden.

Daher wurden weitere Recherchen durchgeführt, um die Faktoren zu identifizieren, die Einfluss auf den Modellierungsaufwand nehmen können und daher in einer Schätzung berücksichtigt werden sollten. Dabei wurden die Aspekte Prozessmaturität [Allweyer 2007; Fisher 2004], Einflussgrößen und Modellgröße [Baklizky u. a. 2013; Marin und Quinteros 2014; Mishra und Kumar 2014], Komplexität [Cardozo 2005; Gruhn und Laue 2006; Kluza und Nalepa 2012], Risikomanagement [Jakoubi und Tjoa 2009; Lhannaoui, Kabbaj und Bakkoury 2013], Zusammenarbeit und Personen [Freund und Rücker 2012; Gadatsch 2013; Meziani und Saleh 2011; Posluschny 2012; Van Der Aalst, Ter Hofstede und Weske 2003] und Best Practices für die Modellierung [Becker, Rosemann und Uthmann 2000; *BPMN Modeling Best Practices*; Komus 2011; Mendling, Reijers und Aalst 2010] betrachtet.

Die Ergebnisse daraus werden für die Entwicklung der eigenen Schätzmethode in Kapitel 5.3 herangezogen. Dabei werden relevante Arbeiten genauer erläutert.

2.2 Entfernte Forschungsfelder

Geschäftsprozesse enthalten viele verschiedene Komponenten, die auf die unterschiedlichsten Ebenen und Akteure innerhalb eines Unternehmens verteilt sein können. Die Einführung oder Anpassung eines Geschäftsprozesses tangiert daher sehr viele Bereiche innerhalb einer Organisation. Da also BPM mit mitunter stark einschneidenden Veränderungen einer

⁶<http://www.bpm-plus.de/2013/03/aufwandsschatzung-fur-die-prozessmodellierung/>

Organisation einhergeht, ist das Forschungsfeld „Change Management“ aus der betriebswirtschaftlichen Sicht interessant. Dieses Feld beschreibt das Veränderungsmanagement innerhalb einer Organisation.

Im Gegensatz zu einem Softwareprojekt, welches das Entwickeln eines Produktes zum Ziel hat, ist die Einführung von BPM betriebswirtschaftlich gesehen ein organisatorisches Projekt, was durchaus auch in den Bereich des Change Management fällt. Eventuell lassen sich auch hier Ansätze und Anregungen zur Aufwandsabschätzung aus der Projektsteuerung finden [Wieczorrek und Mertens 2010].

Da Change-Management, genau so wie BPM einen Unterzweig aus dem Projektmanagement, bildet wird hier das Feld des Projektmanagements betrachtet [Kraus, Becker-Kolle und Fischer 2010]). [Tiemeyer 2013] beschreibt BPM nicht als Projekt: „Alle eingeführten, permanent durchzuführenden Vorgänge und Prozesse in einer Organisation sind nicht als Projekt anzusehen.“ [Tiemeyer 2013]. Allerdings sind die Einführung von BPM, sowie Verbesserungsprozesse oder Business Process Reengineering (BPR) als Projekte betrachtbar, da sie klare Zielsetzungen haben, in Projektphasen aufgeteilt werden kann und in klare Aktionen und Aufgaben mit den jeweiligen Personen aufgeteilt werden kann.

Durch Vergleich des BPM-Lifecycle mit dem Plan Do Control Act (PDCA)-Cycle aus Abbildung 6 kann die Modellierungsphase mit der Planungsphase von Projekten verglichen werden, da sie dort mit enthalten ist. Die Ausführungsphase in BPM entspricht hierbei der Phase Do, Check und Act entsprechen jeweils der Überwachung und Optimierung.

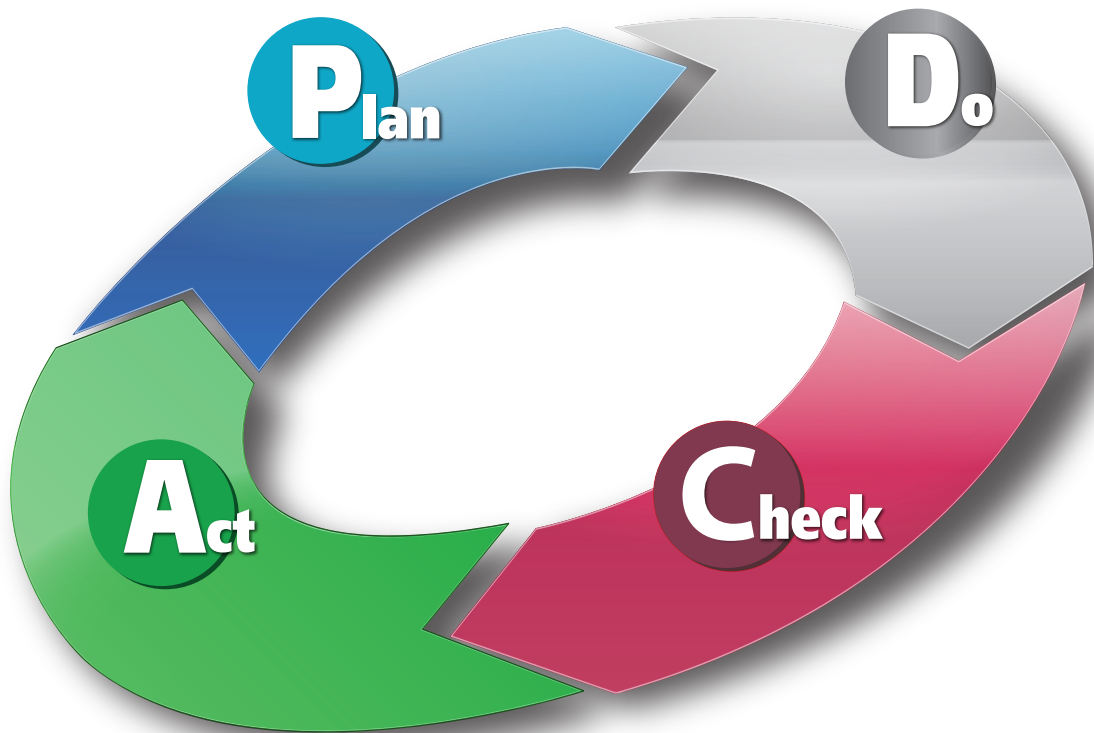


Abbildung 2.1: PDCA-Zyklus adaptiert aus [[PDCA Cycle](#)]

Verfahren für die Aufwandsschätzung von Projekten basieren allerdings alle auf vorhandenen Erfahrungswerten auf die zurückgegriffen werden kann. Im angrenzenden Feld der Softwareentwicklung gibt es ebenfalls Schätzverfahren, die auf Erfahrungswerte zurückgreifen, jedoch dabei auch Annahmen über die Größe des Projektes aus Eingangsgrößen verwenden, um eine genauere Schätzung abzugeben.

Die Methoden aus dem Projektmanagement und Change-Management sind daher in erfahrungswertbasierten Schätzmethoden aus der Softwareentwicklung bereits eingeflossen und werden in der Literatur übereinstimmend erwähnt [[Gadatsch 2013](#); [Herzwurm und Pietsch 2009](#); [Posluschny 2012](#); [Wieczorrek und Mertens 2007, 2010](#)].

2.3 Angrenzende Forschungsfelder

In Kapitel 2.1 wurden bereits die Erkenntnisse aus dem Forschungsfeld für BPM selbst beschrieben. Da BPM die Brücke zwischen Betriebswirtschaft und IT schlägt, ist die Softwareentwicklung ein angrenzendes Forschungsfeld.

Phasen Softwareentwicklung
1.1 – Requirements Specification
1.2 – System and Software Design
2.1 – Implementation and Unit Test
2.2 – Integration and Systems Test
3. – Operation and Maintenance

Tabelle 2.1: SE Lebenszyklenphasen, Quelle: [Khatibi und Jawawi 2011]

Betrachtet man die Phasen der Softwareentwicklung in Tabelle 2.1, so können bereits ähnliche Phasenbeschreibungen im Vergleich zum BPM-Lifecycle identifiziert werden.

Bei der Softwareentwicklung und bei BPM entsteht nach Durchlauf aller beschriebenen Phasen ein ausführbares Produkt. Abstrahiert ist das Prozessmodell also ein Stück Quellcode, welcher auf einer Plattform ausgeführt wird. Es ist naheliegend, dass die Erstellung eines Modells und insbesondere die IT-technische Verfeinerung und Spezifikation mit der Softwareentwicklung verglichen werden kann.

Daher wurden aus dem Bereich der Softwareentwicklung Schätzmethode und –verfahren betrachtet und werden in Kapitel 5 auf ihre Übertragbarkeit bewertet.

Methoden aus der Softwareentwicklung sind algorithmische Methoden. Die Genauigkeit der Schätzung resultiert dabei aus der Exaktheit der Ausprägung der erhobenen Einflussfaktoren [Wieczorrek und Mertens 2007].

[Jones 2007] stellt in seinem Buch „Estimating Software Costs“ die Herangehensweise von Schätzungen für die Softwareentwicklung und die über die letzten Jahrzehnte entstandenen Methoden sehr detailliert vor.

Eine gute Einführung in die in der Industrie verbreiteten und anerkannten Schätzverfahren gibt auch der Autor Stephan Frohnhoff in seiner Dissertation zur Weiterentwicklung der UseCase Methode [Frohnhoff 2009].

Grundlegende Vergleiche der vorhandenen Methoden, sowie die Entwicklung von neuen Herangehensweisen wie „Soft Computing“ stellen die Autoren Khatibi et. al in ihren Arbeiten „Software Cost Estimation Methods: A Review“ [Khatibi und Jawawi 2011] und „Neural Networks for Accurate Estimation of Software Metrics“ [Khatibi u. a. 2011] vor.

Eine Auswahl von, für die Adaption geeigneten, Schätzverfahren und deren detaillierte Beschreibung wird in Kapitel 4 vorgenommen.

2.4 Übertragbarkeit der Ergebnisse

Die in dieser Arbeit untersuchten Quellen gehen bei ihren Vorschlägen zur Aufwandsschätzung stets von einem bereits vorliegenden Prozessmodell aus. Dies entspricht bereits der Implementierungsphase des BPM-Lifecycles. Ferner wird hierbei davon ausgegangen, dass dieses Prozessmodell korrekt und vollständig ist [Mishra und Kumar 2014].

Allerdings ist es in der Realität vielmehr so, dass viele Firmen nicht über korrekte Modelle verfügen, der Prozess unvollständig oder veraltet ist, oder sich manche gar nicht erst die Mühe machen einen Prozess zu dokumentieren [Freund und Rücker 2012; Hinsch 2014],

Die Erstellung eines Geschäftsprozessmodells kann durchaus mit der Durchführung eines Softwareerstellungsprojektes verglichen werden, da abstrahiert betrachtet das Prozessmodell auch als ein zu schreibender Code dargestellt werden kann. Insbesondere ist bei WS-BPEL der Modellierungsprozess aufgrund der Spezifikation ⁷ und durch nicht umfassende Unterstützung der graphischen Modellierung innerhalb der Werkzeuge, wie Eclipse BPEL Designer⁸, hauptsächlich nicht visuell, sondern erfolgt im Quelltext [Lessen, Lübke und Nitzsche 2011].

Betrachtet man nun die Phasen des BPM-Lifecycles aus Kapitel 1.2.2 mit den Phasen der Softwareentwicklung, so erhält man folgende in Tabelle 2.2 dargestellte Adaption:

Aufgrund dieser möglichen Adaption werden in den weiteren Untersuchungen der Arbeit, Verfahren und Methoden zur Aufwandsschätzung aus der Softwareentwicklung verwendet und auf ihre Übertragbarkeit für BPM untersucht.

Phasen BPM-Lebenszyklus	Phasen Softwareentwicklung
1 – Analyse	1.1 – Requirements Specification
2 – Modellierung	1.2 – System and Software Design
3 – Implementierung	2.1 – Implementation and Unit Test
	2.2 – Integration and Systems Test
4 – Ausführung	3 – Operation and Maintenance
5 – Überwachung	
6 – Optimierung	

Tabelle 2.2: Mapping von BPM und SE Lebenszyklenphasen, Quelle: [Baklizky u. a. 2013]

⁷<http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

⁸<https://eclipse.org/bpel/>

Kapitel 3

Voraussetzungen zur Schätzung

Ein Schätzmodell ermittelt den Aufwand eines Projektes anhand von Daten, die in dieses Modell einfließen und zur Berechnung einer Schätzung verwendet werden. Diese Daten sind Annahmen über die Projektgröße (auch als Einflussgrößen bezeichnet) und Annahmen über die Projekteigenschaften, welche über verschiedene Faktoren abgebildet werden [Jones 2007].

Bevor ein Schätzmodell mit Eingabegrößen und Faktoren gefüttert werden und damit eine Schätzung erstellt werden kann, müssen ebendiese auch bekannt, aktuell und möglichst korrekt sein, um eine genauere Schätzung zu erhalten.

Insbesondere bei Prozessmodellen und deren Dokumentation (vgl. Kapitel 1.2.2) gibt es zu erwartende Diskrepanzen zwischen dem in der Realität ausgeführten Prozess und dem dokumentierten Prozessablauf [Freund und Rücker 2012].

Dieser Abschnitt befasst sich mit den Gründen für den Einsatz von BPM im Unternehmen und leitet daraus die Notwendigkeit eines Schätzmodells für die Modellierungsphase ab. Es wird analysiert, welche Bedingungen bei Erstellung einer solchen Schätzung vorliegen können und welche Informationen für die Schätzung benötigt werden, um eine Aussage über den zu erwartenden Aufwand treffen zu können.

3.1 Einsatz von BPM in Unternehmen

Wie in Kapitel 1 bereits erwähnt findet BPM immer mehr Einzug in Unternehmen und wird dort bereits als Notwendigkeit erachtet. Die Ziele von BPM sind dabei die Verbesserung von: Effektivität, Effizienz, Flexibilität, Schnelligkeit, Pünktlichkeit, Transparenz, (Qualität). Das Ziel ist es also, Prozesse so zu gestalten, dass Kosten, Zeiten und Qualitäten optimal sind. Optimale Prozesse führen auch zu optimalen Produkten („Process equals Product“).

Die jährlich durchgeführte Chief Information Officer (CIO) Studie der Firma IBM in Deutschland hat im Jahre 2011 ergeben, dass 32% der CIOs eine Verbesserung der Prozesseffizienz von ihrem IT-Bereich erwarten (CIO11). Der IT-Bereich wird hierbei vor allem als Förderer der Prozesseffizienz angesehen. Schätzungen zufolge wenden allerdings CIOs durchschnittlich geringfügig weniger Zeit für diese fundamentalen IT-Services auf.

Betrachtet man die Ergebnisse aus Umfragen aus den Jahren zwischen 2003 und 2014, so wird BPM als aktuelles Thema bei den Teilnehmern dieser Umfrage angeführt. (parenciteGadatsch¹³, mueller¹⁴).

Gründe für Veränderungsprojekte der vorhandenen Geschäftsprozesse wären aus strategischer Sicht [Posluschny 2012] :

- ◇ Überkapazität in Produktion
- ◇ Sättigung / geringes Wachstum der Nachfrage
- ◇ Zunehmende Differenzierung der Kundenwünsche
- ◇ Internationalisierung / Globalisierung
- ◇ Angleichung von Qualität und Know-How

Ein weiterer Grund für eine Einführung von BPM in Unternehmen ist die Dokumentationspflicht. Prozessorientierung ist ein Instrument der Effizienzsteigerung [Posluschny 2012] und sogar zwingend für die Zertifizierung nach ISO 9001 Norm [Hinsch 2014]. Die Autoren Freund, Rücker und Müller stellen allerdings aus den jeweiligen Erfahrungsberichten, bzw. der Umfrage, fest, dass Einführungsprozesse eher den Ausnahmefall bilden und BPM in den Unternehmen schon weit verbreitet ist [Freund und Rücker 2012; Müller 2014]

Typische Beispiele einer Anfrage zur Prozessmodellierung sind in Anlehnung an den BPM Blog von Wieschollek¹:

- ◇ Dokumentation eines Ablaufprozesses einer Abteilung
- ◇ Dokumentation einer bestimmten Applikation
- ◇ Standortbeschreibung und Dokumentation
- ◇ Unabhängigkeit von einzelnen Personen hin zur Rollenbeschreibung

Mit der Aufwandsschätzung zur Modellierung wird, durch die Entwicklung einer Schätzmethode im Rahmen dieser Arbeit, ein hilfreiches Werkzeug zur Seite gestellt, mit dessen Hilfe Projekte zur Geschäftsprozessoptimierung (GPO) besser geplant und geschätzt werden können.

¹<http://www.bpm-plus.de/2013/03/aufwandsschatzung-fur-die-prozessmodellierung/>

3.2 Prozessdokumentation

In Ihrem Buch nennen [Freund und Rücker 2012] aus ihrem Erfahrungsschatz als BPM-Berater drei mögliche Ausgangssituationen für ein BPM-Projekt:

1. Verbesserung bereits bestehender Prozesse
2. Dokumentation bereits bestehender Prozesse
3. Einführen neuer Prozesse (Prozesserhebung)

Der erste Fall, auch als Geschäftsprozessoptimierung (GPO) [Gadatsch 2013] bezeichnet, stellt hierbei die häufigste Anzahl der Projekte. Im Hinblick auf die Ziele und Gründe von BPM sollen hier unter anderem Effizienz, sowie Überwachung der Prozesse und kennzahlenbasierte Auswertung der Key Performance Indicators (KPI) verbessert werden. Hierbei ist der Ablauf von Prozessen bereits bekannt und im Falle einer zu erstellenden Modellierung des Prozesses, können die bekannten Abläufe und Kenntnisse über den Prozess direkt als Eingabegrößen in die Schätzung einfließen [Freund und Rücker 2012; Posluschny 2012; *Aufwandsschätzung für die Prozessmodellierung*].

Der zweite Fall, die Dokumentation der Geschäftsprozesse, kommt meist aus zwei Gründen vor. Neben den Zielen aus Kapitel 3.1 ist auch eine Prozessorientierung und damit deren Dokumentation Pflicht für die Zertifizierung des Unternehmens nach ISO:9001 Norm [Hinsch 2014]. Durch das eingeführte Qualitätsmanagement wird sichergestellt, dass Arbeitsabläufe nach einem gewissen Standard durchgeführt werden und damit nicht mehr zwingend personenabhängig sind.

Der dritte Fall ist vergleichsweise der seltenste Fall und stellt die Einführung von neuen Prozessen im Unternehmen dar. Dies kann aus verschiedenen Gründen heraus angestoßen werden. So könnte der Eintritt in einen neuen Markt der entscheidende Treiber hierfür sein, oder die Einführung eines neuen Produktes [Freund und Rücker 2012; Gadatsch 2013]. Hierbei fehlt eine Prozessdokumentation, welche nach dem camundaBPM Lifecycle [Freund und Rücker 2012], beschrieben in Abbildung 3.1, zuerst erstellt werden sollte, bevor der eigentliche BPM-Lifecycle existieren kann.

Das treffenderweise mit BPMN beschriebene Diagramm zeigt auf, dass vor dem erkennbaren Kreislauf über die Prozessdokumentation ein IST-Prozessmodell entwickelt wird, welches dann im Kreislauf modelliert und kontinuierlich verbessert wird. Dieses Dokument beschreibt den Prozessablauf und dient als Grundlage zur Prozessmodellerstellung bzw. dessen Verbesserung. Das erstellte Prozessmodell ist dabei nicht zwangsweise auch die Prozessdokumentation. Freund¹² unterscheiden hierbei auch zwischen Prozessmodellen der Ebene 1 und der Ebene 2². Ebene 1 Prozessmodelle beschreiben eine „logische“ Abbildung des Prozesses, während Ebene 2 Modelle die Interaktionen zwischen den Prozessbeteiligten darstellt. Durch die bessere Lesbarkeit von Ebene 1 Modellen eignen sich diese auch zur

²<http://www.bpm-guide.de/2011/08/19/bpmn-bei-der-deutschen-nationalbibliothek/>

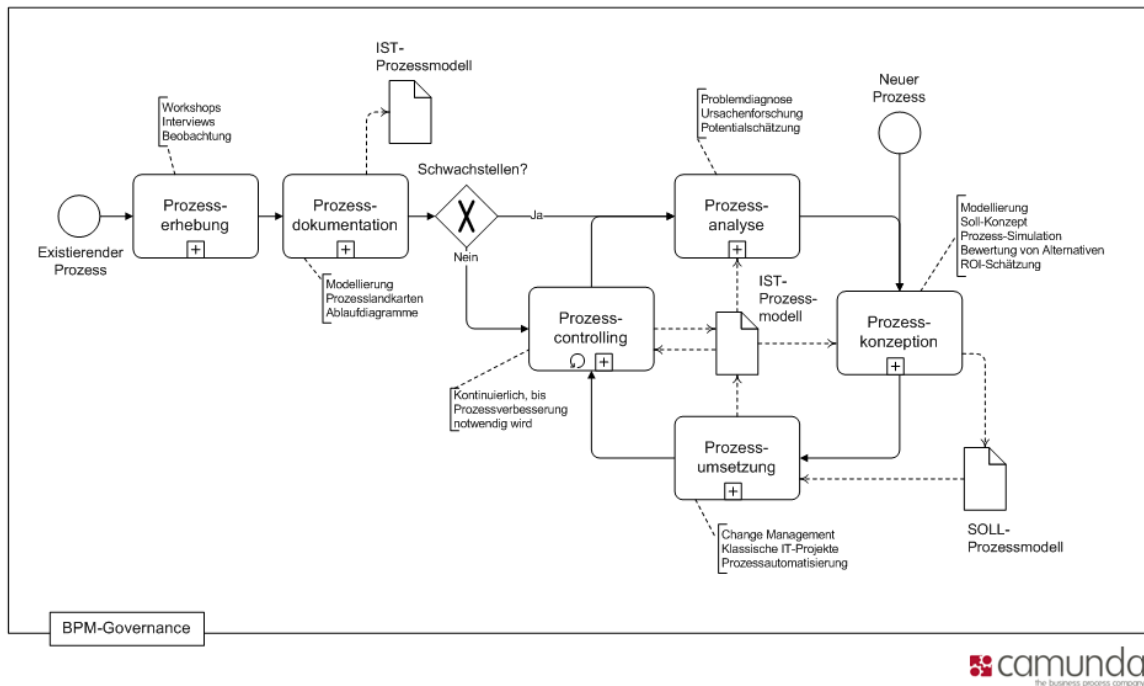


Abbildung 3.1: camundaBPM Kreislauf – Quelle: [Freund und Rücker 2012]

Prozessdokumentation. Wie allerdings Wiescholke in seinem Blog³ berichtet, eignen sich in BPMN erstellte Prozessmodelle nicht als Übersicht über die Prozesslandschaft, da diese, insbesondere auf Ebene 2, den Inhalt des Prozesses nicht sofort erkennen lassen.

3.2.1 Vorhandene Prozessdokumentation

Bei vorhandener Prozessdokumentation ist davon auszugehen, dass sich die prozessverantwortlichen Personen zumindest zu einem Zeitpunkt mit dem Prozess auseinandergesetzt haben. Nach ISO zertifizierte Unternehmen müssen die strikte Prozessorientierung der Norm erfüllen.

Die vorhandene Prozessdokumentation kann somit als Grundlage für die Aufwandsschätzung der Modellierung verwendet werden.

So stellt sich hierbei die Frage nach der Qualität der Prozessdokumentation und der damit verbundenen Validität der ableitbaren Einflussgrößen für die Schätzung. Qualitätseinbußen wären beispielsweise stark abstrahierte Sichten auf den Prozess für unterschiedliche

³<http://www.bpm-plus.de/2012/12/warum-man-fur-prozesslandkarten-kein-bpmn-verwenden-sollte-2/>

Betrachtergruppen oder Ebenen der Vertraulichkeit. Umgekehrt sind unübersichtliche Dokumentationen auch ein Hindernis, da man hieraus eventuell den Ablauf des Prozesses nicht auf den ersten Blick erkennen kann.

Weiterhin ist die Aktualität der Prozessdokumentation ebenfalls mit dem aktuellen Stand des tatsächlichen Prozesses zu gewährleisten, um eine möglichst korrekte Aussage über den Prozess zu erhalten.

Aus der Prozessdokumentation sollen auf daher Eingangsgrößen und Faktoren für eine Aufwandsschätzung ablesbar sein. Im Vorgriff auf Kapitel 5 könnten dies beispielsweise die Größe des Prozessmodells in Abhängigkeit der zu modellierenden Objekte oder die voraussichtliche Komplexität des zu erstellenden Prozessmodells sein.

3.2.2 Prozessdokumentation erstellen

Trotz der Zunahme der Wichtigkeit von Qualitätsmanagement und der zunehmenden Anzahl an ISO-Zertifizierten Unternehmen, kann es dennoch vorkommen, dass ein Unternehmen die Geschäftsprozesse nicht dokumentiert hat.

Daneben existiert auch die Möglichkeit, dass die zur Modellerstellung herangezogene Dokumentation unbrauchbar ist, da sie veraltet oder inkorrekt ist und daher in vollem Umfang neu erstellt werden muss.

Um das BPM-Projekt überhaupt beschreiben zu können, wird der momentane Zustand betrachtet und durch Analyse und Projektbeschreibung zu einem SOLL-Zustand entwickelt. Grundlage bei BPM-Projekten ist hierbei auch die vorhandene Prozessdokumentation. Da Schätzverfahren auf Einflussgrößen (vgl. Kapitel 4) oder Erfahrungswerten basieren, müssen diese Größen zunächst bestimmt werden oder bekannt sein, um Inputfaktoren für eine Schätzung zu erhalten.

Dies ist allerdings mit zusätzlichem Aufwand verbunden, da hierbei zunächst der Ablauf der Prozesse in Erfahrung gebracht werden muss, bevor man diesen mit Größen quantifizieren kann.

Abbildung 3.2 zeigt den Aufwand zur Prozessdokumentation in Relation zur Entwicklungsphase und dessen Streuung in Abhängigkeit der gewünschten Genauigkeit des Prozesses.

Die Abbildung ist im Bereich der Softwareentwicklung entstanden und beschreibt die Abweichung des geschätzten Personenaufwands (PE) in Abhängigkeit zur Entwicklungsphase und den jeweilig vorliegenden Daten.

Vorschlag zur Erstellung einer Prozessdokumentation:

[Tiemeyer 2013] schlägt vor, IT-Projekte mit einem Kick-Off Meeting zu beginnen und dies als Informationsveranstaltung für alle betroffenen Personen durchzuführen. Da für die Erstellung einer Prozessdokumentation nur die Phasen der Projektierung und die Erstellung des fachlichen Soll-Konzeptes in Frage kommen, wird dieses Kick-Off Meeting abgewandelt.

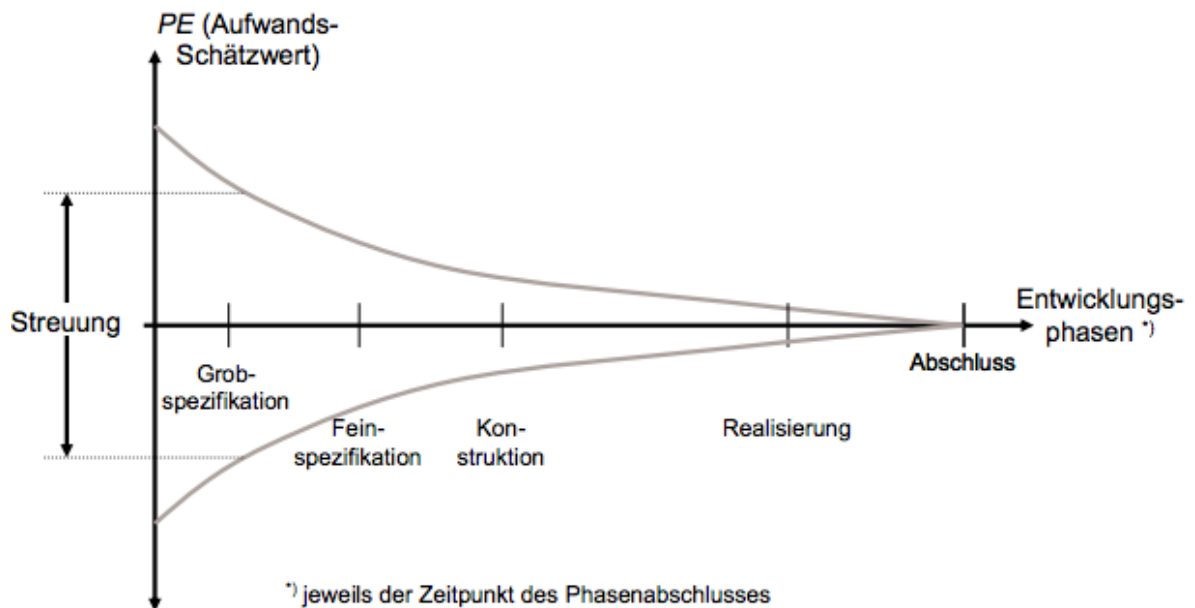


Abbildung 3.2: Streuung der Schätzung in Abhängigkeit von Projektphase – Quelle: [Frohnhoff 2009]

Abstrahiert auf die Erstellung eines Prozessdokumentation könnte dies wie folgt ablaufen: Im Kick-Off Meeting werden die Ziele besprochen und die Prozessbeteiligten befragt. [Freund und Rücker 2012] schlagen vor im ersten Schritt eine grobe Übersicht der zu modellierenden Prozesse zu erstellen. Dabei soll die Prozesserhebung maximal 10 Aufgaben oder Ereignisse (Flussobjekte) und 8 Artefakte (Kommentare und Datenobjekte) in der Übersicht beinhalten. Das Prozessmodell soll zudem auf einer Din A4-Seite darzustellen sein [Freund und Rücker 2012].

[Posluschny 2012] stellt folgende Empfehlungen zur Prozessgestaltung basierend auf Praxiserfahrung mit Hinblick auf Business Process Reengineering (BPR) und Geschäftsprozessoptimierung (GPO) auf:

- ◇ Jeder Geschäftsprozess ist in Teilprozesse, Prozess- und Arbeitsschritte zu unterteilen
- ◇ Für jeden Geschäftsprozess ist eine zeit- und ressourcengünstige Ablaufstruktur festzulegen
- ◇ Die Anzahl der an der Prozessdurchführung beteiligten Personen bzw. Funktionsbereiche ist zu minimieren
- ◇ Kontrollen sind im geringstmöglichen Ausmaße vorzunehmen

Die Abbildung von Geschäftsprozessen kann durch Betrachtung der folgenden fünf Aspekte erfolgen [Posluschny 2012]:

- ◇ Steuerungsaspekt: Was wird wann und warum getan (Reihenfolge, Planlänge, Parallelität...)?
- ◇ Organisationsaspekt: Wer erledigt wo welche Tätigkeit (Pools, Lanes...)?
- ◇ Informationsaspekt: Welche Informationen werden wie weitergegeben (Input, Output, Daten...)?
- ◇ Kontrollaspekt: Erreicht der Prozess sein Ziel (Menge, Kosten, Zeit, Qualität,...)?
- ◇ Sicherheitsaspekt: Wer darf was im Prozess (Sichten, Zugriffe,...)?

Durch die Beantwortung dieser Fragen kann innerhalb des Workshops der dokumentierte Prozess verifiziert und auf Vollständigkeit hin untersucht werden.

Daraus lassen sich für den Workshop folgende Empfehlungen ableiten:

- ◇ Top-Down Ansatz, der den zu dokumentierenden Prozess immer weiter aufspaltet und als Prozesslandschaft darstellt
- ◇ Möglichst geringe Anzahl an beteiligten Personen
- ◇ Experten miteinbeziehen, wo notwendig
- ◇ Iterativ untersuchen, bis gewünschte oder benötigte Tiefe erreicht
- ◇ „Keep it simple“

Die Darstellungsmöglichkeiten der erstellten Prozessdokumentation sind daher offen zu lassen, sollten allerdings wie bereits erwähnt die Entnahme der für die Schätzung wichtigen Informationen nicht erschweren. Ebenfalls sollte die erstellte Prozessdokumentation nicht gleichzeitig dem Prozessmodell selbst entsprechen.

Aus diesem Kapitel gewonnene Erkenntnisse fließen in die Entwicklung der Schätzmethode in Kapitel 5 ein.

Kapitel 4

Überblick Schätzmethoden

In diesem Abschnitt wird eine Einführung zu Schätzmethoden der Softwareentwicklung gegeben. Neben der Vorstellung von, für die Anpassung auf das Feld von BPM, geeigneten Schätzmethoden, werden auch die Eingabegrößen und Faktoren für die Erstellung identifiziert.

Das Standardwerk *Estimating Software Costs* [Jones 2007] gibt hier eine sehr umfangreiche Einführung in das Themengebiet der Aufwandsschätzung für die Softwareentwicklung. Diese Quelle kann für die Auswahl und Entwicklung einer für BPM anwendbaren Schätzmethode dienen. Aus Gründen der Übersichtlichkeit, werden nur ausgewählte Schätzmethoden und die zugrunde liegenden Vorgehensweisen vorgestellt.

4.1 Schätzmethoden aus der Softwareentwicklung

Es werden ausgewählte Schätzverfahren aus der Softwareentwicklung im Hinblick auf deren Fähigkeit zur Adaption für BPM vorgestellt. Wie bereits in Kapitel 2.4 erörtert, bildet die Softwareentwicklung ein angrenzendes Forschungsfeld und dient daher als Quelle für verschiedene Schätzverfahren. Insbesondere basieren nahezu alle Schätzmethoden aus dem Projektmanagement auf Erfahrungswerten.

Aufgrund der Vielzahl an vorhandenen Schätzmethoden, können in dieser Arbeit nicht alle untersucht werden. Hauptsächlich werden agile Methoden ausgeschlossen, da diese bereits in Kapitel 2 betrachtet wurden. Sie eignen sich, gemäß der jeweiligen Beschreibung, als Schätzmethode bei bereits erstelltem Prozessmodell für die Phase der Implementierung innerhalb des BPM Lebenszyklusmodells [Baklizky u. a. 2013; Çulha und Doğru 2014; Meziani und Saleh 2011; Mishra und Kumar 2014; Thiemich und Puhlmann 2013]. In ihrer Arbeit „Business Process Points-A Proposal To Measure BPM Projects“ schlagen die Autoren [Baklizky u. a. 2013] et al. eine Vorgehensweise zur Zählung der BPM Projektpunkte vor. Dadurch soll sich die Projektgröße ermitteln lassen [Baklizky u. a. 2013]. Da die Zählung der Punkte auf bereits vorliegenden Modellen und spezifischen Dokumentationen basiert, eignet sich diese nicht direkt zur Schätzungsermittlung während der Modellierungsphase, da zum Zeitpunkt der Zählung im Verfahren bereits alle Größen bekannt sind. Wie auch die erwähnten agilen Methoden ist dieses Verfahren ebenfalls in der Implementierungsphase anwendbar.

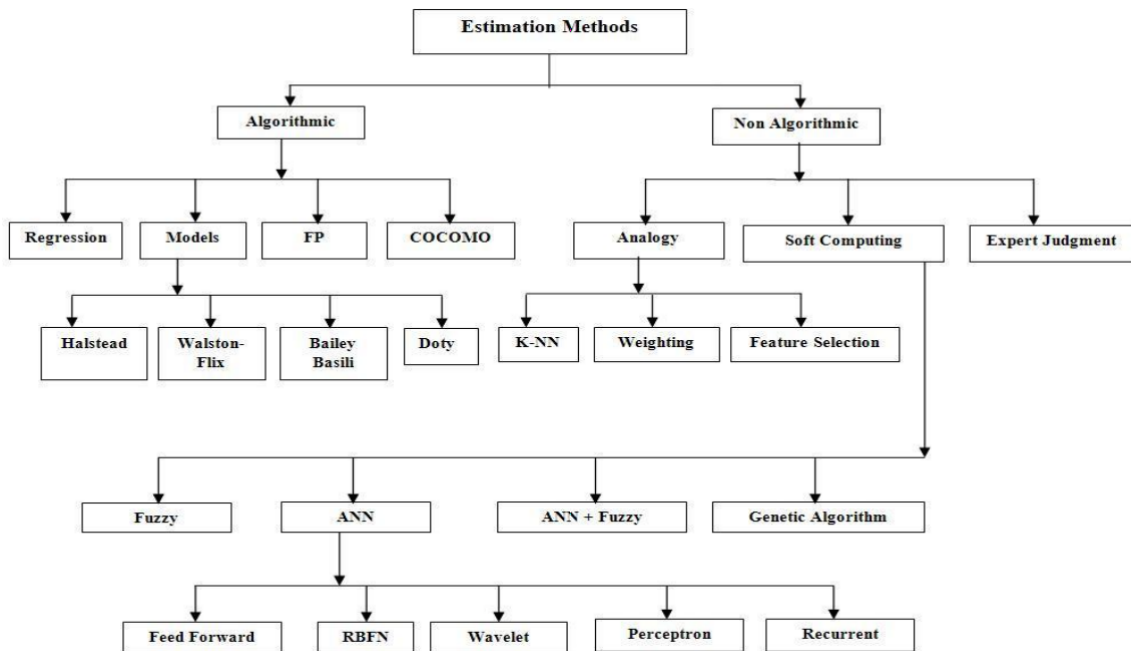


Abbildung 4.1: Übersicht Schätzmethoden für SE, Quelle: [Khatibi und Jawawi 2011]

Da der Fokus dieser Arbeit auf der Prozessmodellerstellung liegt und zu diesem Zeitpunkt im BPM Lebenszyklus noch nicht alle benötigten Daten zur Verwendung in bereits existierenden Schätzmodellen (vgl. Kapitel 2) vorliegen, werden noch andere Verfahren und Vorgehensweisen zur Schätzung benötigt. Aus dem Bereich des Projektmanagements in Abschnitt 2.2 basieren alle diese Daten auf Erfahrungswerten, auf die sich die Schätzung stützt. Da im Falle einer fehlenden oder noch zu erstellenden Prozessdokumentation noch keine Erfahrungswerte und Eingabegrößen vorliegen, sollte eine Methode gewählt werden, die neben schätzbaren Eingabegrößen auch detailliertere Annahmen über die Komplexität mit einbeziehen kann. Allein die Anzahl der Aktivitäten als Komplexitätsmetrik für ein GPM reicht nicht aus [Gruhn und Laue 2006].

Die nachfolgende Abbildung aus [Khatibi und Jawawi 2011] gibt eine Übersicht über vorhandene Schätzmethoden aus der Softwareentwicklung und teilt diese entsprechend ihrer Vorgehensweise in Kategorien ein.

Die Schätzmethoden lassen sich in algorithmische Methoden und nicht-algorithmische Methoden unterteilen. Algorithmische Methoden errechnen über Parametrisierung oder Gewichtung von Faktoren den Aufwand in Abhängigkeit von den gewählten Eingangsgrößen. Bei nicht-algorithmischen Methoden kommen Verfahren, wie die „rule of thumb“ [Jones 2007] oder Analogiemethoden zum Einsatz. Hier wird auf einen Erfahrungswert zurückgegriffen, um die Schätzung zu erstellen. Die nicht-algorithmischen Methoden bilden den größeren Teil in der Abbildung. Neben der Expertenschätzung und den Analogiemethoden gibt es

auch Soft Computing Methoden, die algorithmische und nicht-algorithmische Methoden verbinden [Khatibi und Jawawi 2011].

Im Rahmen ihrer Untersuchung von 20 Schätzmethoden haben die Autoren [Noth und Kretzschmar 1986] festgestellt, dass sich FP besonders für den Einsatz der Phase 1 der Softwareentwicklung (SE) (Bedarfsanalyse und Lösungsentwurf) eignet [Noth und Kretzschmar 1986]. Wie in Tabelle 1 aufgezeigt, entspricht dies den in dieser Arbeit betrachteten Phasen 1 und 2 des BPM Lebenszyklus. [Noth und Kretzschmar 1986] schreiben weiter über FP, dass dieses Verfahren eher für kaufmännische Anwendungen geeignet sei. Hier lässt sich ein Bezug zu BPM herstellen. Ebenso empfehlen die Autoren FP und COCOMO für den praktischen Einsatz.

FP ist am weitesten in der Industrie verbreitet [*COCOMO II Model Definition Manual*; Khatibi und Jawawi 2011; Noth und Kretzschmar 1986] und auch seit 2003 als ISO-Standard Nr. 20926:2003¹ eingeführt. COCOMO, FP und UseCase basieren auf algorithmischen Schätzungen, bei welchen in der Auswertung auf Erfahrungswerte zurückgegriffen wird. So existieren für FP und UseCase beispielsweise Umrechnungstabellen von Punkten zu PM, die nach Projektabschluss aktualisiert werden [Jones 2007; Noth und Kretzschmar 1986]. Bei COCOMO wurden in einer umfangreichen Untersuchung von [B. Boehm und Harrowitz 2000] Einflussfaktoren und deren Werte für die Gewichtung innerhalb der Schätzmethode ermittelt. Damit decken sie das breite Spektrum von verschiedensten Einflussgrößen in die Schätzung ab. Von zählbaren Faktoren, wie LOC, über die Gewichtung von weiteren Faktoren wie Komplexität und Qualität. Da neben Eingangsgrößen für die Schätzung auch beispielsweise der Einfluss von Modellkomplexität, die Erfahrung des Modellierers oder der Umgang mit Modellierungswerkzeugen über Skalenfaktoren und Kostentreiber eingebracht werden kann, können diese drei Schätzmethoden durch Adaption der Parameter für BPM Schätzungen angepasst werden.

Aufgrund der weiten Verbreitung von UseCase und Function Point sowie der Kombinationsmöglichkeit beider mit COCOMO II (UseCase3.0, COCOMO Boehm) sowie der Adaptionfähigkeit von COCOMO, wie ADA COCOMO [Frohnhoff 2009] belegt, werden diese Schätzmethoden in den engeren Betracht gezogen und nachfolgend vorgestellt.

4.1.1 Function Point (FP)

Die im Jahre 1973 von Allan J. Albrecht [Jones 2007; Noth und Kretzschmar 1986] entwickelte Function-Point Methode ist eine Analogie- und Gewichtungsmethode, welche den Aufwand des Projekts anhand von dessen Umfang und Komplexität ermittelt. Die Function-Point Methode wurde von Albrecht als IBM-Mitarbeiter entwickelt und wird dort seit 1981 eingesetzt [Noth und Kretzschmar 1986]. FP ist eine akzeptierte und bis heute weit verbreitete Methode in der Industrie [Baklizky u. a. 2013]. Die International Function Point Users Group (IFPUG) geht sogar soweit und nennt FP „eine der besten verfügbaren Methoden“ (DASMA – www.dasma.org).

¹<http://www.dasma.org>

Function-Point Kategorien	Komplexitätsgewichtung		
	Leicht	Mittel	Komplex
Eingabedaten	3	4	6
Abfragen	4	5	7
Ausgabedaten	3	4	6
Datenbestände	7	10	15
Schnittstellen	5	7	10

Tabelle 4.1: Function Point Kategorien adaptiert aus [Khatibi und Jawawi 2011]

Bei FP werden die Funktionen der Software gezählt und in Kategorien eingeordnet. Tabelle 4.1 gibt eine Übersicht über die Kategorien und deren Gewichtung zur Ermittlung des Schwierigkeitsgrades.

Aus den fünf Kategorien (Eingabe, Ausgabe, Dateien, Abfragen, Schnittstellen) erhält man zusammen mit deren Komplexitätsgewichtung die „Unadjusted Function Points“ (UFP). Diese erlangt man durch Zählung der Funktionen der Software und Einordnung in die beschriebenen Kategorien. Neben der Zählung der Function-Points werden diese zusätzlich in die drei Komplexitätsklassen „leicht“, „mittel“ und „komplex“ eingeteilt. Je nach Einteilung in die verschiedenen Kategorien werden die jeweiligen Punkte gezählt.

Im nächsten Schritt bestimmt man den „degree of influence“ (Grad des Einflusses) [Noth und Kretzschmar 1986] über 14 technische Komplexitätsfaktoren wie in Tabelle 4.2 beschrieben.

Jeder Faktor wird nach seinem Einfluss von 0 bis 5 bewertet [Noth und Kretzschmar 1986]:

- ◇ 0 = kein Einfluss
- ◇ 1 = gelegentlicher Einfluss
- ◇ 2 = mäßiger Einfluss
- ◇ 3 = mittlerer Einfluss
- ◇ 4 = bedeutsamer Einfluss
- ◇ 5 = starker Einfluss

Insgesamt können somit bis zu 70 Punkte vergeben werden. Den „Total Complexity Factor“ (Gesamter Komplexitätsfaktor, TCF) erhält man durch Division der Summe aller Faktoren durch 100 und dessen Addition des Quotienten zu 0,65. Anschließend werden die UFP mit dem TCF multipliziert, um die gewichteten FP zu erhalten. Das endgültige Ergebnis der gewichteten FP liegt somit zwischen 65% und 135% ($\pm 35\%$) der ursprünglich ermittelten UFP. Diese Spanne basiert auf empirisch gewonnenen Untersuchungsergebnissen während der Untersuchung von Albrecht [Noth und Kretzschmar 1986].

Faktor	Beschreibung	Faktor	Beschreibung
F1	Verlässliche Datensicherung	F8	Datenkommunikation
F2	Verteilte Datenverarbeitung	F9	Leistungsfähigkeit
F3	Verfügbare Systemressourcen	F10	Online Dateneingabe
F4	Benutzerfreundlichkeit	F11	Online Aktualisierung
F5	Komplexe Schnittstellen	F12	Aufwändige Verarbeitung
F6	Wiederverwendbarkeit	F13	Einfache Installation
F7	Mehrere Standorte	F14	Anpassungsmöglichkeit

Tabelle 4.2: TCF in Function Point – Übersetzt aus [Frohnhoff 2009; Khatibi und Jawawi 2011]

Mit Hilfe der Erfahrungstabelle, welche initial von der IBM erstellt wurde, lässt sich der Aufwand in Relation zu FP in Personenmonaten (PM)² ablesen. Das Unternehmen kann dann die Personenmonate in Kosten je nach eigenem Erfahrungswert umrechnen. Nach Projektabschluss sollte diese Tabelle mit Hilfe der neu gewonnenen Erfahrungswerte aktualisiert werden. Mit steigender Anzahl der Projekte und mit jeder Aktualisierung der Tabelle wird die Function-Point-Aufwand-Relation [Noth und Kretschmar 1986] verfeinert und die Schätzung präziser.

Mit den gewichteten FP kann auch über Umrechnungstabellen in Abhängigkeit von der benutzten Sprache die Anzahl der LOC bestimmt werden, um diese Größe dann für andere Schätzmethoden, wie beispielsweise COCOMO, zu verwenden. Diese Tabellen sind bei der International Function Point Users Group (IFPUG)³ erhältlich. Für die Modellierungssprache BPMN 2.0 war zum Zeitpunkt der Erstellung der Arbeit noch keine Faktortabelle erhältlich.

Eine detailliertere Beschreibung des Ablaufs bei einer Schätzung im Bezug zur GPM mittels der Function-Point Methode folgt in Kapitel 5.3.2

4.1.2 Use Case Points (UCP)

Die Use Case Points Methode ist eine von [Karner 1993] entwickelte Schätzmethode. Die Methode wurde im Jahre 1993 im Rahmen seiner Diplomarbeit entworfen und basiert auf der Vorgehensweise von FP. Dabei ist UCP auch für die objektorientierte (OO) Softwareentwicklung anwendbar. FP und UCP sind nicht direkt aufeinander abbildbar [Frohnhoff 2009].

„Ein Use Case erfasst einen Vertrag zwischen den Stakeholdern eines Systems hinsichtlich des Systemverhaltens. Der Use Case beschreibt das Systemverhalten unter verschiedenen Bedingungen als Antwort auf eine Anfrage von einem der Stakeholder, genannt Actor. Der Actor löst eine Interaktion

²Personenmonate ist ein anderer Begriff für Mannmonate (MM)

³<http://www.ifpug.org>

Komplexität	Beschreibung	Gewicht
einfach	Ein Use Case ist einfach, wenn er über 3 oder weniger Transaktionen inklusive alternativer Pfade verfügt. Der Use Case sollte mit weniger als 5 Analyseobjekten realisiert werden können.	5
mittel	Ein Use Case ist mittel, wenn er über zwischen 3 und 7 Transaktionen inklusive alternativer Pfade verfügt. Der Use Case sollte mit 5 bis 10 Analyseobjekten realisiert werden können.	10
komplex	Ein Use Case ist komplex, wenn er über mehr als 7 Transaktionen inklusive alternativer Pfade verfügt. Der Use Case wird mit mindestens 10 Analyseobjekten realisiert.	15

Tabelle 4.3: Gewichtung Use Cases - deutsche Übersetzung aus [Frohnhoff 2009]

mit dem System aus, um ein bestimmtes Ziel zu erreichen. Das System antwortet unter Berücksichtigung der Interessen aller Stakeholders. Abhängig von der speziellen Systemanfrage und dem Anfragekontext können sich eine unterschiedliche Reihenfolge des Systemverhaltens oder unterschiedliche Szenarien ergeben. Der Use Case fasst alle möglichen unterschiedlichen Szenarien zusammen.“ [Frohnhoff 2009].

Für die Ermittlung der „Unadjusted Use Case Points“ (UUCP) werden Use Cases und Actors gezählt und in Komplexitätsklassen von „einfach“, „mittel“ oder „komplex“ eingeteilt, wobei hier wieder Gewichte angehängt werden.

Erster Schritt:

Bewertung der Use Cases nach Anzahl der Transaktionen innerhalb eines UseCases. [Karner 1993] gibt keine Definition für eine Transaktion. Es sind jedoch für jeden Use Case Schritte aufgezeichnet, die die auszuführenden Aktionen beschreiben. Transaktionen können dabei als Interaktionen mit den Actors und dem System (Automatisierte Aktionen) betrachtet werden [Frohnhoff 2009].

In Tabelle 4.3 wird die Zuordnung der Gewichte in Abhängigkeit von der Anzahl der Transaktionen des Use Cases dargestellt.

Die Summe aller gewichteten Use Case Points ergibt die „Unadjusted Use Case Weights“ (UUCW) [Frohnhoff 2009].

Zweiter Schritt:

Nun folgt die Bewertung der Actors, welche den Use Cases zugeordnet sind. Analog zum ersten Schritt werden ebenfalls Gewichte nach den drei Komplexitätsklassen zugeordnet. Tabelle 4.4 beschreibt die Einteilung und Zuordnung der Gewichte für die Actors.

Komplexität	Beschreibung	Gewicht
einfach	Mit Systemschnittstellen (API) kommunizierende Actors werden als ?einfach? betrachtet.	1
mittel	Ein Actor ist ?mittel?, wenn er: 1. an ein anderes System über ein Protokoll (Wie z.B. TCP/IP ^a) angebunden ist 2. menschliche Interaktion über Kommandozeile besitzt	2
komplex	Ein Actor ist ?komplex?, wenn er über eine graphische Benutzerschnittstelle (GUI) interagiert.	3

Tabelle 4.4: Zuordnung Actor Gewichte - adaptiert aus [Frohnhoff 2009]

^a<https://www.rfc-editor.org/rfc/rfc793.txt>

Die Summe aller gewichteten Actors ergibt die „Unadjusted Actor Weights“ (UAW).

Die Summe aus UUCW und UAW ergibt die „Unadjusted Use Case Points“ (UUCP).

Wie bei FP werden im letzten Schritt die UUCP mittels eines TCF und dem in UCP neu eingeführten Umgebungsfaktor „Environmental Factor“ (EF) korrigiert und dadurch die „Adjusted Use Case Points“ (AUCP) ermittelt.

Dritter Schritt:

Die Ermittlung des TCF erfolgt, im Vergleich zur Vorgehensweise in FP, zwar mit der selben Einteilung des Einflussgrades von 0 bis 5 (vgl. Kapitel 4.1.1), allerdings werden hier den einzelnen Einflussgrößen noch Gewichte angehängt, mit denen der Einflussgrad multipliziert wird. Die Berechnung des TCF erfolgt dann wieder analog zu FP: Summe der nun gewichteten Faktoren durch 100 dividieren und zu 0,58 addieren. Der TCF kann somit zwischen 0,58 und 1,28 schwanken.

Tabelle 4.5 gibt eine Übersicht über die 13 Einflussgrößen des TCF.

Analog zum TCF wird der EF durch acht Einflussgrößen bestimmt, wobei die Bewertungsskala von 0-5 mit unwichtig bis sehr wichtig bezeichnet ist.

Die Berechnung des EF erfolgt anders als die des TCF. Die Summe der nun gewichteten Faktoren wird durch 300 dividiert und von 1,405 subtrahiert. Durch die Subtraktion des EF von der Konstante vergrößern nun Einflussfaktoren, die mit 0 bewertet wurden, den EF. Für die Gewichte bei E2 und E7 wurde ein negativer Wert gewählt, sodass sich die Bewertungsbezeichnung dort umdreht. Der EF kann somit zwischen 0,430 und 1,705 schwanken.

Tabelle 4.6 zeigt die Einflussgrößen und deren Gewichte:

Faktor	Einflussgröße	Gewicht
F1	Verteilte Systeme	2
F2	Leistungsfähigkeit der Anwendung, in Reaktionszeit oder Durchsatz	1
F3	Endbenutzer Effizienz (Online)	1
F4	Komplexe interne Verarbeitung	1
F5	Wiederverwendbarkeit des Codes	1
F6	Einfache Installation	0,5
F7	Einfache Bedienung	0,5
F8	Portierbarkeit	2
F9	Anpassbarkeit	1
F10	Parallelität	1
F11	Besondere Sicherheitsfunktionen	1
F12	Zugriff für Drittparteien	1
F13	Besondere Benutzertrainingsmöglichkeiten	1

Tabelle 4.5: Einflussgrößen des TCF übersetzt aus [Frohnhoff 2009]

Einfluss	Einflussgröße	Gewicht
E1	Vertraut mit Rational Unified Process ⁴	1,5
E2	Teilzeit Arbeiter	-1
E3	Fähigkeiten Analyst	0,5
E4	Anwendungserfahrung	0,5
E5	Erfahrung Objektorientierung	1
E6	Motivation	1
E7	Diffizile Programmiersprache	-1
E8	Stabile Bedingungen	2

Tabelle 4.6: Einflussgrößen des EF – übersetzt aus [Frohnhoff 2009]

Vierter Schritt:

Abschließend werden die „Adjusted Use Case Points“ (AUCP) durch Multiplikation von UUCP mit TCP und EF berechnet. Der Projektaufwand in Bearbeiter-Stunden errechnet sich aus der Multiplikation der AUCP mit einem, durch die Organisation zu eichenden, Produktivitätsfaktor im Bereich zwischen 20 und 35 Stunden pro AUCP [Frohnhoff 2009].

⁴Softwareentwicklungsprozess der Produktlinie Rational IBM – www.ibm.com

4.1.3 Constructive Cost Model (COCOMO)

Da das Constructive Cost Model (COCOMO) wurde von Barry W. Boehm Anfang der 1980er Jahre entwickelt. Eingabegröße für die Schätzung sind KLOC⁵. Der Projektaufwand (Project Effort PE) in PM wird durch die Formel $PE = m \text{ KLOC}^n$ berechnet. Der Faktor m und die Metrikzahl n basieren auf Erfahrungswerten [Barry W. Boehm 1981; Frohnhoff 2009]. Dabei werden die Projekte in drei Komplexitätsklassen eingeteilt [Barry W. Boehm 1981]:

- ◇ „organic mode“ entspricht einfachen Projekten
- ◇ „semi-detached“ beschreibt mittelschwere Projekte
- ◇ „embedded“ Projekte sind komplex

Projektart	m	n	Definition
einfach	2,4	1,05	Kleines Team; geringer Zeitdruck; bekannte Werkzeuge, Arbeitsumgebung und Hardware; bis 50 KLOC
mittelschwer	3	1,12	Nicht alle unter ?einfach? genannten Punkte treffen zu; bis 300 KLOC
komplex	3,6	1,2	Unbekannte Umgebung und Hardware, Echtzeitanforderungen; komplexe Rahmenbedingungen; über 300 KLOC

Tabelle 4.7: Übersicht Berechnungsfaktoren und Projektarten [Barry W. Boehm 1981; Frohnhoff 2009]

Tabelle 4.7 gibt die Werte zu m und n in Abhängigkeit von der Einteilung der Projektart in ihre Komplexitätsklasse und Auskunft darüber, diese richtig einzuordnen.

Boehm veröffentlichte im Sommer 2000 als Weiterführung zum ersten Modell COCOMO II (boehmoo). Ziel des neuen Modells war es, die Schätzgenauigkeit zu verbessern und verschiedene Projektarten abzubilden [Frohnhoff 2009]. Dabei führte er drei Submodelle ein:

- ◇ *Application Composition Model*
- ◇ *Early Design Model*
- ◇ *Post-architecture Model*

⁵Boehm verwendet in [Barry W. Boehm 1981] den Ausdruck Kilo Delivered Service Instructions, der zu KLOC identisch ist.

Das Application Composition Model ist für die Schätzung von Projekten gedacht, bei denen Entwicklungstools zum Einsatz kommen. Dieses Modell ist für diese Arbeit nicht relevant und wird daher nicht weiter erläutert.

Das Early Design Model ist für die Planungsphase der Softwareentwicklung gedacht und kann daher auf die Modellierungsphase innerhalb des BPM Lebenszyklus übertragen werden (vgl. Tabelle 2.2). Aufgrund dieser möglichen Zuordnung eignet sich dieses Modell für die weitere Untersuchung

Das Post-architecture Model ist für die Ermittlung von Wartungsaufwänden gedacht, und wird angewendet, wenn schon mit der Programmierung der Software begonnen wurde.

Sowohl dem Early Design Model als auch dem Post-architecture Model (Nachfolgend auch COCOMO II Stufe 2, respektive Stufe 3 genannt) liegt die gleiche Berechnungsart zugrunde. Wie bei COCOMO werden auch hier KLOC als Eingangsgröße verwendet. Allerdings werden die exponentiellen Faktoren (vorher n) als Skalenfaktoren und der lineare Faktor (vorher m) als Kostentreiber bezeichnet. Zudem führt [B. Boehm und Harrowitz 2000] einen Faktor A ein, der auf Erfahrungswerten beruht und initial von Boehm mit 2,94 vorgeschlagen wird.

Die Formel zur Berechnung lautet dann wie folgt:

$$PE = A \cdot Size^E \cdot \prod_{i=1}^{17} EM_i$$

Da sich zwar das Early Design Model für die weitere Verwendung empfiehlt, werden dennoch alle Skalenfaktoren und Kostentreiber aus dem Post-architecture Model aufgeführt und erläutert. Diese sind ebenfalls im Early Model enthalten. Zudem kann somit später auch ein Ausblick auf mögliche Faktoren zur Schätzung des Wartungsaufwandes gegeben werden. Allerdings sind Schätzungen für die Ausführungs- und Optimierungsphasen des BPM-Lebenszyklus nicht Bestandteil dieser Arbeit und werden daher auch nicht weiter erläutert.

Skalenfaktoren:

Die Skalenfaktoren fließen exponentiell in die Aufwandsberechnung mit ein, wie die folgende Formel zeigt:

$$E = B + 0,01 \cdot \sum_{i=1}^5 SF_i$$

Dabei ist B eine Konstante, die von Boehm empirisch zu 0,91 bestimmt wurde. SF_i sind die fünf Skalenfaktoren, beschrieben in Abbildung 4.2. Der Einfluss auf die Schätzung jeder dieser Faktoren kann mit VL („very low“), L („low“), N („normal“), H („high“), VH („very high“) und XH („extreme high“) bewertet werden. Hierbei entspricht „extreme high“ der höchsten Integrationsstufe des jeweiligen Faktors und wird daher immer mit „o“ bewertet, wodurch sich der Aufwand nicht verändert.

SF_i	COCOMO-Kürzel	Scale Factor	Gewichtung					
		Beschreibung	VL	L	N	H	VH	XH
SF_1	PREC	<u>Precedentedness / Vertrautheit</u> Ähnlichkeit der Fachlichkeit zu bereits existierenden Systemen? Faktoren, die zu beachten sind: Generelles Verständnis der Zielsetzung, Erfahrung mit der Funktion ähnlicher Systeme, involvierte Hardware oder betriebliche Funktionalität ist noch in Entwicklung, innovative Ablaufsteuerungen oder Algorithmen.	6,2	4,96	3,72	2,48	1,24	0
SF_2	FLEX	<u>Development Flexibility / Flexibilität der Entwicklung</u> Wie flexibel kann entwickelt werden bezüglich technischer Anforderungen? Faktoren, die zu beachten sind: Konformität der Software mit technischen Anforderungen und Spezifikationen von Schnittstellen.	5,07	4,05	3,04	2,03	1,01	0
SF_3	RESL	<u>Architecture/Risk Resolution / Architektur und Risikoausschluss</u> Wie gründlich war das Review der Grobspezifikation, und wie gut sind Punkte bekannt, die mit einem Risiko belastet sind? Faktoren, die zu beachten sind: Identifikation mögliche Risiken, stimmiger Budget- und Zeitplan, Aufwand zur Erstellung einer technische Architektur, Unterstützung durch Tools für Design, Sicherheitsfaktor für Technologie und Hardware.	7,07	5,65	4,24	2,83	1,41	0
SF_4	TEAM	<u>Team Cohesion / Teaminteraktion</u> Wie gut funktioniert das Team? Bewertet den Zusammenhalt und die Kooperation aller beteiligten Stakeholder (Auftraggeber, Anwender, Entwickler, Projektmanagement, usw.).	5,48	4,38	3,29	2,19	1,1	0
SF_5	PMAT	<u>Process Maturity / Prozessreife</u> Bewertet die Reife des Entwicklungsprozesses, die Laufzeiten der Prozesse und die Produktivität.]	7,8	6,24	4,68	3,12	1,56	0

Abbildung 4.2: Übersicht Skalenfaktoren COCOMO II – Quelle: [Frohnhoff 2009]

Aufgrund der vorgegebenen unkalibrierten Skalenwerte aus Abbildung 4.2 kann der Skalenfaktor E Werte zwischen 0,91 (alle sind mit „extreme high“ bewertet) und 1,266 (alle sind mit „very low“ bewertet) annehmen.

Kostentreiber:

Wie die Skalenfaktoren werden auch die Kostentreiber mit der Skala von „very low“ bis „extreme high“ bezeichnet. Abbildung 4.3 gibt eine Übersicht über die einzelnen Kostentreiber und deren jeweiligen Werte. Diese liegen unkalibriert zwischen 0,71 und 1,74. Niedrige Werte stehen für eine hohe Produktivität und damit geringeren Aufwand. Der Wert des gesamten Effort Multipliers kann daher zwischen 0,0569 und 115,58 liegen [Frohnhoff 2009].

Eine genauere Erläuterung der Skalenfaktoren und Kostentreiber aus den Abbildungen 4.2 und 4.3 wird in Kapitel 5.3.3 vorgenommen.

Bereich	EM_i COCO- MO- Kürzel	<u>Effort Multiplier</u> Beschreibung	Gewichtung					
			VL	L	N	H	VH	XH
Produkt	EM_1 RELY	<u>Required Software Reliability / Sicherheit</u> Wie zuverlässig muss die Software sein? Wenn die Auswirkung eines Fehlers nur leichte Unannehmlichkeiten bedeutet, ist RELY niedrig, bei Gefahr für Menschen hoch.	0,82	0,92	1	1,1	1,26	-
	EM_2 DATA	<u>Database Size / Datenbankgröße</u> Wie groß ist die Datenbank in Bytes/Programm Lines of Code? Ein größerer Datenbestand verursacht aufwändigere Testdaten.	-	0,9	1	1,14	1,28	-
	EM_3 CPLX	<u>Product Complexity / Komplexität der Anwendung</u> Gewichteter Mittelwert aus der Bewertung der Funktionen: Anwenderschnittstellen-, Berechnungs-, Kontroll-, Datenmanagement- und hardwareabhängige Funktionen.	0,73	0,87	1	1,17	1,34	1,74
	EM_4 RUSE	<u>Developed for Reusability / Wiederverwendbarkeit</u> Wie hoch sind die Anforderungen an die Wiederverwendbarkeit des Codes im selben System oder als Framework in anderen Projekten?	-	0,95	1	1,07	1,15	1,24
	EM_5 DOCU	<u>Documentation Match to Life-Cycle Needs / Anforderungen an Dokumentation</u> Wie aufwändig ist die Dokumentation (auch innerhalb der einzelnen Projektphasen) für uns bzw. als Lieferteilung für den Kunden?	0,81	0,91	1	1,11	1,23	-
Plattform	EM_6 TIME	<u>Execution Time Constraint / Auslastung d. Rechenzeit</u> Wie stark ist die Rechenleistung auf der Zielumgebung ausgelastet?	-	-	1	1,11	1,29	1,63
	EM_7 STOR	<u>Main Storage Constraint / Auslastung der Speicherkapazität</u> Wie stark ist das Storage-System auf der Zielumgebung ausgelastet?	-	-	1	1,05	1,17	1,46
	EM_8 PVOL	<u>Platform Volatility / Häufigkeit der Plattformwechsel</u> Wie oft wird die Hard- und Software (auch Datenbank oder Middleware) ausgetauscht?	-	0,87	1	1,15	1,3	-
Mitarbeiter	EM_9 ACAP	<u>Analyst Capability / Leistung des Chefdesigners</u> Wie hoch ist das Leistungsvermögen der Analysten (hierzu zählt nicht die Erfahrung der Chefdesigner, diese ist in APEX enthalten)?	1,42	1,19	1	0,85	0,71	-
	EM_{10} PCAP	<u>Programmer Capability / Leistung der Programmierer</u> Wie hoch ist das Leistungsvermögen der Programmierer im Team (hierzu zählt nicht die Erfahrung der Programmierer, diese ist in APEX enthalten)?	1,34	1,15	1	0,88	0,76	-
	EM_{11} PCON	<u>Personal Continuity / Kontinuität der Mitarbeiter</u> Wie hoch ist die Kontinuität der Mitarbeiter im Projekt?	1,29	1,12	1	0,9	0,81	-
	EM_{12} APEX	<u>Application Experience / Erfahrung über die Anwendung</u> Wie viel Erfahrung hat das Team (Analysten und Programmierer) mit der Fachlichkeit der zu entwickelnden Software?	1,22	1,1	1	0,88	0,81	-
	EM_{13} PLEX	<u>Platform Experience / Erfahrung mit der Plattform</u> Wie viel Erfahrung hat das Team über die zu nutzende Plattform und deren Middleware?	1,19	1,09	1	0,91	0,85	-
	EM_{14} LTEX	<u>Language and Tool Experience / Erfahrung mit Programmiersprache und Tools</u> Wie viel Erfahrung hat das Team mit der Programmiersprache und den Werkzeugen für Analyse und Darstellung von Anforderungen und Entwürfen, Konfigurationsmanagement, Dokumentationsgenerierung, Bibliotheksverwaltung, Programmstil und Formatierung?	1,2	1,09	1	0,91	0,84	-
Projekt	EM_{15} TOOL	<u>Use of Software Tools / Tool-Einsatz</u> Wie gut ist die Unterstützung der Entwickler durch Tools, und wie gut sind diese in die Prozesse eingebunden?	1,17	1,09	1	0,9	0,78	-
	EM_{16} SITE	<u>Multisite Development / Verteiltes Arbeiten</u> Wird standortübergreifend entwickelt? Welche Möglichkeiten zur Kommunikation stehen zur Verfügung? Bei verteilten Teams mit geringen oder aufwändigen Möglichkeiten zur Kommunikation wird SITE mit niedrig eingestuft, bei Arbeiten an einem Standort mit hoch.	1,22	1,09	1	0,93	0,86	0,8
	EM_{17} SCED	<u>Required Development Schedule / Zeitplan</u> Steht ausreichend Zeit für die Projektabwicklung zur Verfügung, wird SCED mit hoch eingestuft, bei Zeitdruck mit niedrig.	1,43	1,14	1	1	1	-

Abbildung 4.3: Übersicht Kostentreiber COCOMO II – Quelle: [Frohnhoff 2009]

4.2 Auswahl der Faktoren zur Schätzung

In den drei vorgestellten Verfahren werden jeweils Eingabegrößen als Basis für die Schätzung verwendet. Bei UseCase und der Function-Point Methode sind dies jeweils Punkte, die dann über weitere Gewichtung und unter der Verwendung einer Erfahrungstabelle indirekt oder direkt in den Aufwand umgewandelt werden [Frohnhoff 2009].

COCOMO II verwendet als Eingabegröße Lines of Code (LOC). Hierbei wird der Begriff SLOC (Single Line of Code) für eine einzelne Zeile Programmcode und der Begriff KSLOC⁶ für jeweils 1000 Zeilen verwendet.

Die LOC eines Prozessmodells lassen sich schwer schätzen und hängen auch von der verwendeten Modellierungssprache ab (vgl. Kapitel 1.5). Insbesondere kann auch keine qualifizierte Aussage über die LOC des Prozessmodells getroffen werden, wenn noch keine Prozessdokumentation vorliegt und der Prozess zunächst dokumentiert werden muss (vgl. Kapitel 3.4). Somit müssen andere Methoden gewählt werden, um die Eingabegrößen für das Schätzmodell zu ermitteln.

In diesem Kapitel geht es um die Identifikation möglicher Eingabegrößen und weitere Faktoren. Eine genaue Einordnung und Adaption im Hinblick auf die zu entwickelnde Schätzmethode folgt in Kapitel 5.

Eingabegrößen

Wie bereits erwähnt ist es für die Prozessmodellierung nicht möglich, die LOC zu schätzen. Eine Möglichkeit wäre die Zählung von Punkten, die dann über ein spezifisches Verfahren in Abhängigkeit von der gewählten Modellierungssprache in LOC durch die Multiplikation mit einem Faktor umgerechnet werden können. Man kann graphisch oder textuell modellieren – in beiden Fällen werden XML-ähnliche Dateien erzeugt deren Quelltext für jedes Modellierungselement eine gewisse Anzahl an Zeilen vorweist [Milutinovic 2014; *BPEL 2.0 Specification*; *BPMN 2.0 Specification*]. Für die hier vorgestellten Schätzmethoden wären dann sowohl Punkte, als auch LOC als Eingabegröße verfügbar.

Um die Punkte zu ermitteln, wäre hierbei die simpelste Methode die visuellen Elemente zu zählen, die im Rahmen der Modellierung entstehen und diese Anzahl aufzusummieren. Diese Summe mit einem Faktor multipliziert würde dann die Anzahl von LOC ergeben.

Bei Verwendung der BPMN 2.0 Notation kommen dabei vorwiegend:

- ◇ Aktivitäten und Transaktionen
- ◇ Ereignisse
- ◇ Gateways
- ◇ Pools

⁶Von [B. Boehm und Harrowitz 2000] verwendete Bezeichnung. Teilweise wird auch KLOC verwendet.

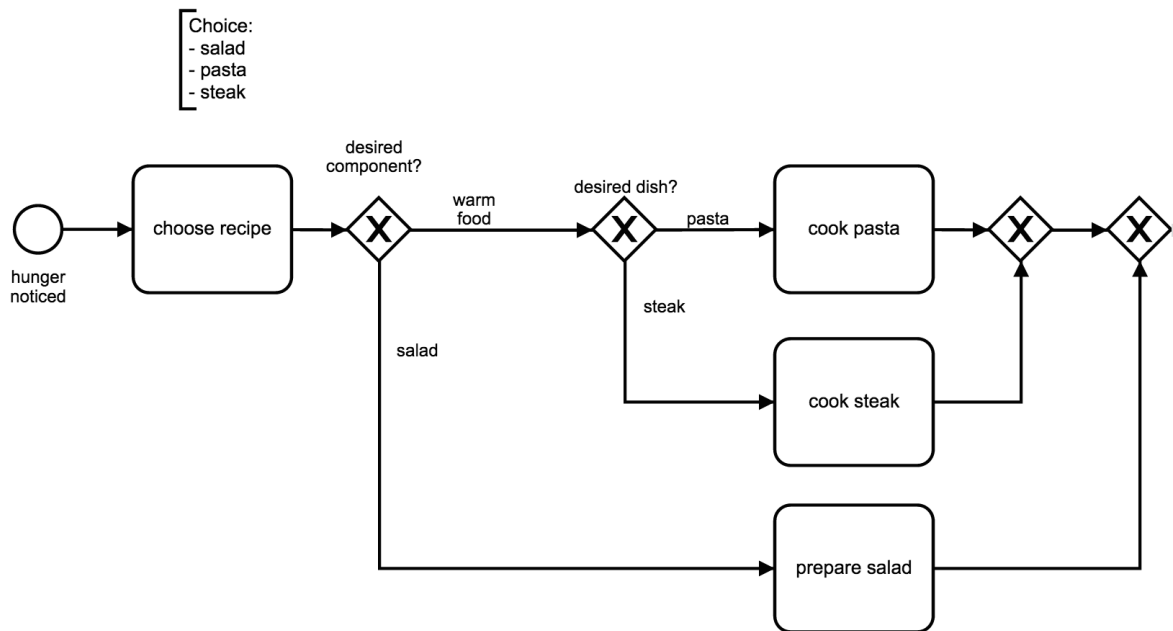


Abbildung 4.4: Beispielprozess Abendessen – Symmetrische Modellierung mit Gateways –
Quelle: [Freund und Rücker 2012]

- ◇ Lanes
- ◇ Datenobjekte
- ◇ Ein/Ausgehende Kommunikation

zum Einsatz. ([Freund und Rücker 2012] – Beispielfälle zusammengefasst). Während Aktivitäten und Transaktionen, sowie beteiligte Akteure in Form von Pools und Lanes eines Prozessmodells gut geschätzt werden können, ist es ungleich schwieriger, die Anzahl von Gateways vorab zu kennen.

Wie das folgende Beispiel zur „Auswahl des Abendessens“ aus Freund12 zeigt gibt es auch mehrere Möglichkeiten einen gewünschten Prozessablauf zu modellieren.

Abbildung 4.4 zeigt eine Modellierung für die Frage nach dem Abendessen. Dabei werden vier Gateways eingesetzt. Dem Prozess nach, kann man entweder Salat *oder* Pasta *oder* Steak auswählen. Es ist allerdings nicht möglich, Salat *und* Steak auszuwählen, was an den exklusiven Gateways liegt.

Möchte man den Prozess noch etwas flexibler gestalten und die Auswahl des Abendessens auf eine der drei Möglichkeiten gestalten:

- ◇ nur einen Salat

- ◇ einen Salat und „etwas Ordentliches“ wie Pasta oder Steak
- ◇ nur „etwas Ordentliches“

könnte der Prozess wie auf Abbildung 4.5 aussehen.

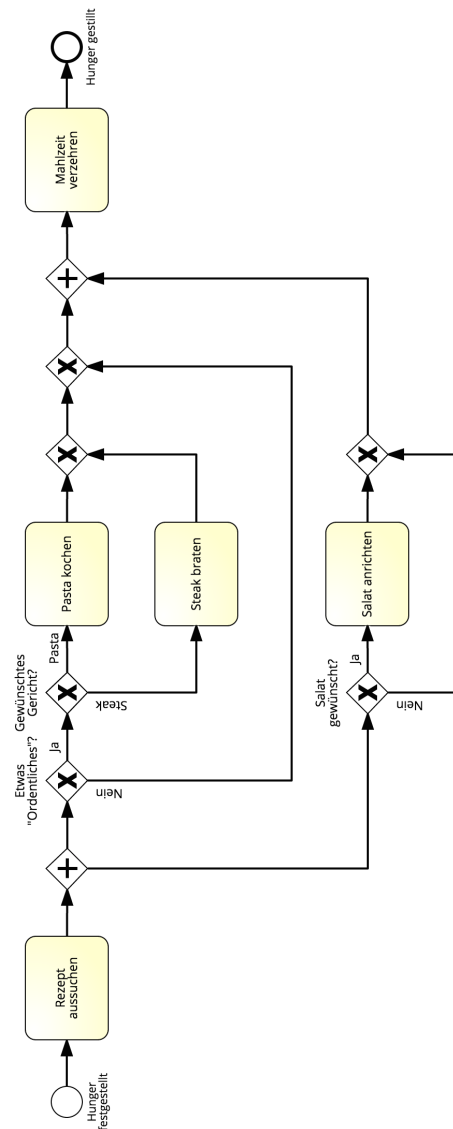


Abbildung 4.5: Beispielprozess Abendessen – Modellierung mit mehr Gateways – Quelle: [Freund und Rücker 2012]

Bei gleichbleibender Anzahl der Aktivitäten des Prozesses gibt es nun statt vier acht Gateways. Man könnte hier den Prozess mit dem Einsatz von datenbasierten Gateways vereinfachen und damit die zu modellierenden Elemente reduzieren, allerdings könnte dadurch

die Semantik des Modells verändert werden. Auch die von [Mendling, Reijers und Aalst 2010] vorgeschlagenen „Seven Principles Modeling Guidelines“ (7PMG) empfehlen, so wenig Elemente wie nötig zu modellieren und den Einsatz von Gateways auf AND und XOR zu beschränken. Dadurch wird eine Abschätzung der Anzahl von Gateways im Vorfeld sehr schwierig.

Da auch für den Fall einer fehlenden Prozessdokumentation eine Schätzung erstellbar sein soll, empfiehlt es sich hierbei, Eingabegrößen zu wählen, die durch eine Expertenanalyse oder einen Workshop mit den am Prozess beteiligten Personen (vgl. Kapitel 3), ermittelt werden können. Durch die Dokumentation oder Diskussion des Prozessablaufs lassen sich zumindest die dem Ablauf zugrundeliegenden Aktivitäten und Ereignisse sowie Pools und Lanes ermitteln. Ohne den genauen Kontrollfluss und die Verteilung von möglichen Gateways zu kennen lässt sich so zumindest eine Aussage über die Anzahl der Aktivitäten und Ereignisse sowie Pools und Lanes treffen. Diese Kennzahlen können daher für die Erstellung einer Schätzung verwendet werden und dabei selbst geschätzt werden.

Gewichtungsfaktoren

Neben den Eingabegrößen gibt es noch weitere Faktoren, die berücksichtigt werden müssen: Nicht jeder Prozess mit der gleichen Anzahl an Aktivitäten und Ereignissen ist gleich. Diese können eine zum Teil sehr unterschiedliche Komplexität, Ausführungsdauer und Parallelität von Aktivitäten haben und bedeuten daher auch unterschiedlichen Aufwand für den Modellierer. Der erhöhte Aufwand liegt beispielsweise daran, dass durch erhöhte Komplexität des Prozesses vom Modellierer zwar semantisch korrekte Prozessmodelle erstellt werden, diese allerdings zum Zeitpunkt der Ausführung einen Laufzeitfehler verursachen können [Freund und Rücker 2012]. Dies ist gleichzeitig ein Beispiel für „softe“ Faktoren wie die Erfahrungheit des Modellierers. Ein erfahrenerer Modellierer kennt die Einschränkungen der gewählten Modellierungssprache und kann mit weniger Aufwand diese „Falle“ vermeiden.

Ebenfalls sind Skalenfaktoren, wie die Teamzusammengehörigkeit, die Neuartigkeit des zu modellierenden Ablaufs oder der Reifegrad des Modellierungsprozesses an sich zu berücksichtigen und in die Schätzung mit einzubringen, da auch diese den Aufwand der Modellierung beeinflussen.

Angelehnt an die vier P's des Marketing [Schmelzer und Sesselmann 2008] „Product, Price, Placement and Promotion“ existieren in der Softwareentwicklung auch vier Kategorien für die Kostentreiber bei Erstellung von Software [*COCOMO II Model Definition Manual*]:

Produkt, Personal, Plattform, Projekt.

Nachfolgend einige kurze Beispiele für diese Kategorien:

- ◇ Ein Kostentreiber aus der Kategorie *Produkt* wäre die Größe des zu modellierenden Prozesses.
- ◇ Die Anzahl der an der Modellierung beteiligten Analysten und Programmierer wäre ein Beispiel für den Bereich *Personal*.

- ◇ Zeit- und Größenbeschränkungen bei der Modellierung sind Beispiele für *Plattform-Treiber*.
- ◇ *Projektkosten* könnten sich beispielsweise durch die (gleichzeitige) Modellierung an verschiedenen Standorten erhöhen.

Im Bezug auf die Modellierung von Geschäftsprozessen sind diese Kostentreiber zu erwarten und sollten daher berücksichtigt werden.

Die Schätzmethode sollte es daher ermöglichen, neben Einflussgrößen auch die Gewichtungsfaktoren zu berücksichtigen und in unterschiedlicher Abstufung einzubringen. Eine genaue Auswahl der Faktoren wird in Kapitel 5.2 getroffen.

4.3 Weitere Faktoren und Einflussgrößen

Die beschriebenen Schätzmethoden besitzen einige Faktoren, die sich direkt auf die Geschäftsprozessmodellierung (GPM) übertragen lassen. So sind in allen vier Bereichen Produkt, Personal, Plattform und Projekt aufgrund der Vergleichbarkeit des BPM Lebenszyklus mit den Phasen der Softwareentwicklung, die jeweiligen Faktoren auch für die GPM anwendbar. Eine erschöpfende Untersuchung oben genannter Faktoren erfolgt in Kapitel 5.3.3.

Dennoch gibt es einige Einflussgrößen und Faktoren die im Bezug auf die GPM nicht betrachtet werden:

Bereich Produkt

Die Komplexität des Prozessmodells lässt sich nicht allein durch die Komplexität der einzelnen Objekte oder deren Anzahl ausdrücken. Die Komplexität eines Prozesses ist vielschichtig und kann durch verschiedene Metriken beschrieben werden [Gruhn und Laue 2006]. Auch die Komplexitätsabschätzung in COCOMO ist nicht direkt auf BPM adaptierbar (vgl. Kapitel 5.3.3).

Bereich Personal

Der an einem Prozess beteiligte Personenkreis unterscheidet sich von dem eines Softwareentwicklungsprojektes. In BPM sind „Chief Process Officer“ (CPO), „Process Owner“ (deutsch: Prozessmanager), Prozessmitarbeiter, Prozessexperten, Projektleiter, Prozessberater, Prozess-/Workflowmodellierer und Prozessauditoren beteiligt [Freund und Rücker 2012; Gadatsch 2013]. Eine zentrale Rolle spielt hier die Rolle des CPO. Dieser ist in einer Organisation unter anderem für die prozessorientierte Gestaltung der Organisation [Gadatsch 2013] verantwortlich. Er arbeitet eng mit dem „Chief Information Officer“ (CIO) zusammen und agiert somit als Schnittstelle zwischen dem fachlichen und technischen Bereich.

Bereich Projekt

Wie bei „Model Driven Architecture“ (MDA) in der Softwareentwicklung, werden bei BPM mit Hilfe von Tools Modelle erstellt. Bei MDA werden von Programmiersprachen

unabhängige Modellbeschreibungen für Software erstellt, die dann in die gewünschte Sprache übersetzt werden können [Kleppe, Warmer und Bast 2003].

Dabei besitzen die Werkzeuge zur Modellierung teilweise bereits Beispieldatenbanken (Repositories) mit Modellen und erleichtern hierbei die Erstellung. Eine Übersicht zu Modellierungswerkzeugen und vorhandenen Beispieldatenbanken bietet [*Comparison of Business Process Modeling Notation tools*]⁷. Dieser Aspekt wurde bei den untersuchten Schätzmodellen bisher nicht betrachtet.

Bereich Skalenfaktoren

Wichtigster Punkt hierbei ist die Unterscheidung aus Kapitel 3 zwischen fehlender und vorhandener Prozessdokumentation. Bei vorhandener Dokumentation sind die Einflussgrößen für die Schätzung vorhanden und können daher direkt verwendet werden. Sind diese nicht vorhanden, müssen sie zunächst geschätzt werden. Abhängig von der Erstellung dieser Schätzung und dem dafür verwendeten Verfahren entsteht zusätzlicher Aufwand, der berücksichtigt werden muss.

⁷https://en.wikipedia.org/wiki/Comparison_of_Business_Process_Modeling_Notation_tools

Kapitel 5

Einführung BPM Schätzmethode

Im vorherigen Kapitel wurden Methoden zur Aufwandsschätzung von Softwareprojekten und deren mögliche Verwendung innerhalb des Kontextes von BPM erläutert. In diesem Kapitel wird eine spezielle Methode ausgewählt und konkret auf BPM angepasst. Dabei wird die Adaption der Schätzmethode detailliert beschrieben.

5.1 Auswahl der Methode

Alle drei Schätzmethoden aus Kapitel 4 eignen sich um den Aufwand während der Modellierungsphase im BPM-Lebenszyklus zu schätzen. Da bei diesen Verfahren neben den gewichteten Einflussgrößen auch weitere Faktoren wie Qualität, Komplexität, Wiederverwendbarkeit etc. einfließen, kann die Schätzung innerhalb einer Organisation durch den Aufbau von Erfahrungswerten immer genauer werden. Ferner ist es bei diesen Verfahren möglich, neue Faktoren einzuführen, welche für eine BPM-Schätzung notwendig sind.

Der Hauptkritikpunkt an der UseCasePoints Methode ist die Ungenauigkeit der Schätzung und das Fehlen von vergleichbarer Praxiserfahrung [Frohnhoff 2009]. Durch den großen Interpretationsspielraum wegen des Fehlens einer Definition von Actors und der schwierigen Adaption von Transaktionen zu Aktivitäten bei der Durchführung, kann die Schätzung im Grundsatz gestört sein und falsche Ergebnisse liefern [Frohnhoff 2009]. Dadurch, dass die UseCasePoints Methode auf Function-Point basiert und dafür wesentlich mehr Erfahrungswerte vorliegen, wird keine weitere Betrachtung von UseCasePoints vollzogen.

Function-Point eignet sich nicht als gesamtheitliche Schätzmethode für die Modellierung, da die Faktorenliste mit entsprechenden Gewichten nicht vorhanden ist und auch aufgrund von fehlenden Realweltdaten im Rahmen dieser Arbeit nicht erstellt werden konnte. Daher sind große Streuungen bei den Ergebnissen und eine damit einhergehende Ungenauigkeit bei Verwendung dieser Methode zu erwarten. FP betrachtet auch nicht die Qualität des zu entwickelnden Produktes [Khatibi und Jawawi 2011]. Allerdings kann FP mit COCOMO II kombiniert werden, um die Eingabegröße LOC in COCOMO II über „Unadjusted Function Points“ (UFP) und die Auswahl der Modellierungssprache zu ermitteln. Diese Kombination der beiden Verfahren wird daher nachfolgend adaptiert und beschrieben.

5.2 BPM Function-Point

Function-Point ermittelt den Aufwand eines Projekts in Abhängigkeit von dessen Umfang und Schwierigkeitsgrad [Noth und Kretzschmar 1986].

Das zugrunde liegende Verfahren von Function-Point ist die Ermittlung von Function-Points, indem man die Ein- und Ausgabedaten, Datenbestände, Schnittstellen und Abfragen erfasst und mit Punkten bewertet [Frohnhoff 2009; Noth und Kretzschmar 1986]

Diese fünf genannten Kriterien sind Funktionen der Software, die im weitesten Sinne als „Geschäftsvorfälle“ beschrieben werden können. Dabei wird der Begriff Geschäftsvorfälle hier im Kontext mit der Abarbeitung einer Aktivität, wie der Eingabe von Daten, verwendet [Noth und Kretzschmar 1986]. Für die Schätzung von BPM ist der Begriff Geschäftsvorfälle überaus passend. Ein Prozess hat demnach sehr ähnliche Charakteristika; auch bei einem Prozessmodell gibt es Ein- und Ausgabedaten. Zwischendurch müssen die Daten und Prozesszustände innerhalb des BPMS (Oder auch DBMS WFMS) gespeichert werden [Freund und Rücker 2012]. Eine Aktivität innerhalb des Prozessmodells entspricht im weitesten Sinne einem Geschäftsvorfall. Ebenfalls ist es möglich, jederzeit den Zustand des laufenden Prozesses abzufragen oder mitzuteilen; so ist zum Beispiel die Benachrichtigung über den Versand eines Pakets bei einem Bestellvorgang im Internet eine Abfrage und würde mit einem Function-Point bewertet werden.

Während der Zählung der Function-Points werden die zugeteilten Kategorien auch in die drei Komplexitätsklassen „leicht“, „mittel“ und „komplex“ eingeteilt. Je nach Einteilung in die verschiedenen Klassen werden die jeweiligen Kategorien dann mit Gewichten von 3-15 multipliziert. Diese Gewichtung bedarf einer Validierung basierend auf Erfahrungswerten. Als Grundlage wurde die initiale Gewichtung für die Kategorien gewählt (vgl. Tabelle 4.1) [COCOMO II Model Definition Manual; Noth und Kretzschmar 1986].

Diese Werte für die Gewichtung sollten nach Projektabschluss, wie in Kap 4.1 beschrieben, angepasst werden, da mit steigender Projektanzahl auch eine höhere Genauigkeit der Schätzmethode erreicht wird.

Die Schritte zur Zählung der Function-Points sind wie folgt übertragbar:

Erster Schritt:

Eine Einordnung der Anforderungen in die fünf Kategorien kann für BPM wie folgt angepasst werden:

Dabei lassen sich die fünf Kategorien für BPM genauer beschreiben:

Ein Vorschlag zur Selektion der Komplexitätsgewichtung während der Zählung der Punkte in den jeweiligen Kategorien wird in Kapitel 5.3 vorgestellt.

Zweiter Schritt:

In diesem Schritt erfolgt die Einordnung der Komplexität der jeweiligen Funktionen innerhalb der zuvor definierten Kategorien in niedrig / mittel / hoch. Hierbei werden die einzelnen Punkte gewichtet und mit dem jeweiligen Faktor multipliziert.

Function-Point	BPM Function-Point	Komplexitätsgewichtung		
		Leicht	Mittel	Komplex
Eingabedaten	Eingabedaten (Message)	3	4	6
Abfragen	Aktivitäten und Ereignisse	4	5	7
Ausgabedaten	Ausgabedaten (Message)	3	4	6
Datenbestände	Datenobjekte	7	10	15
Schnittstellen	Pools und Lanes	5	7	10

Tabelle 5.1: Vergleichstabelle Function-Point und BPM

Kategorie	Beschreibung
Eingabedaten (Message)	Aktivitäten, die Daten empfangen (Message) Startzustand
Aktivitäten und Ereignisse	Aktivitäten und Ereignisse, die den Zustand des Prozesses beschreiben
Ausgabedaten (Message)	Aktivitäten, die Daten versenden (Message) Auch alle Endzustände
Datenobjekte	Datenobjekte, die verarbeitet werden
Pools und Lanes	Anzahl der Pools und Lanes

Tabelle 5.2: Beschreibung der FP Kategorien für BPM

An dieser Stelle kann nicht die Komplexität des zu erwartenden Prozessmodells im Ganzen geschätzt werden. Vielmehr geht es um die einzelnen zu modellierenden Objekte.

Nach Durchführung des ersten und zweiten Schritts erhält man die nicht gewichteten Function-Points (UFP). Diese können auch mit COCOMO II kombiniert werden und als Eingangsgröße dienen, sofern eine Bestimmung der LOC schwierig oder nicht möglich ist.

Dritter Schritt:

Hier folgt eine Gewichtung der einzelnen Kategorien mit Faktoren. Die Faktorenliste ist über die IFPUG zu beziehen – allerdings stand zum Zeitpunkt der Bearbeitung dieser Arbeit keine für BPM oder Modelliersprachen zur Verfügung. Dies ist eine Einschränkung für die weitere Verwendung dieses Verfahrens. Da man keinen Zugriff auf Realweltdaten und Beispielprojekte erhalten konnte, ist es nicht möglich gewesen, das Modell wie ursprünglich geplant zu evaluieren und eine Faktorenliste auszuarbeiten. Dennoch werden die weiteren Schritte dargestellt um das Modell vollständig zu beschreiben.

Vierter Schritt:

Die Errechnung von gewichteten Function-Points über die in Kapitel 4.1.1 beschriebene Tabelle der technischen Komplexitätsfaktoren passt nicht zum BPM Kontext. Andere Methoden wie Mark II Function-Point [Symons 1988] erweitern die Anzahl der Faktoren und

führen eine Anpassung durch. Hier kann auch eine Anpassung, wie in Tabelle 9 vorgeschlagen, vorgenommen werden. Hierbei werden wieder Punkte vergeben, die bei der gesamten Berechnung eine Gewichtung darstellen. Es würden den Faktoren jeweils Gewichte zugeteilt und summiert, wobei diese Summe bis zu 60 Punkte umfassen darf [Frohnhoff 2009; Wieczorrek und Mertens 2007].

Einflussfaktoren aus FP können ebenfalls wie in Tabelle 5.3 beschrieben angepasst werden:

Function-Point	BPM Function-Point
Quantität (wie LOC)	Summe der Modellentitäten
Qualität	Qualität / Genauigkeit des Modells
Komplexität	Komplexität des Prozessmodells
Erfahrung	Prozesserfahrung / BPM Erfahrung
Entwicklungstools	Modellierungstools
Programmiersprache	Modellierungssprache

Tabelle 5.3: Einflussfaktoren FP adaptiert aus [Wieczorrek und Mertens 2007]

Tabelle 5.3 gibt hierbei einen Vorschlag zur möglichen Adaption der Einflussfaktoren. Aufgrund des Verzichts auf die weitere Verwendung und Untersuchung dieser Schätzmethode wird hier auf die Begründung der Auswahl der Adaptionsfaktoren verzichtet.

Fünfter Schritt:

Im fünften Schritt werden die gewichteten Function-Points berechnet, um dann im letzten Schritt den Aufwand aus der IBM Aufwandstabelle abzulesen. Hier könnte man auch die Erfahrungstabelle selbst anlegen und nach Projektabschluss die Originaltabelle, die man von der IFPUG bezogen hat, aktualisieren. So würden sich bei jedem weiteren Projekt die Erfahrung und damit auch die Schätzung verbessern.

Zusammenfassung:

Durch die fehlende Möglichkeit, die Schritte drei bis fünf mit entsprechenden Umrechnungstabellen und weiteren Projektdaten zu validieren, kann das beschriebene Verfahren nach derzeitigem Untersuchungsstand nicht verwendet werden, um eine vollständige Schätzung zu erhalten. Dennoch ist BPM Function-Point ein valider Ansatz, welcher im Nachgang zu dieser Arbeit umfassender und genauer untersucht werden sollte. Allerdings können mit dem beschriebenen Verfahren bis einschließlich Schritt zwei die benötigten UFP, die für die Eingabe in COCOMO II benötigt werden, errechnet oder geschätzt werden.

5.3 BPM COCOMO

Aufgrund der sehr granularen Auswahlmöglichkeiten an Faktoren und Einflussgrößen für die Schätzung, mit der Möglichkeit neben der Eingangsgröße LOC auch Function-Points für die Schätzung zu verwenden, eignet sich COCOMO von den drei vorgestellten Verfahren ideal für eine Anpassung der Schätzmethode.

Im Folgenden wird ein Vorschlag erarbeitet, ein neues, für die Modellierung von Geschäftsprozessen angepasstes Schätzmodell, namens

Costructive Cost Model (COCOMO) for BPM: *BPM COCOMO*

zu entwickeln.

Dabei wird die Eingangsgröße analog zum Function-Point ermittelt, um eine virtuelle Einheit für die LOC zu erhalten. Die nicht angepassten FP, beschrieben aus dem vorherigen Abschnitt 5.2, können mit Hilfe von COCOMO II besser und granularer gewichtet werden. Die umfangreichen Einflussfaktoren der beiden COCOMO Phasenmodelle „Early Design Model“ und „Post-Architecture Model“ (auch als Stage 2 und 3 beschrieben) [B. Boehm und Harrowitz 2000] bieten umfassende Möglichkeiten zur Übertragung von Skalenfaktoren und Kostentreibern auf BPM COCOMO.

Abbildung 5.1 verdeutlicht den Ablauf während der Schätzung mit BPM COCOMO:

Nachfolgend werden alle Skalenfaktoren und Kostentreiber des „Post-Architecture Models“ ausführlich beschrieben. Für die Berechnung des Aufwandes wird allerdings das „Early Design Model“ verwendet, da dieses auch der Entwicklungsphase der Modellierung im BPM-Lebenszyklus entspricht. Daher werden einige Einflussfaktoren bei der Berechnung nicht berücksichtigt, da diese teilweise erst nach erfolgter Modellierung ermittelt werden können oder für die Modellierung selbst keine Relevanz besitzen. Eine genaue Beschreibung folgt in den Kapiteln 5.3.3 und 5.3.4.

5.3.1 Bestimmen der Eingangsgrößen

Um eine Aussage über die LOC des Prozessmodells zu erhalten, müssen diese zunächst ermittelt werden, da diese nicht direkt geschätzt werden können (siehe Kap. 4.2).

Eine direkte Bestimmung der LOC wäre die Zählung der Anzahl folgender Objekte:



Abbildung 5.1: Ablauf der der Schätzung mit BPM COCOMO

- ◇ Aktivitäten
- ◇ Ereignisse
- ◇ Nachrichtenflüsse
- ◇ Kontrollflüsse
- ◇ Gateways
- ◇ Datenobjekte
- ◇ Prozessbeteiligte (In BPMN 2.0 über Lanes und Pools)

Die jeweilige Summe der Objekte multipliziert mit der Anzahl von minimal notwendigen LOC zur Beschreibung, ergibt dann die LOC insgesamt und dient als ungefähre Schätzung [Gruhn und Laue 2006; Kluza und Nalepa 2012].

Hierbei wird allerdings nicht berücksichtigt, dass sich die Modellierung der Objekte unterschiedlich komplex darstellt. So kann es vorkommen, dass eine Aktivität mehrere Kontrollflüsse besitzt, was in einer höheren Anzahl an LOC resultieren würde. Ereignisse können an mehrere Bedingungen verknüpft sein und müssen daher auch umfangreicher beschrieben werden als in der Spezifikation intendiert.

Darüber hinaus sind nicht alle Objekte wie bereits in Kapitel 4.2 erwähnt zum Zeitpunkt der Modellierung vorhanden und zählbar.

Eine Kategorisierung und Bewertung dieser Eingabegrößen über deren Anzahl erscheint daher notwendig, um einen Querschnitt über die Komplexität der einzelnen Objekte zu legen. Wie in Kapitel 5.2 beschrieben können Unadjusted Function Points (UFP) als Eingangsgröße für COCOMO II verwendet werden.

In Kapitel 5.2 wurde die Möglichkeit angeführt, die Function Point Methode für die Aufwandsschätzung zu verwenden. Durch fehlende Erfahrungswerte konnte die Methode nicht weiter entwickelt werden. Dennoch kann der Vorschlag zur Ermittlung der UFP für BPM übernommen werden, um dann über einen Faktor in Abhängigkeit von der gewählten Modellierungssprache die Eingangsgröße LOC zu erhalten.

Ermittlung von (Business Model Points) BMP:

Wie in 5.2 vorgeschlagen, können die UFP durch folgende Adaption ermittelt werden (vgl. Tabelle 5.4):

Durch Zählung der Objekte und Einordnung in die Komplexitätsklassen können die Business Model Points (BMP) ermittelt werden.

Die Einordnung in die drei Komplexitätsklassen erfolgt nach folgenden Kriterien:

Nachdem man die Summe der BMP durch Addition aller in den fünf jeweiligen Kategorien gezählter Punkte ermittelt hat, werden diese in LOC umgerechnet (vgl. Kapitel 5.3.2).

BPM Function-Point	Komplexitätsgewichtung		
	Leicht	Mittel	Komplex
Eingabedaten (Message)	3	4	6
Aktivitäten und Ereignisse	4	5	7
Ausgabedaten (Message)	3	4	6
Datenobjekte (Artefakte)	7	10	15
Pools und Lanes	5	7	10

Tabelle 5.4: Adaptionstabelle UFP für BPM

Kategorie	Eingaben	Aktivitäten / Ereignisse	Ausgaben	Datenobjekte	Pools / Lanes
einfach	Einfach zu erfassen / in Sequence	Einfacher Flow	Einfach zu erfassen / in Sequence	Einfache Verwendung, kurze Speicherdauer	1-3 Lanes / Pool
mittel	Zu verschiedenen Zeitpunkten	mittlerer Flow / viele JoinConditions / viele Tokens	Zu verschiedenen Zeitpunkten	Wiederholte Verwendung / Lange Speicherdauer / Kommunikation	3-10 Lanes / Pool
komplex	Zu verschiedenen Zeitpunkten / Abhängigkeiten	Komplexer Flow / Komplexe GWs	Zu verschiedenen Zeitpunkten / Abhängigkeiten	Generierung von Ereignissen / viel Kommunikation	>10 Lanes / Pool

Abbildung 5.2: Übersicht der Komplexitätsmetriken für BPM adaptiert aus [Gruhn und Laue 2006])

5.3.2 Ermitteln der Lines of Code (LOC)

Um für die Berechnungen in BPM COCOMO die Eingangsgröße verwenden zu können, müssen die ermittelten BMP in LOC umgerechnet werden.

Hierzu wurden die XML-Beschreibungen aus der Spezifikation¹ von Prozessmodellen untersucht und dabei die Mindestanzahl an LOC für die Beschreibung eines Objekts gezählt.

¹<http://www.omg.org/spec/BPMN/2.0/>

Listing 5.1 BPMN XML-Beschreibung für das Starterereignis [*BPMN 2.0 Specification*]

```

1 <xsd:element name="startEvent" type="tStartEvent" substitutionGroup="
  flowElement"/>
2 <xsd:complexType name="tStartEvent">
3   <xsd:complexContent>
4     <xsd:extension base="tCatchEvent">
5       <xsd:attribute name="isInterrupting" type="xsd:boolean" default="
        true"/>
6     </xsd:extension>
7   </xsd:complexContent>
8 </xsd:complexType>

```

Je nach verwendetem Werkzeug variiert dabei die Anzahl der Zeilen. Daher wurde die BPMN 2.0 Spezifikation der OMG zur Zählung verwendet.

Objekt	XML-Schema Zeilen
Aktivität	6
Artefakt	6
Messages	10
Events	8
Pools	7
Lanes	7
Durchschnitt	7 (gerundet)

Tabelle 5.5: Ermittlung von Codezeilen für BPMN Objekte

Es wurden nur die für BMP relevanten Objekte untersucht und dabei die Mindestanforderung des Schemas betrachtet. Da die BMP durch Komplexitätsklassen gewichtet wurden, fließen hier auch die optionalen Attribute mit ein, die modelliert werden könnten. Ebenfalls werden hier die Schema-Spezifikationen für den Aufbau des gesamten Prozesses nicht betrachtet. Allerdings ist die Anzahl der Zeilen dafür zu vernachlässigen.

Bei der Evaluation sollte überprüft werden, ob die Größenangabe korrekt gewesen ist, oder ob man bereits hier eingreifen und eine Anpassung der Gewichte und Umrechnungsschlüssel vornehmen muss.

Listing 5.1 zeigt die XML-Beschreibung für das Starterereignis.

Listing 5.2 zeigt die XML-Beschreibung für eine Aktivität.

Zur Verifikation der Ergebnisse wurde diese Untersuchung auch für BPEL vorgenommen:

Hierbei wurde ein ähnlicher Wert ermittelt, der plausibel erscheint. Pressman⁰⁵ schlägt für graphische Sprachen einen Wert von „4“ LOC pro FP vor.

Listing 5.2 BPMN XML-Beschreibung für eine Aktivität [*BPMN 2.0 Specification*]

```

1 <xsd:element name="task" type="tTask"/>
2 <xsd:complexType name="tTask">
3   <xsd:complexContent>
4     <xsd:extension base="tActivity"/>
5   </xsd:complexContent>
6 </xsd:complexType>

```

Objekt	XML-Schema Zeilen
Aktivität	10
Variables	4
Messages	15
Events	15
Correlations	10
PartnerLinks	10
Durchschnitt	11 (gerundet)

Tabelle 5.6: Ermittlung von Codezeilen für BPEL Objekte

Dadurch ergibt sich folgende Umrechnungstabelle:

Modellierungssprache	LOC pro BMP
BPMN 2.0	7
WS-BPEL	11
Visuelle Modelle	4

Tabelle 5.7: Umrechnungstabelle BMP zu LOC**5.3.3 Selektion der Faktoren**

Dieser Abschnitt betrachtet die Relevanz von Skalenfaktoren und Kostentreibern für das BPM COCOMO Schätzmodell. Hier wird überprüft ob diese Faktoren auch im Fall der Geschäftsprozessmodellierung einen Einfluss haben. Eine Gesamtübersicht bietet Tabelle X. Die jeweiligen Einflussfaktoren werden in diesem Abschnitt eingehend beschrieben. Eine Adaption dieser wird in Kapitel 5.3.4 vorgenommen. Die ausführliche Beschreibung der Faktoren basiert auf [B. Boehm und Harrowitz 2000; Frohnhoff 2009] und wird daher nicht in jedem Absatz zitiert.

a) Skalenfaktoren*1. Precedentedness (PREC)*

Precedentedness wird in diesem Fachbezug mit „Grad der Nicht-Neuartigkeit“ übersetzt [Dowie 2008]. Mit diesem Faktor soll ausgedrückt werden, ob man bereits Erfahrungswerte im Rahmen des zu bearbeitenden Projektes besitzt. [Frohnhoff 2009] übersetzt den Faktor mit „Ähnlichkeit der Fachlichkeit zu bereits existierenden Systemen“.

So beschreibt dieser Faktor den Grad der Kenntnis über Projektziele, den voraussichtlichen Aufwand sich in neue Hardware und Betriebsabläufe einarbeiten zu müssen, sowie den Grad an neu zu entwickelnden Datenverarbeitungsarchitekturen und Algorithmen. [COCOMO II Model Definition Manual; B. Boehm und Harrowitz 2000; Frohnhoff 2009]. Ein Beispiel für einen hohen Grad an Nicht-Neuartigkeit wäre die Implementierung einer Textverarbeitungssoftware. Damit verglichen hat die erste Implementierung eines Social-Media Netzwerkes wie Facebook² hohe Neuartigkeit und daher einen geringen Grad an Precedentedness.

Im Bezug auf die Modellierung eines Geschäftsprozessmodells (GPM) kann dieser Faktor als Einflussgröße betrachtet werden, da hiermit beschrieben werden kann, inwiefern in der Organisation BPM bereits eingeführt wurde. Darüber hinaus lässt sich die Nicht-Neuartigkeit des GPM selbst beschreiben. So wie Softwareprojekte auf Model Driven Architecture (MDA) (mdao) basieren können, kann im Bereich der Modellierung von GPM ebenfalls auf Vorlagen zur Weiterverwendung und Anpassung innerhalb der Werkzeuge zurückgegriffen werden [Comparison of Business Process Modeling Notation tools].

Ferner kann mit diesem Faktor auch die Projektnatur beschrieben werden. Ein BPR-Projekt hat einen anderen Grad an Neuartigkeit als die initiale Einführung von BPM im Unternehmen [Freund und Rücker 2012; Posluschny 2012].

In Kapitel 5.3.3 wird eine genaue Abgrenzung zwischen den beiden Faktoren Process Maturity (PMAT) und der Precedentedness (PREC) vorgenommen. Das BPM Maturitätslevel der Organisation beschreibt gleichzeitig nicht die direkten Fähigkeiten der (fachlichen und technischen) Geschäftsprozessmodellierung [Allweyer 2007; Fisher 2004].

2. Development Flexibility (FLEX)

Dieser Faktor beschreibt den Grad der Einhaltung von gegebenen Anforderungen und Spezifikationen, welche die Software erfüllen muss. Bei restriktiveren Vorgaben ist die Flexibilität geringer, bei freien höher. Zudem wird auch berücksichtigt, dass es eine zusätzliche Vergütung bei Fertigstellung vor dem Termin geben kann.

Da bei der Modellierung kreative Lösungen und verschiedene Ansätze zur Erstellung eines Modells einfließen - unter anderem auch mehrere Lösungen für ein Modell in BPMN, die semantisch identisch sind [Freund und Rücker 2012] - ist der Flexibilitätsfaktor ebenfalls zu selektieren.

²<http://www.facebook.com>

3. Architecture / Risk Resolution (RESL)

Dieser Faktor kombiniert zwei Faktoren aus ADA COCOMO [[COCOMO II Model Definition Manual](#)]:

- ◇ Design Thoroughness by Product Design Review (PDR)
- ◇ Risk elimination by PDR

Damit sind die Erhöhung der Robustheit des zu entwickelnden Produkts und die damit einhergehende Risikominimierung gemeint. Bisher wurden BPM und Risikomanagement als zwei verschiedene Disziplinen betrachtet [Lhannaoui, Kabbaj und Bakkoury 2013]. Die Autoren [Lhannaoui, Kabbaj und Bakkoury 2013] schlagen in ihrer Arbeit „Towards an approach to improve business process models using risk management techniques“ ein Verfahren vor, was BPM und Risikomanagement verbindet. Da dieser Themenbereich bereits wissenschaftlich untersucht wird, besitzt er Relevanz und wird mit einbezogen.

4. Team Cohesion (TEAM)

Team Cohesion lässt sich weitestgehend mit Qualität des Teams übersetzen [Frohnhoff 2009]. Dieser Faktor soll unter anderem die Zusammenarbeit und den Zusammenhalt im Team beschreiben. Ebenfalls drückt dieser Faktor die Erfahrung eines Teams aus, also wie lange dieses Team schon zusammenarbeitet. Dieser Faktor ist auch im Bereich der Modellierung wichtig, da hierbei insbesondere die Zusammenarbeit zwischen den Modellierern und den am Prozess beteiligten Personen und Analysten wichtig ist [Freund und Rücker 2012].

5. Process Maturity (PMAT)

Mit diesem Faktor ist der Reifegrad des Softwareentwicklungsprozesses beschrieben. COCOMO verwendet hierzu in Anlehnung an CMMI (Capability Maturity Model Integrated) fünf Reifegradstufen. Dabei stellt die geringste Stufe keine definierten Abläufe im Bezug auf das Projekt dar. Die höchste Stufe beinhaltet dagegen neben vorhandener Ablaufbeschreibung und –definition auch einen kontinuierlichen Verbesserungsprozess. Die Stufen bauen hierbei aufeinander auf [Herzwurm und Pietsch 2009].

Der adaptierte Faktor beschreibt den Reifegrad der BPM-Einführung innerhalb einer Organisation. Es existieren angepasste Reifegradmodelle, wie das Business Process Maturity Model (BPMM) [[BPMM Specification](#)], welche diesen Faktor beschreiben könnten.

b) Kostentreiber

Die Kostentreiber sind jeweils in vier Bereiche eingeteilt (vgl. Kapitel 4.2).

Bereich Produkt

1. Required Software Reliability (RELY)

Required Software Reliability beschreibt die Verlässlichkeit der Software. Diese soll auf „very high“ gesetzt werden, falls Menschenleben davon abhängen. Während bei der Beschreibung von [B. Boehm und Harrowitz 2000] bei Ausfall oder Unzuverlässigkeit der Software den

Verlust von Menschenleben oder Lebensgefahr bedeutet, passt diese Beschreibung im Allgemeinen nicht für BPM. In seltenen Fällen (beispielsweise in Krankenhäusern) kann durch unzuverlässig modellierte Prozesse und damit deren Ausfall (beispielsweise das Ausbleiben einer Benachrichtigung von Herzstillstand auf einer Intensivstation) die Bewertung des Faktors mit „very high“ sinnvoll sein. Da allerdings auch bei Verwendung von GPM „kleinere Unannehmlichkeiten“ bis hin zu hohen finanziellen Ausfällen entstehen können, sollte dieser Faktor miteinbezogen werden und auf die Verlässlichkeit im Sinne von fehlerfreien Modellen adaptiert werden.

2. *Data Base Size (DATA)*

Die Bewertung dieses Faktors erfolgt durch Errechnen des Quotienten aus D/P mit D = DataBaseSize in Bytes durch P = ProgramSize in SLOC. Da GPM innerhalb eines BPMS laufen und dort auch über ein DBMS verwaltet werden, existiert die Aussage über eine Datenbankgröße erst nach der Implementierungsphase. Zum Zeitpunkt der Modellierung kann dieser Faktor nicht berechnet werden. Er ist aber für die Adaption dennoch interessant, um über die Ermittlung der Prozessgröße über geeignete Verfahren, wie das von [bppm13] vorgestellt, eine Aussage über den Aufwand zur Wartung und Pflege im Nachgang zur Modellierung zu erhalten.

3. *Product Complexity (CPLX)*

Dieser Faktor beschreibt die Komplexität des Produkts über die Bewertung von Kategorien wie beispielsweise „Control Operations“, „Computational Operations“ oder „Device-Independent Operations“ [B. Boehm und Harrowitz 2000]. Diese Zuteilung scheint für GPM nicht passend oder adaptierbar. Da die Komplexität des GPM über diverse Metriken beschrieben werden kann [Gruhn und Laue 2006], wird dieser Faktor zur Bewertung verwendet und adaptiert.

4. *Developed for Reusability (RUSE)*

RUSE Beschreibt den Prozentsatz der zu erstellenden Softwarekomponenten, die wiederverwendbar sein sollen. Da BPM ein Kreislauf ist und kontinuierlicher Verbesserung unterliegt, sind dessen Modelle immer vollständig wiederzuverwenden. In Kapitel 5.3.3 wird erläutert, ob dieser Faktor mit dem konstanten Wert von „very high“ bewertet werden soll, oder entfällt.

5. *Documentation Match to Lifecycle Needs (DOCU)*

Hier wird der Grad der benötigten Dokumentation des Produkts beschrieben. In Anlehnung an das camundaBPM Lebenszyklus-Modell aus Kapitel 2 wird die Prozessdokumentation getrennt vom Prozessmodell beschrieben [Freund und Rücker 2012]. Gerade bei komplexen Prozessen, die in BPMN 2.0 modelliert sind, kann das Prozessmodell unleserlich werden, wodurch ein einfaches Verständnis des Prozesses bei Betrachtung des Modells erschwert wird (BLOG und Freund12). Daher ist neben einem Prozessmodell auch immer eine Prozessdokumentation / Prozesslandschaft zu erstellen. Wie bei RUSE sollte dieser Faktor mit hohem Einfluss bewertet oder entfernt werden (Vgl. Kapitel 5.3.3).

Bereich Personal

6. *Analyst Capability (ACAP)*

Dieser Faktor beschreibt die Leistungsfähigkeit des Analysten / Chefdesigners. Dieser soll allerdings nicht seinen Erfahrungswert widerspiegeln – hierzu folgt der Faktor Application Experience in Punkt 9. Leistungsfähigkeit ist ein wichtiger Faktor in Projekten und daher als Faktor für die Schätzung relevant.

7. *Programmer Capability (PCAP)*

Analog zu (6) wird hier die Leistungsfähigkeit der Programmierer beschrieben und ist daher ebenfalls ein relevanter Kostentreiber.

8. *Personnel Continuity (PCON)*

Dieser Faktor beschreibt die Kontinuität der Mitarbeiter im Projekt. Da ein BPM-Projekt auf verschiedenen fachlichen und technischen Ebenen ansetzt, hat eine Kontinuität des zusammengestellten Projektteams positiven Einfluss auf das Projekt. Dieser Faktor wird selektiert.

9. *Application Experience (APEX)*

Dieser Faktor soll die Erfahrung des Teams mit der Fachlichkeit der zu entwickelnden Software beschreiben. Der fachliche Kontext und die damit verbundene Erfahrung ist bei BPM-Projekten ein sehr wichtiger Aspekt und sollte mit in die Schätzung einfließen.

10. *Platform Experience (PLEX)*

Mit diesem Faktor wird die Erfahrung des Teams über die zu nutzende Plattform und deren Middleware abgebildet. Im Bezug auf BPM spiegelt dies konkret die Erfahrung mit dem BPMS als Plattform und DBMN als Middleware wieder [Allweyer 2014].

11. *Language and Toolset Experience (LTEX)*

Analog zu PLEX drückt dieser Faktor die Erfahrung des Teams über die verwendete Programmiersprache und deren Werkzeuge aus. Dabei sind die Aspekte der Erfahrung sehr umfassend beschrieben (vgl. Abbildung 4.3). Dieser Faktor ist direkt auf BPM übertragbar und zu selektieren.

Bereich Plattform

12. *Time Constraint (TIME)*

Hiermit ist nicht das Zeitbudget des Projekts gemeint, sondern die Auslastung der Rechenleistung auf dem Zielsystem, also wie viel Prozessorzeit für dieses Projekt zur Verfügung steht. Da ein BPMS neben Kapazitäten für die Ausführung von Prozessen, auch über Ressourcen für die Modellierung verfügen muss, ist dieser Faktor relevant.

13. *Storage Constraint (STOR)*

Analog zu TIME beschreibt dieser Faktor den Auslastungsgrad des Storage-Systems der Zielumgebung und ist damit ebenfalls zu berücksichtigen.

14. *Platform Volatility (PVOL)*

Dieser Faktor beschreibt die Häufigkeit des Plattformwechsels. Damit ist der Austausch von Hard- und Software – auch DBMS oder Middleware – gemeint. Wie die beiden anderen Plattformfaktoren ist dieser ebenfalls zu berücksichtigen.

Bereich Projekt

15. Use of Software Tools (TOOL)

Tool-Einsatz beschreibt die Unterstützung der Programmierer durch die verwendeten Werkzeuge und wie gut diese in den Entwicklungsprozess eingebunden sind. Da BPMS auch über Modellierungstools verfügen [Allweyer 2014] ist dieser Faktor relevant. Zudem sind die Möglichkeiten der Werkzeuge – insbesondere bei vorhandener Unterstützung durch Prozessvorlagen – für den Modellerstellungsprozess sehr einflussreich.

16. Multisite Development (SITE)

Beschreibt verteiltes Arbeiten über mehrere Standorte hinweg und welche Möglichkeiten der Kommunikation dabei zur Verfügung stehen. Zu Zeiten der Entwicklung von COCOMO waren die Vernetzung des Internets und die damit verbundenen Möglichkeiten zur Kommunikation noch nicht so ausgeprägt und weiterentwickelt wie heute. Daher gibt es heute im Rahmen von Heimarbeitsplätzen und Outsourcing viele Entwickler, die nicht am Standort arbeiten. Dieser Faktor besitzt daher hohe Relevanz.

17. Required Development Schedule (SCED)

Bewertet den Zeitplan dahingehend, ob ausreichend Zeit für die Projektentwicklung zur Verfügung steht. Dies ist für BPM-Projekte ein relevanter Kostentreiber.

5.3.4 Adaption der Faktoren

a) Skalenfaktoren

1. Precedentedness (PREC)

Neben dem Grad der Nicht-Neuartigkeit fließen hier bei der Adaption noch weitere Aspekte ein. So kann auch die Neuartigkeit eines Projektes durch das Vorhandensein oder das Fehlen von Beispielvorgängen beschrieben werden. [Freund und Rücker 2012] nennen dies Vorkenntnisse, die über Baukästen innerhalb des Modellierungswerkzeuges oder über eine Art „Kochbuch“ [Freund und Rücker 2012] zur Verfügung stehen.

Weiter ist auch der Grad der Neuartigkeit des BPM-Projekts an sich relevant (vgl. Kapitel 3). Dabei ist der Aufwand für die häufigsten BPM-Projekte, wie Veränderungen oder Optimierung von bestehenden Prozessen oder die Einführung von neuen Prozessen unterschiedlich und sollte hier in die Bewertung mit einfließen.

Die von Boehm beschriebene Einteilung in die Gewichtungsklassen von VL bis XH wird daher um die Einschätzung der Neuartigkeit des zu modellierenden Prozesses und die Beschreibung der Projektart erweitert. Dabei sind Einführungsprojekte mit VL, also mit sehr niedriger Vertrautheit, zu bewerten.

2. *Development Flexibility (FLEX)*

wird in COCOMO unterteilt in:

- ◇ Softwarekonformität mit vorab definierten Anforderungen
- ◇ Softwarekonformität zu externen Schnittstellendefinitionen

Diese Faktoren lassen sich sehr gut auf die Modellierung übertragen. Die (ausführbare) Modellierung kann Voraussetzungen aufweisen, um beispielsweise mit bereits modellierten Prozessen zu kommunizieren. Ebenfalls kann diese mit externen Schnittstellen kommunizieren, wodurch entsprechende Spezifikationen in der Umsetzung einzuhalten sind.

Neben den Best Practices der OMG³, die eventuell eingehalten werden müssen, gibt es auch GoM und 7PMG als Richtlinien [Komus; Becker, Rosemann und Uthmann 2000; Mendling, Reijers und Aalst 2010], die hier eventuell als verpflichtend vorgeschrieben sein können. Ebenfalls beschreiben ([Komus; Becker, Rosemann und Uthmann 2000] in ihrer Arbeit entsprechende Verhaltensweisen. Dies bestimmt unter anderem die Flexibilität des Modellierers und damit auch des Modellierungsprozesses. Restriktivere Vorschriften sind daher mit VL zu bewerten.

3. *Architecture / Risk Resolution (RESL)*

Diese Faktoren lassen sich direkt übertragen. Die hauptsächliche Adaption des Faktors liegt im Bereich des Reviews der Spezifikation. Hier kann abgebildet werden, wie gut die vorhandene Prozessdokumentation ist. Abhängig von der Qualität des Dokuments wird auch das Prozessmodell erstellt. Eventuelle Unsicherheiten und erhöhter Aufwand durch veraltete und inkorrekte Dokumentationen fließen hierbei mit in die Schätzung ein.

Ferner können sich Risiken auch auf Schwierigkeiten in der Umsetzung beziehen: Ausfall eines Modellierers, Verlust des Zugangs zu Dokumenten, Datenverlust, etc.

4. *Team Cohesion (TEAM)*

Neben den Beschreibungen des Team Zusammenhalts aus [B. Boehm und Harrowitz 2000] gibt es insbesondere auf das Thema BPM bezogen weitere begünstigende bzw. einschränkende Faktoren. So kann die Einbindung der Personen in und am Prozess in die Phase der Modellierung Einfluss nehmen.

Das BPMM enthält keine Aussage darüber ob, und in welcher Stufe ein Chief Process Officer positiven oder negativen Einfluss auf den BPM Prozess innerhalb der Organisation hat [Allweyer 2007]. Ein Hauptverantwortlicher CPO [Gadatsch 2013; Posluschny 2012] könnte hier als Promoter für den Prozess wirken und den Zusammenhalt und die Verfolgung der gemeinsamen Ziele verstärken.

Die Einteilung lautet hier wie folgt:

³http://www.bpmnquickguide.com/quickguide/index.html?modeling_best_practices.htm

Die Unterstützung des Teams und des Projektes durch einen CPO ist ein stark begünstigender Faktor und kann mit H und XH bewertet werden. Viele Process Owner, unterschiedliche Zielsetzungen und generell fehlende Unterstützung (analog zu der ursprünglichen Beschreibung) ergeben VL.

5. Process Maturity (PMAT)

Analog zum verwendeten Reifegradmodell CMMI gibt es für BPM das Business Process Maturity Model (BPMM)⁴, welches von der Object Management Group (OMG) 2007 als Standard veröffentlicht wurde.

Dabei sind die Stufen an CMMI angelehnt und daher direkt für diesen Skalenfaktor adaptierbar⁵:

- ◇ *Initial*: Auf dieser untersten Stufe befindet sich jedes Unternehmen, in dem überhaupt kein Prozessmanagement stattfindet, wo die Prozesse also weitgehend ungeplant ablaufen.
- ◇ *Managed*: Prozesse sind definiert und wiederholbar, Abteilungen managen ihre (Teil-)prozesse isoliert.
- ◇ *Standardized*: Es sind unternehmensweite, standardisierte „end-to-end“-Prozesse definiert.
- ◇ *Predictable*: Es findet eine quantitative Planung und Überwachung der Prozesse statt, um vorhersagbare Ergebnisse zu erzielen.
- ◇ *Innovating*: Die Prozesse werden kontinuierlich weiterentwickelt und verbessert.

Zudem ist der BPMN-Reifegrad seit 2009 stetig gestiegen [Freund und Rücker 2012]. Ein begünstigender Faktor für die zunehmende Verbreitung des Standards BPMN 2.0 war die zunehmende Bereitschaft vieler namhafter Hersteller, die Unterstützung des Standards in ihren Produkten einzuführen. Dies könnte auch die BPM Prozessmaturität in den Unternehmen begünstigend beeinflussen.

b) Kostentreiber

Hier finden sich die meisten Anpassungen an das Schätzmodell neben den veränderten Inputfaktoren. Dies ist hauptsächlich dadurch bedingt, dass das Produkt ein anderes ist. Man erhält zwar Software, diese ist allerdings mit der klassisch ausführbaren Software nicht direkt vergleichbar. Die Projektbedingungen eines zu modellierten Prozesses sind anders zu definieren als die eines Softwareentwicklungsprojekts.

Kategorie Produkt

1. Required Software Reliability (RELY)

⁴<http://www.omg.org/spec/BPMM/>

⁵<http://www.kurze-prozesse.de/2007/11/08/das-business-process-maturity-model-bpmm-der-omg/>

Required Software Reliability beschreibt die Verlässlichkeit der Software. Diese soll auf „very high“ gesetzt werden, falls Menschenleben davon abhängen. Übertragen auf die Modellerstellung, wäre es in diesem Fall wichtig, dass der Prozess auch absolut korrekt modelliert wird. Dabei wird nicht nur die geforderte semantische Korrektheit aus der Prozessdokumentation, sondern auch die korrekte syntaktische Modellierung in der jeweiligen Modellierungssprache beschrieben.

Der Faktor wird daher in **Process Model Correctness** (CORL) (deutsch: Korrektheit des Prozessmodells) umbenannt.

Die Einordnung von Boehm für die Kategorien von VL bis XH kann übernommen werden.

2. Data Base Size (DATA)

Der Faktor findet für die Aufwandschätzung zur Modellerstellung keine Anwendung und entfällt. Allerdings kann für die Implementierungsphase mit diesem Faktor eine Aussage über die zu erstellenden Testfälle oder Wartung des BPMS getroffen werden.

In ihrer Arbeit schlagen die Autoren [Baklizky u. a. 2013] et al. eine Vorgehensweise zur Zählung der BPM Projektpunkte (BPPM) vor. Damit lässt sich die Projektgröße ermitteln und unter Einbeziehung der Datenbankgröße des BPMS der Faktor berechnen.

3. Product Complexity (CPLX)

(Gruhno6) schlagen in ihrem Paper „Komplexitätsmetriken für Geschäftsprozessmodelle“ verschiedene Metriken zur Bestimmung der Komplexität vor. Tabelle 5.8 stellt diese Metriken vor:

Die Metriken aus Tabelle 16 sind zum Teil erst zu ermitteln, wenn das Modell schon erstellt ist. Da bei der Schätzung der Eingangsgrößen nicht direkt die Anzahl der Elemente wie Gateways oder Kontrollflüsse bekannt ist, können die in Gruhno6 vorgestellten Metriken bei der Schätzung nicht verwendet werden. Lediglich die Verschachtelungstiefe des Prozessmodells lässt sich über die Anzahl der Pools und Lanes ermitteln.

Daher wird dieser Faktor auf einer Expertenschätzung basieren. An dieser Stelle können jedoch Empfehlungen für die Kategorisierung in die Komplexitätsklassen vorgenommen werden, wobei die Metrik „Zahl der Aktivitäten“ um die Betrachtung der fehlenden Aspekte erweitert wird:

Längste Abfolge von Aktivitäten und Parallelität

Über die Relation zwischen der längsten Abfolge der Aktivitäten und der Anzahl der Aktivitäten kann unter Zuhilfenahme der Pools und Lanes auf den Grad der Parallelität des Prozesses geschlossen werden. Eine höhere Parallelität bedeutet meist eine komplexere Abfolge der Aktivitäten, was auch einen Einfluss auf die Modellierung haben wird (Vermeidung von Deadlocks etc.).

Einhaltung von GoM und 7PMG

Software-Metrik	GPM-Metrik	Anwendung, Bewertung
Lines of Code	Zahl der Aktivitäten	Sehr einfache Metrik, Kontrollfluss nicht berücksichtigt, ebenso wie Parallelität und Prozesslänge
Zyklomatische Zahl	Kontrollfluss-Komplexität nach Cardoso	Berücksichtigt Komplexität an Entscheidungsknoten (Gateways)
Verschachtelungstiefe	Verschachtelungstiefe	Aufschlüsselung der Prozesse in Unterprozesse.
Knotenzahl	Zahl der Handles / Zahl der Transaktionen in BPMN 2.0	Misst ?Wohlstrukturierbarkeit? (Ein und Aussprünge in bzw. aus Kontrollstrukturen ? in BPMN 2.0 Transaktionen)
Kognitives Gewicht	Verallgemeinertes Kognitives Gewicht	Misst den von einem Leser eines Modells zu erfassenden ?Informationsgehalt?
Fan-in / Fan-out	Fan-in / Fan-out	Zu große Fan-In / Fan-out Zahlen bedeuten eine schlechte Unterteilung des Modells in Teilmodelle

Tabelle 5.8: Übersicht Komplexitätsmetriken. Adaptiert aus [Gruhn und Laue 2006]

Neben der semantischen Korrektheit können auch die Einhaltung der GoM und 7PMG das Prozessmodell komplexer gestalten, da dadurch mehr Elemente modelliert werden müssen, um die Übersichtlichkeit zu verbessern.

Ausführungsdauer des Prozesses

Eine lange Ausführungsdauer des Prozesses bedeutet möglichst resiliente Modellierung des Prozesses und damit eine gesteigerte Komplexität. Eine lange Ausführungsdauer bedeutet in der Regel, dass die Daten und Zustände des Prozesses zwischengespeichert und so lange vorgehalten werden müssen, bis der Prozess terminieren kann. Ebenfalls können viele per Definition lange laufende Prozesse viele Ressourcen belegen. Dies ist für die Modellierung nicht relevant, sondern spielt für die Implementierung eine Rolle. Daher wird der Faktor nicht in die Schätzung der Modellierung einfließen.

4. Developed for Reusability (RUSE)

Da das Prozessmodell kontinuierlicher Verbesserung unterliegt (vgl. BPM Lebenszyklus Kapitel 3), kann hier der Aufwand für die Berücksichtigung der Wiederverwendbarkeit nicht

als gering eingeschätzt werden. Daher entfallen die Auswahlmöglichkeiten für VL und L, sodass nur Werte ≥ 1 zur Wahl stehen. Dadurch werden zum Teil auch Aufwände für Wartung und Maintenance abgedeckt [*COCOMO II Model Definition Manual*].

Der Faktor wird daher in **Levels of Model Reuse** (deutsch: Grad der Wiederverwendung des Modells) umbenannt.

Die Einteilung kann adaptiert wie folgt geschehen:

N beschreibt den Normalfall in dem das Prozessmodell nur der eigenen Verbesserung unterliegt. H bis XH beschreiben die Wiederverwendung und auch, inwiefern andere Prozesse von diesem abhängen oder darauf basieren.

5. *Documentation Match to Lifecycle Needs (DOCU)*

Analog zu 4. ist der Aufwand für die Dokumentation nicht als mindernder Faktor zu betrachten, da neben der Erstellung des Prozessmodells auch stets die Prozessdokumentation aktualisiert werden muss, sofern diese sich nicht aus dem Prozessmodell ablesen lässt [Freund und Rücker 2012]. Daher entfallen auch hier die Werte $< „1“$ und VL und L stehen nicht mehr zur Auswahl.

Der Faktor wird daher in **Documentation Requirements** (deutsch: Anforderungen an Dokumentation) umbenannt.

Die Einteilung kann wie folgt geschehen:

- ◇ Normal bedeutet keine zusätzliche Erstellung oder Aktualisierung der Prozessdokumentation, da diese direkt aus dem Prozessmodell abgelesen werden kann.
- ◇ XH würde die komplette Neuerstellung der Dokumentation in Abhängigkeit des erstellten Prozessmodells bedeuten.

Kategorie Personal

6. *Analyst Capability (ACAP)*

Analog zu den Punkten (7) und (8) wird hier die Leistungsfähigkeit der Person beschrieben und kann direkt übernommen werden.

7. *Programmer Capability (PCAP)*

Der Faktor kann mit einer Umbenennung in **Modeller Capability** adaptiert werden, da beschrieben werden soll, wie leistungsfähig der Modellierer ist.

8. *Personnel Continuity (PCON)*

Dieser Faktor ist in SE-Projekten und BPM-Projekten gleichermaßen relevant und wird daher übernommen.

9. *Application Experience (APEX)*

Wie beschrieben ist hier der Grad der Kenntnisse über die zu erstellende Applikation abgebildet. Da keine Applikation, sondern ein Prozessmodell erstellt wird, sollten hier

sowohl die fachlichen Kenntnisse über BPM in als auch die über den zu modellierenden Prozess einfließen. Eine Überschneidung mit der adaptierten Auffassung von PREC findet nicht statt, da hier der Kenntnisstand über den Prozess selbst relevant ist und nicht die Art des Modellierungsprojektes oder dessen Grad an Nicht-Neuartigkeit. Da der Chefdesigner, beziehungsweise in diesem Kontext der Modellierer, den Prozess genau kennen sollte, ist hier auch sein Kenntnisgrad enthalten und bietet dadurch nochmal eine Abgrenzung zu ACAP.

Der Faktor wird daher in **BPM Experience** (BPEX) (deutsch: BPM Erfahrung) umbenannt.

Die Einteilung kann aus [B. Boehm und Harrowitz 2000] ohne Adaption übernommen werden.

10. Platform Experience (PLEX)

Dieser Faktor kann direkt adaptiert werden, da hier eine Verwendung von BPMS und entsprechenden Modellierungswerkzeugen obligatorisch ist, um das Prozessmodell korrekt zu erstellen [Freund und Rücker 2012].

11. Language and Toolset Experience (LTEX)

Dieser Faktor kann direkt adaptiert werden und die Einteilung nach [B. Boehm und Harrowitz 2000] mit der Erweiterung um folgende Aspekte erfolgen:

- ◇ Wie viel Erfahrung hat das Team mit der Modellierungssprache, den dafür verwendeten Werkzeugen und des verwendeten BPMS?
- ◇ Wie viel Erfahrung hat das Team mit 7PMG und GoM?

Kategorie Plattform

12. Time Constraint (TIME)

Da auch ein BPMS zur Ausführung eines Prozessmodells Ressourcen benötigt, findet dieser Faktor direkte Anwendung und kann ohne weitere Adaption übernommen werden.

13. Storage Constraint (STOR)

Analog zu oben genanntem Punkt 12.

14. Platform Volatility (PVOL)

Analog zu den Punkten 12. und 13. kann hier ein Wechsel des BPMS, oder die Notwendigkeit zur Bereitstellung von mehr Ressourcen und damit einhergehendem Hardwarewechsel den Aufwand beeinflussen. Die Adaption erfolgt daher ebenfalls ohne Einschränkungen.

Kategorie Projekt

15. Use of Software Tools (TOOL)

Da BPMS neben der Process Engine auch über Werkzeuge zur Modellierung verfügen [Allw-eyer 2014], haben diese einen Einfluss auf den Aufwand. Einige Werkzeuge zur Modellierung

besitzen Datenbanken über vorgefertigte Prozessmodelle⁶, die der Modellierer dann verwenden und damit den Aufwand reduzieren kann. Aufgrund der Integration von Process Engine und Modellierwerkzeug in einem Tool kann der Faktor auch stark begünstigend für die Aufwandsschätzung wirken. Daher wird XH hierfür zusätzlich zur Auswahl gestellt und mit dem Faktor 0,75 belegt.

Die Einteilung von Boehmoo wird daher um folgendes ergänzt:

Der Faktor XE wird gewählt, wenn das Werkzeug „alles aus einer Hand“ bietet und vollständig in den Entwicklungsprozess integriert ist. VL dagegen, wenn das erstellte Prozessmodell im Nachgang zur Modellierung noch manuell für die Implementierungsphase und eine Ausführung auf der Process Engine angepasst oder umgeschrieben werden müsste.

Der willkürlich gewählte Faktor von 0,75 für XH wäre in der Evaluation zu prüfen. Ebenfalls wäre zu prüfen, inwiefern zusätzliche Kosten für den Kauf von solchen hoch integrierten Werkzeugen eine Auswirkung auf die Schätzung haben können und dem begünstigten Faktor entgenspielen.

16. Multisite Development (SITE)

Durch die stärkere Vernetzung und Outsourcing besitzt dieser Faktor nach wie vor Relevanz und muss daher nicht adaptiert werden. Es wird nur zunehmend unwahrscheinlicher, dass der Faktor XH hierfür gewählt wird und alle Arbeiten im Hause selbst erledigt werden. Diese Faktoren sind daher zwar nicht zu erwarten, bleiben allerdings aus Gründen der Vollständigkeit bestehen.

17. Required Development Schedule (SCED)

Dieser Faktor bedarf, aufgrund der Allgemeingültigkeit für Projekte [Wieczorrek und Mertens 2010] aller Art, keiner Adaption und wird direkt übernommen.

5.3.5 Zusätzliche Faktoren

Da für die Aufwandsschätzung noch weitere Faktoren erkannt wurden, wird das Early Design Model von COCOMO um jeweils einen Skalenfaktor und Kostentreiber erweitert.

Levels of Process Documentation

Es besteht die Möglichkeit, dass ein Prozess noch nicht dokumentiert worden ist und daher die Aufwandsschätzung dafür erstellt wird. Dadurch ist dieser Skalenfaktor in Abhängigkeit von der zu erwartenden Komplexität und der Iterationen zur Schätzung der Einflussgrößen für die gesamte Schätzung relevant. Falls der Prozess korrekt und aktuell dokumentiert wurde, würde dies dem Grad „high“ entsprechen. Einem undokumentierten Prozess hingegen würde „very low“ entsprechen, da hier die Vorkenntnisse kaum vorhanden sind. Dieser Faktor würde dann mit dem zu erwartenden Grad der Komplexität verbunden werden. Hier sollte dann auch die Art der Dokumentationsanalyse wiedergespiegelt werden. Von

⁶https://en.wikipedia.org/wiki/Comparison_of_Business_Process_Modeling_Notation_tools

der Verbesserung bis hin zur Neuerstellung der Prozessdokumentation kann graduell in den fünf Faktoren gewählt werden. Multiplikation deckt dann sowohl den Aufwand für den Workshop zur Ermittlung des Prozesses, sowie auch den Fehler in der Schätzung der Eingangsgröße ab. Daher sollten die Faktoren hier ein relativ hohes Gewicht erhalten.

SFi	COCOMO -Kürzel	Skalenfaktor	Gewichtung					
		Beschreibung	VL	L	N	H	VH	XH
SF6	PDOC	Levels of Process Documentation / Stand der Prozessdokumentation	8,6	6,88	5,16	3,44	1,72	0
		Ist eine Prozessdokumentation vorhanden?						
		Muss diese aktualisiert werden?						
		Können daraus die Eingangsgrößen abgeleitet werden?						

Tabelle 5.9: Skalentreiber „Levels of Process Documentation“ (Adaption aus [Frohnhoff 2009])

Der höchste Wert wurde unter der Annahme berechnet, dass der Aufwand für die Erstellung einer Prozessdokumentation ungefähr der Hälfte des Modellierungsaufwandes entspricht [*Aufwandsschätzung für die Prozessmodellierung*]. Mit den weiteren Annahmen von einer Eingangsgröße von 10 KLOC und der Bewertung aller sonstiger Skalenfaktoren mit XH, ist nach $\log_{10}$ 8,6 als höchster Wert errechnet worden. Dieser wird linear bis zum Wert 0 in äquidistanten Schritten in die jeweiligen Kategorien eingeteilt.

Process Confidentiality

Als zusätzlicher Kostentreiber ist Process Confidentiality als Grad der Vertraulichkeit des zu modellierenden Prozesses in der Kategorie Produkt eingeführt. Dieser soll die Tatsache beschreiben, dass nicht alle an der Modellierung beteiligten Personen genaue Kenntnis über den Prozess haben dürfen. So wird innerhalb des Projektes eventuell eine „anonymisierte“ Version des Prozessmodells benötigt, welche zwar die Aktivitäten und Flows zeigt, nicht aber deren genaue Bezeichnung und Funktion innerhalb des Prozesses. Freund¹² nennen auch verschiedene Sichten auf Prozesse und Schnittstellen. Eine Organisation möchte unter Umständen bei Kooperation mit Dritten, für die Schnittstellen modelliert werden müssen, nicht ihre Betriebsgeheimnisse offen legen.

Da dadurch zusätzliche Modelle erstellt werden müssen, bedeutet dies auch zusätzlichen Aufwand und sollte daher bei der Schätzung mit berücksichtigt werden. Es gibt allerdings Vorschläge zur automatisierten Erstellung von anonymisierten Prozessmodellen [Skouradaki u. a. 2014]. Daher wären die Faktoren hier mit relativ geringen Werten zu beziffern.

Normal (N) beschreibt hier den Zustand von nicht bestehenden Einschränkungen. Während VL die völlige Offenlegung der Prozesse darstellt und damit die Zusammenarbeit der am Projekt beteiligten Personen begünstigt, stellt VH die vollständige Anonymisierung der Modelle während der Zusammenarbeit dar. Auf Basis von N wurde in jede Richtung mit 0,1 linear skaliert.

EFi	COCOMO -Kürzel	Kostentreiber	Gewichtung					
		Beschreibung	VL	L	N	H	VH	XH
EF6	CONF	Process Confidentiality / Vertraulichkeit des Prozesses	0,8	0,9	1	1,1	1,2	-
		Sind die zu modellierenden Prozesse Vertraulich? Müssen diese für die Zusammenarbeit mit Dritten anonymisiert werden? Müssen mehrere Sichten auf ein Prozessmodell erstellt werden? Kann die Anonymisierung des Prozessmodells (voll-)automatisch erfolgen?						

Tabelle 5.10: Kostentreiber "Process Confidentiality" (Adaption aus [Frohnhoff 2009])

5.3.6 Gesamtüberblick Adaption

Nachdem nun die Quantifizierung der Eingangsgröße und alle Einflussgrößen selektiert und adaptiert wurden, geben die nachfolgenden Tabellen einen Überblick über BPM COCOMO.

Die Adaptionstabelle (vgl. Abbildung 5.3) enthält einen gesamten Überblick über die Eingangsgrößen, Skalenfaktoren und Kostentreiber. Dabei sind neue Faktoren **fett**, adaptierte Faktoren *kursiv* markiert. Aus Gründen der Übersichtlichkeit wird auf die Darstellung der deutschen Übersetzung verzichtet.

Abbildung 5.4 zeigt die nun adaptierten Skalenfaktoren für BPM COCOMO. Die restlichen Werte entsprechen den unkalibrierten Werten aus Abbildung 4.2 (vgl. Kapitel 4.1.3):

Abbildung 5.5 zeigt die nun adaptierten Kostentreiber für BPM COCOMO. Die restlichen Werte entsprechen den unkalibrierten Werten aus Abbildung 4.3 (vgl. Kapitel 4.1.3):

5.3.7 Gesamtüberblick Berechnung

Erinnerung (vgl. Kapitel 4.1.3): Der Aufwand in Personenmonaten errechnet sich aus folgender Formel:

$$PE = A \cdot Size^E \cdot \prod_{i=1}^{17} EM_i$$

Auch hier wird der von [B. Boehm und Harrowitz 2000] vorgeschlagene Erfahrungswert A mit 2,94 übernommen.

Einfluss der Faktoren SF und EF

Die möglichen Werte für SF und EF haben sich durch die Einführung und Streichung von Faktoren verändert. Für die übernommenen und nur in der Beschreibung adaptierten Faktoren wurden die unkalibrierten Werte aus [B. Boehm und Harrowitz 2000] verwendet.

	COCOMO II	BPM COCOMO	Kürzel
Eingangsgröße	(K)LOC / Unadjusted Function Points	<i>Business Model Points (BMP) to LOC Conversion</i>	
Skalenfaktoren			
	Precedentedness	Precedentedness	PREC
	Development Flexibility	Development Flexibility	FLEX
	Architecture / Risk Resolution	Architecture / Risk Resolution	RESL
	Team Cohesion	Team Cohesion	TEAM
	Process Maturity (CMM)	<i>BPM Process Maturity (BPPM)</i>	PMAT
		Levels of Process Documentation	PDOC
Kostentreiber			
Produkt	Required Software Reliability	<i>Process Correctness-Level</i>	CORL
	Data Base Size	-	-
	Product Complexity	<i>Process Model Complexity</i>	CPLX
	Developed for Reusability	<i>Levels of Model Reuse</i>	RUSE
	Documentation Match to Lifecycle Needs	<i>Documentation Requirements</i>	DOCU
		Process Confidentiality	CONF
Personal	Analyst Capability	Analyst Capability	ACAP
	Programmer Capability	<i>Modeller Capability</i>	PCAP
	Personnel Continuity	Personnel Continuity	PCON
	Application Experience	<i>BPM Experience</i>	<i>BPEX</i>
	Platform Experience	Platform Experience	PLEX
	Language and Toolset Experience	Language and Toolset Experience	LTEX
Plattform	Time Constraint	Time Constraint	TIME
	Storage Constraint	Storage Constraint	STOR
	Platform Volatility	Platform Volatility	PVOL
Projekt	Use of Software Tools	Use of Software Tools	TOOL
	Multisite Development	Multisite Development	SITE
	Required Development Schedule	Required Development Schedule	SCED

Abbildung 5.3: Adaptionstabelle BPM COCOMO mit Eingangsgrößen, Skalenfaktoren und Kostentreibern

SF _i	COCOMO-Kürzel	Skalenfaktor	Gewichtung					
		Beschreibung	VL	L	N	H	VH	XH
SF ₁	PREC	Precedentedness / Vertrautheit	6,2	4,96	3,72	2,48	1,24	0
		Ähnlichkeit der Fachlichkeit zu bereits existierenden Systemen? Faktoren, die zusätzlich zu beachten sind: Neuartigkeit des zu modellierenden Prozesses, Beschreibung der Projektart						
SF ₂	FLEX	Development Flexibility / Flexibilität der Entwicklung	5,07	4,05	3,04	2,03	1,01	0
		Wie flexibel kann entwickelt werden bezüglich technischer Anforderungen? Faktoren, die zusätzlich zu beachten sind: Vorgeschriebene Einhaltung von Modellierungsgrundsätzen oder Best Practices						
SF ₃	RESL	Architecture/Risk Resolution / Architektur und Risikoausschluss	7,07	5,65	4,24	2,83	1,41	0
		Wie gründlich war das Review der Grobspezifikation, und wie gut sind Punkte bekannt, die mit einem Risiko belastet sind? Faktoren, die zusätzlich zu beachten sind: Qualität der Prozessdokumentation						
SF ₄	TEAM	Team Cohesion / Teaminteraktion	5,48	4,38	3,29	2,19	1,1	0
		Wie gut funktioniert das Team? Faktoren, die zusätzlich zu beachten sind: Vorhandensein eines CPO						
SF ₅	PMAT	BPM Process Maturity (BPMM) / BPM Prozessreife (BPMM)	7,8	6,24	4,68	3,12	1,56	0
		Reifegrad des Business Process Managements im Unternehmen						
SF ₆	PDOC	Levels of Process Documentation / Stand der Prozessdokumentation	8,6	6,88	5,16	3,44	1,72	0
		Faktoren, die zusätzlich zu beachten sind: Ist eine Prozessdokumentation vorhanden? Muss diese aktualisiert werden? Können daraus die Eingangsgrößen abgeleitet werden?						

Abbildung 5.4: Übersicht Skalenfaktoren BPM COCOMO (Adaption aus [Frohnhoff 2009])

	EF _i	COCOMO-Kürzel	Kostentreiber	Gewichtung					
			Beschreibung	VL	L	N	H	VH	XH
Produkt	EF ₁	CORL	Process Correctness-Level	0,82	0,92	1	1,1	1,26	-
			Erforderliche semantische und syntaktische Fehlerfreiheit						
	EF ₂	CPLX	Process Model Complexity	0,73	0,87	1	1,17	1,34	1,74
			Komplexität des Prozessmodells						
	EF ₃	RUSE	Levels of Model Reuse	-	-	1	1,07	1,15	1,24
			Wiederverwendung des Prozessmodells						
	EF ₄	DOCU	Documentation Requirements	-	-	1	1,1	1,23	-
			Anforderungen an Prozessdokumentation						
	EF ₅	CONF	Process Confidentiality	0,8	0,9	1	1,1	1,2	-
			Vertraulichkeit des Prozessmodells						
Personal	EF ₆	ACAP	Analyst Capability	1,42	1,19	1	0,85	0,71	-
			Leistungsfähigkeit Analyst						
	EF ₇	PCAP	Modeller Capability	1,34	1,15	1	0,88	0,76	-
			Leistungsfähigkeit Modellierer						
	EF ₈	PCON	Personnel Continuity	1,29	1,12	1	0,9	0,81	-
			Fortbestand des Teams						
	EF ₉	BPEX	BPM Experience	1,22	1,1	1	0,88	0,81	-
			BPM Erfahrung						
	EF ₁₀	PLEX	Platform Experience	1,19	1,09	1	0,91	0,85	-
			Erfahrung mit BPMS und Middleware						
Plattform	EF ₁₁	TLEX	Language and Toolset Experience	1,2	1,09	1	0,91	0,84	-
			Erfahrung mit Modellierungssprache und Werkzeugen						
	EF ₁₂	TIME	Time Constraint	-	-	1	1,11	1,29	1,63
			Verfügbarkeit von Rechenleistung – Auslastung des Systems						
	EF ₁₃	STOR	Storage Constraint	-	-	1	1,05	1,17	1,46
			Verfügbarkeit von Speicherplatz – Auslastung des Systems						
Projekt	EF ₁₄	PVOL	Platform Volatility	-	0,87	1	1,15	1,3	-
			Häufigkeit des Hard- und Softwareaustausches						
	EF ₁₅	TOOL	Use of Software Tools	1,17	1,09	1	0,9	0,78	0,75
			Verwendung und Möglichkeiten der Werkzeuge zur Modellierung						
	EF ₁₆	SITE	Multisite Development	1,22	1,09	1	0,93	0,86	0,8
			Standortübergreifende Zusammenarbeit und Kommunikation						
	EF ₁₇	SCED	Required Development Schedule	1,43	1,14	1	1	1	-
			Zeitbeschränkungen des Projekts						

Abbildung 5.5: Übersicht Kostentreiber BPM COCOMO (Adaption aus [Frohnhoff 2009])

Der Skalenfaktor berechnet sich aus:

$$E = B + 0,01 \cdot \sum_{i=1}^6 SFi$$

Dadurch ergibt sich für den Skalenfaktor E der Wertebereich von 0,91 (unverändert) bis 1,312.

Der Aufwands-Multiplikator aus den Kostentreibern kann durch die veränderten Werte und den eingefügten Faktor CONF zwischen 0,0618 und 108,35 liegen. Die veränderten Werte liegen damit im Wertebereich von 0,0569 als geringster und 115,58 als höchster Wert des originalen COCOMO II Verfahrens und scheinen plausibel.

Damit ist die Beschreibung von BPM COCOMO vollständig.

5.4 Web-basiertes Berechnungstool

In Anlehnung an das Web-basierte Berechnungstool der Universität of Southern California⁷ wurde für die vorgestellte Schätzmethode *BPM COCOMO* eine ähnliche Oberfläche erstellt.

Bei der Erstellung des Designs, waren eine vereinfachte Verwendung und die nicht-technische Ausrichtung der Auswahlmöglichkeiten Gründe für die Auswahl der simplifiziert gestalteten Benutzeroberfläche. Dadurch ist das Werkzeug nicht nur auf Modellierer ausgerichtet, sondern auch an die nicht IT-affinen Prozessbeteiligten, die eine Schätzung erstellen oder ein Gefühl für den Umgang mit BPM und dessen Aufwand erhalten möchten.

Die Berechnungen der Faktoren und Ergebnisse basieren auf JavaScript⁸, das Design und die Formularfelder wurden mit dem Bootstrap Framework⁹ erstellt.

Die Berechnung der Schätzung ist in vier Schritte unterteilt:

1. Eingabe der Einflussgrößen
2. Eingabe der Skalen- und Kostentreiber
3. Auswahl der Modellierungssprache
4. Berechnung des Aufwandes

⁷http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html

⁸<http://www.w3.org/standards/webdesign/script>

⁹<http://www.getbootstrap.com>

BPM CoCoMo Tool - Diplomarbeit

Dieses Tool wurde im Rahmen der Diplomarbeit: Aufwandsschätzung bei Geschäftsprozessmodellerstellung an der Universität Stuttgart erstellt.

Mit Hilfe dieser Anwendung soll eine Aufwandsschätzung für die Modellerstellungsphase erstellt werden können. Hierbei wurde das CoCoMo II Modell entsprechend verändert und für die Auswahl von für BPM relevanten Faktoren angepasst.

Zur Berechnung wird das "Early Design Model" verwendet. Die LOC werden über die Ermittlung von für BPM Adaptierten Function-Points namens Business Modeling Points (BMP) ermittelt.

Schritt 1: Bestimmen der Business Model Points und Einteilung in die jeweilige Komplexitätsklasse

Ermittlung BMP							
Eingabegrößen	Anzahl		Niedrig ●	Mittel ●	Hoch ●		Gesamt
Anzahl der Aktivitäten		X	3	4	6	=	
Anzahl der Ereignisse		X	4	5	7	=	
Anzahl der Abfragen		X	3	4	6	=	
Anzahl der Datenobjekte		X	7	10	15	=	
Anzahl der Interaktionen		X	5	7	10	=	
Anzahl summieren							

Abbildung 5.6: Screenshot BPM COCOMO Tool - eigene Anfertigung

Schritt 1 – Eingabe der Eingangsgrößen als Business Model Points

Hier werden die Eingangsgrößen, wie in Kapitel 5.3.1 beschrieben, in die Eingabemaske eingetragen. Hierbei können die ermittelten BMP direkt in die jeweilige Komplexitätsklasse eingetragen werden. Durch Klick auf „BMP berechnen“ erhält man die Summe der BMP.

Schritt 2 – Eingabe der Skalenfaktoren und Kostentreiber

In diesem Schritt werden die Skalenfaktoren und Kostentreiber entsprechend ihres Einflusses von „very low“ bis „extra high“ bestimmt. Dabei wurden der Übersichtlichkeit halber die Bezeichnungen in ein Bewertungssystem von 0 – 5 übersetzt.

Schritt 3 – Auswahl der Modellierungssprache

Um die ermittelten BMP in LOC umzuwandeln, wird hier mittels der errechneten Umrechnungsfaktoren für die jeweilige Modellierungssprache der Wert der BMP mit dem in Kapitel 5.3.2 ermittelten Durchschnittswert für die Modellierung eines BMP multipliziert.

Schritt 4 – Berechnung des Aufwandes

Mit einem Klick auf „Aufwand berechnen“ wird das Ergebnis basierend auf der zuvor getroffenen Auswahl und Eingabe in Personenmonaten berechnet.

5.5 Vorschlag zur Evaluation der Methode

Der Zugang zu Projektdaten eines in Deutschland ansässigen Softwarekonzerns ist leider nicht zustande gekommen. Aufgrund von fehlenden Realweltdaten konnte keine eigene Evaluation der entwickelten Methode durchgeführt werden.

In Anlehnung an die Ermittlung der Einflussgrößen und Faktoren in COCOMO II wird eine empirische Evaluation vorgeschlagen [B. Boehm und Harrowitz 2000]. Hierzu werden Projektdaten benötigt, bei denen die ermittelten und tatsächlichen Aufwände protokolliert wurden, um diese weiter zu verwenden. Ebenfalls könnten hierzu Fachstudien in Auftrag gegeben werden, oder Kooperationsprojekte an anderen Hochschulen zu relevanten Projektdaten befragt werden.

Die Auswahl der Projekte sollte einen Querschnitt aus den in Kapitel 3.2 erwähnten Projektarten sein:

- ◇ Einführungsprojekte von BPM mit zu erstellender Dokumentation
- ◇ Einführungsprojekte von BPM mit vorhandener Dokumentation
- ◇ Business Project Reengineering Projekt

Folgende Untersuchungen zur Evaluation werden vorgeschlagen:

1. Evaluation der vorgeschlagenen Methode im Allgemeinen
2. Evaluation der ausgewählten Skalenfaktoren
3. Evaluation der ausgewählten Kostentreiber
4. Evaluation der gewählten Gewichtungen für 2. und 3.

In Abhängigkeit von den Resultaten könnten möglicherweise noch weitere Einflussgrößen identifiziert werden, die dann zu diesem Modell hinzugefügt werden müssen.

1. Evaluation der vorgeschlagenen Methode im Allgemeinen

Khatibi et al evaluieren die Resultate von COCOMO II anhand der Berechnung von Performance Metriken [Khatibi und Jawawi 2011; Khatibi u. a. 2011]. Dabei wird der relative Fehler (Relative Error - RE) wie folgt berechnet:

$RE = (\text{geschätzter Aufwand} - \text{tatsächlicher Aufwand}) / \text{tatsächlicher Aufwand}$.

Damit wird die Abweichung der erstellten Schätzung beschrieben. Kleine Werte bedeuten damit eine geringere Abweichung und bestätigen die Validität des Modells.

2. Evaluation der ausgewählten Skalenfaktoren

Hier sollten alle gewählten Skalenfaktoren auf ihre Relevanz für das Schätzmodell und auf Plausibilität überprüft werden. In Abhängigkeit von den Ergebnissen aus (1), kann hier durch Selektion oder Deselektion von bestimmten Faktoren der relative Fehler minimiert werden.

3. Evaluation der ausgewählten Kostentreiber

Wie bei Skalenfaktoren ist hier die Selektion der Einflussgrößen zu überprüfen. Insbesondere wäre die Relevanz der Faktoren auf die Phase der Modellierung innerhalb des BPM Lebenszyklus zu prüfen. Durch diese Untersuchungen können auch die relevanten Faktoren für die Phasen Implementierung, Ausführung und Optimierung bestimmt werden.

4. Evaluation der gewählten Gewichtungen

Neben der Selektion der relevanten Einflussgrößen ist auch der jeweilige Gewichtungswert relevant, da dieser bei zu hoch oder zu niedrig angesetzten Werten eine hohe Abweichung und damit einen großen relativen Fehler verursachen kann. Wieschollek berichtet aus seinem Erfahrungswert, dass Aufwand von Modellierung und Prozessdokumentation im ungefähren Verhältnis von 2:1 stehen. Dies ist jedoch weder wissenschaftlich quantifiziert, noch durch Untersuchungen belegt und könnte hier bestätigt oder widerlegt werden.

Bei großen Abweichungen der Schätzungen zum tatsächlichen Aufwand sollte auch die Methode der Zählung von Business Model Points (BMP) evaluiert und anschließend einer Überarbeitung unterzogen werden.

Kapitel 6

Zusammenfassung und Ausblick

Dieser Abschnitt dient einem kurzen Rückblick durch die vorangegangene Arbeit. Dabei werden die gewonnenen Erkenntnisse und gefundenen Lösungsansätze zusammengefasst und ein kurzer Ausblick auf weitere, zu untersuchende Aufgabenstellungen gegeben.

6.1 Fazit Forschungsstand

Im Rahmen dieser Arbeit wurden zum Bereich der Geschäftsprozessmodellierung entfernte und angrenzende Forschungsfelder im Hinblick auf Verfahren und Methoden zur Aufwandschätzung für die Prozessmodellerstellung untersucht.

Aus dem entfernten Feld des Projekt- und Change-Managements gibt es keine Methoden, die den Aufwand für die Modellierung direkt schätzen. Hier werden Verfahren für die Aufwandsschätzung von Projekten vorgestellt, welche Kosten der einzelnen Phasen aus der Gewichtung von Gesamtkosten ableiten [Frohnhoff 2009; Gadatsch 2013; Lauer 2014; Posluschny 2012].

Auch im angrenzenden Forschungsfeld der Softwareentwicklung leitet [Bankhofer und Nissen 2013] die Kosten für Modellierung nur aus den Gesamtkosten ab und gibt daher ebenfalls kein direktes Verfahren an.

Insbesondere hat im Feld der Softwareentwicklung keine Betrachtung der Modellierungsphase des BPM Lebenszyklus stattgefunden. Einige Autoren haben sich mit der Schätzung von Geschäftsprozessmodellen beschäftigt [Baklizky u. a. 2013; Kluza und Nalepa 2012; Marin und Quinteros 2014; Mishra und Kumar 2014; Rolon u. a. 2009] und betrachten beispielsweise agile und stark verbreitete Methoden, wie Function-Point (FP) und das Constructive Cost Model (COCOMO). Dabei wird allerdings davon ausgegangen, dass die für BPM zu verwendenden Modelle bereits erstellt worden sind. Daher greifen die Schätzmethode erst in der technischen Ebene [Freund und Rücker 2012] bzw. im BPM Lebenszyklus nach der Modellierung in der Implementierungsphase. Die klare Abgrenzung zwischen fachlicher und technischer Ebene in BPM Projekten scheint hier sehr verschwommen und geht ineinander über [Freund und Rücker 2012]. Dies könnte eine Ursache für das Fehlen von dedizierten Schätzmodellen für die Modellierungsphase sein.

Gerade für kleine und mittelständische Unternehmen (KMU) gibt es hier also wenige wissenschaftliche Untersuchungen zum Thema der Modellerstellung, die als Hilfestellung und Grundlagen verwendet werden können.

Viele Berater dienen bei BPM Projekten als externe Analysten und helfen bei der Prozessmodellerstellung. Diese haben dabei ihre eigenen Schätzverfahren entwickelt oder durch einen reichen Erfahrungsschatz die „Rule of Thumb“ [Jones 2007] entwickelt. Sie können daher sowohl anhand von Faktoren qualifizierte als auch Schätzungen nach Erfahrungswerten liefern [*Aufwandsschätzung für die Prozessmodellierung*].

Allerdings ist durch die steigende Anzahl an Publikationen erkennbar, dass sich die Wissenschaft immer intensiver mit dem Forschungsfeld BPM beschäftigt. Ebenfalls steigt die Relevanz von BPM in Organisationen [Horan 2011; Müller 2014].

6.2 Fazit Schätzverfahren

Aus der Vielzahl von Schätzverfahren aus der Softwareentwicklung, mit denen man die Geschäftsprozessmodellerstellung schätzen könnte, bietet COCOMO einen breiten Erfahrungsschatz und ist durch die Verbreitung in Unternehmen ein anerkanntes Verfahren. COCOMO II hat einige besondere Eigenschaften, die es von anderen Schätzmethoden unterscheidet [Frohnhoff 2009; Khatibi und Jawawi 2011].

Ferner wurde COCOMO seit der ersten Einführung im Jahre 1981 stets weiterverbessert und individualisiert [Frohnhoff 2009], wie die Adaption von ADA COCOMO und die Zählung von ObjectPoints für objektorientierte Softwareentwicklung zeigen.

Da keine Schätzmethoden existieren, die den Aufwand für die Modellerstellung in BPM dediziert schätzen, wurde in dieser Arbeit COCOMO einer weiteren Adaption unterzogen und zu BPM COCOMO entwickelt. Darüber hinaus wird der bisher nicht berücksichtigte Aufwand für die Erstellung einer Prozessdokumentation über einen zusätzlichen Skalenfaktor mit eingebracht. Das Modell bietet daher die Möglichkeit auch bei nicht vorhandenen Eingangsgrößen eine Schätzung anzufertigen, indem es eine Schätzung der Eingangsgrößen über Business Model Points (BMP) zulässt. Die Ermittlung der BMP erfolgt im Rahmen der Erstellung einer Prozessdokumentation, dessen Aufwand auch in die Aufwandsschätzung mit einfließt.

Eine Aussage über Qualität von BPM COCOMO kann erst nach erfolgter Validierung des Modells getroffen werden. Jedoch sollten, durch die Granularität des Modells und der Vielzahl an Einflussfaktoren, einzubringende Erfahrungswerte gut eingearbeitet werden. So lässt sich das Modell in allen Durchführungsschritten graduell verbessern, falls nötig.

Mit Hilfe von BPM COCOMO könnten BPM-Berater wie der Autor Wieschollek [*Aufwandsschätzung für die Prozessmodellierung*] ihre Aussagen und Erfahrungswerte zur Aufwandschätzung überprüfen (vgl. Kapitel 5.5) und dabei ihre eigenen Methoden verfeinern und anpassen.

6.3 Ausblick

Neben der Validierung und Evaluation der in dieser Arbeit entwickelten Schätzmethode, könnten weitere Untersuchungen auf Basis der Kombination von COCOMO mit anderen Schätzmodellen folgen. Ebenfalls könnten agile Schätzmethoden für die Modellierungsphase selbst entwickelt werden. Insgesamt gibt es noch keine Schätzmethode, die BPM in allen Phasen unterstützt und jede Phase dediziert schätzt. Dies könnte ein weiteres Forschungsfeld bilden.

Akronyme und Abkürzungen

BPM Business Process Management.....	5
BPMN Business Process Model and Notation.....	5
BPMM Business Process Maturity Model.....	63
BMP Business Model Points.....	84
COCOMO Constructive Cost Model	83
CMMN Case Managemen Model and Notation	16
DMN Decision Model and Notation	16
FP Funcion-Point	83
KMU kleine und mittelständische Unternehmen.....	84

Literatur

- Allweyer, Thomas (2007). *Das Business Process Maturity Model (BPMM) der OMG*.
URL: <http://www.kurze-prozesse.de/2007/11/08/das-business-process-maturity-model-bpmm-der-omg/> (siehe S. 22, 62, 67).
- (2014). *BPMS: Einführung in Business Process Management-Systeme*.
BoD-Books on Demand (siehe S. 12, 65, 66, 72).
- Baklizky, Maruschia u. a. (2013a).
„Business Process Points - A Proposal To Measure BPM Projects“.
In: *Proceedings of the 21st European Conference on Information Systems*.
ECIS 2013 Completed Research. Paper 2.
URL: http://aisel.aisnet.org/ecis2013_cr/2 (siehe S. 22, 26, 35, 37).
- Baklizky, Maruschia u. a. (2013b).
„Business Process Points-A Proposal To Measure BPM Projects.“ In: *ECIS*, S. 2
(siehe S. 22, 35, 69, 83).
- Bankhofer, Herausgegeben von U und V Nissen (2013). „A cost calculation model for
determining the cost of business process modelling projects“.
In: *Ilmenauer Beiträge zur Wirtschaftsinformatik 2013-01* (siehe S. 22, 83).
- Becker, Jörg, Michael Rosemann und Christoph von Uthmann (2000).
„Guidelines of Business Process Modeling“. English. In: *Business Process Management*.
Hrsg. von Wil van der Aalst, Jörg Desel und Andreas Oberweis. Bd. 1806.
Lecture Notes in Computer Science. Springer Berlin Heidelberg, S. 30–49.
ISBN: 978-3-540-67454-2. DOI: 10.1007/3-540-45594-9_3.
URL: http://dx.doi.org/10.1007/3-540-45594-9_3 (siehe S. 22, 67).
- Boehm, Barry W. *COCOMO II Model Definition Manual*.
URL: <http://sunset.usc.edu/research/COCOMOII/Docs/modelman.pdf>
(siehe S. 37, 50, 54, 62, 63, 71).
- Boehm, Barry W. (1981). *Software Engineering Economics*. New York: Prentice-Hall.
ISBN: 978-0-138-22122-5 (siehe S. 43).
- Boehm, Barry und Ellis Harrowitz (2000). *Software Cost Estimation with Cocomo II*.
London: Prentice Hall. ISBN: 978-0-130-26692-7
(siehe S. 37, 44, 47, 57, 61–64, 67, 72, 75, 81).
- BPMM Specification*. URL: <http://www.omg.org/spec/BPMM/1.0/> (siehe S. 63).
- BPMN Modeling Best Practices*. URL: http://www.bpmnquickguide.com/quickguide/index.html?modeling_best_practices.htm
(siehe S. 22).
- Bulsuk, Karn G. (2009). *PDCA Cycle*.
URL: <http://www.bulsuk.com/2009/02/taking-first-step-with-pdca.html> (besucht
am 2009) (siehe S. 24).

- Çulha, Davut und Ali Doğru (2014).
„Towards an Agile Methodology for Business Process Development“.
In: *S-BPM ONE-Scientific Research*. Springer, S. 133–142 (siehe S. 22, 35).
- Dowie, Ulrike (2008). „Testaufwandsschätzung in der Softwareentwicklung : Modell der Einflussfaktoren und Methode zur organisationsspezifischen Aufwandsschätzung“. ger.
Diss. Holzgartenstr. 16, 70174 Stuttgart: Universität Stuttgart. ISBN: 978-3-89936-789-8.
URL: <http://elib.uni-stuttgart.de/opus/volltexte/2009/4636> (siehe S. 62).
- Fisher, David M (2004). „The business process maturity model. A practical approach for identifying opportunities for optimization“. In: *Business Process Trends* 9.4, S. 11–15 (siehe S. 22, 62).
- Freund, Jakob und Bernd Rücker (2012). *Praxishandbuch BPMN 2.0*. 3. aktualisierte Auflage. München: Hanser. ISBN: 978-3-446-42986-4
(siehe S. 10–12, 14–16, 21, 22, 26–30, 32, 48–51, 54, 62–64, 66, 68, 71, 72, 83).
- Frohnhoff, Stephan (2009).
„Use Case Points 3.0 : Implementierung einer Use Case bezogenen Schätzmethode für das Software-Engineering betrieblicher Informationssysteme“.
Diss. Universität Paderborn
(siehe S. 25, 32, 37, 39–43, 45–47, 53, 54, 56, 61–63, 74, 75, 77, 78, 83, 84).
- Gadatsch, Andreas (2013). *Grundkurs Geschäftsprozess-Management*. 7. Auflage.
Springer Vieweg. ISBN: 978-3-8348-2427-1 (siehe S. 10, 11, 17–19, 21, 22, 24, 29, 51, 67, 83).
- Gruhn, Volker und Ralf Laue (2006). „Komplexitätsmetriken für Geschäftsprozessmodelle“.
In: *Proceedings of the Modellierung 2006*. Hrsg. von H. C. et al. Mayr.
Bonn: Gesellschaft für Informatik, S. 289–292 (siehe S. 10, 22, 36, 51, 58, 59, 64, 70).
- Herzwurm, Georg und Wolfram Pietsch (2009). *Management von IT-Produkten*. dpunkt Verlag.
ISBN: 978-3-89864-562-1 (siehe S. 24, 63).
- Hinsch, Martin (2014). *Die neue ISO 9001:2015 - Status, Neuerungen und Perspektiven* -. 1. Aufl.
Berlin Heidelberg New York: Springer-Verlag. ISBN: 978-3-662-45029-1
(siehe S. 26, 28, 29).
- Horan, Jeanette, Hrsg. (2011). *Schlüsselrolle CIO*. CIO Studienreihe.
IBM Institute for Business Value (siehe S. 10, 84).
- Jakoubi, Stefan und Simon Tjoa (2009).
„A reference model for risk-aware business process management“. In: *Risks and Security of Internet and Systems (CRiSIS)*, 2009 Fourth International Conference on. IEEE, S. 82–89 (siehe S. 22).
- Jones, Capers (2007).
Estimating Software Costs : Bringing Realism to Estimating - Bringing Realism to Estimating.
2. Aufl. Madison: McGraw Hill Professional. ISBN: 978-0-071-48300-1
(siehe S. 25, 27, 35–37, 84).
- Karner, Gustav (1993). „Resource estimation for objectory projects“.
In: *Objective Systems SF AB* 17 (siehe S. 39, 40).
- Kemsley, Sandy (2015). *BPMN, CMMN and DMN at BPMCM15*. URL:
<http://column2.com/2015/06/bpmn-cmmn-and-dmn-with-denisgagne-at-bpmcm15/>
(siehe S. 19).
- Khatibi, Vahid und Dayang N. A. Jawawi (2011).
„Software Cost Estimation Methods: A Review“.

- In: *Journal of Emerging Trends in Computing and Information Sciences* 4.12, S. 21–29.
ISSN: 2079-8407 (siehe S. 25, 36–39, 53, 81, 84).
- Khatibi, Vahid u. a. (2011). „Neural Networks for Accurate Estimation of Software Metrics“.
In: *IJACT: International Journal of Advancements in Computing Technology* 3.10, S. 54–66
(siehe S. 25, 81).
- Kleppe, Anneke G, Jos B Warmer und Wim Bast (2003).
MDA explained: the model driven architecture: practice and promise.
Addison-Wesley Professional (siehe S. 52).
- Kluza, Krzysztof und Grzegorz J Nalepa (2012).
„Proposal of square metrics for measuring business process model complexity“.
In: *Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on*. IEEE,
S. 919–922 (siehe S. 10, 22, 58, 83).
- Komus, Ayelt (2011).
BPM Best Practice - Wie führende Unternehmen ihre Geschäftsprozesse managen. 2011. Aufl.
Berlin Heidelberg New York: Springer-Verlag. ISBN: 978-3-642-16725-6 (siehe S. 22).
- Kraus, Georg, Christel Becker-Kolle und Thomas Fischer (2010). *Change-Management*.
3. Auflage. Cornelsen Verlag Scriptor (siehe S. 23).
- Lauer, Thomas (2014). *Change Management*. 2. Auflage. Berlin Heidelberg: Springer Gabler.
ISBN: 978-3-662-43736-0 (siehe S. 83).
- Lessen, Tammo van, Daniel Lübke und Jörg Nitzsche (2011).
Geschäftsprozesse automatisieren mit BPEL. German. Heidelberg: dpunkt Verlag.
ISBN: 978-3-89864-670-3. URL: <http://taval.de/publications/B00K-2011-01>
(siehe S. 16, 26).
- Leymann, Frank (2011). „Managing business processes via workflow technology“.
In: *Tutorial at VLDB Conference*. Seattle (siehe S. 10).
- Lhannaoui, Hanane, Mohammed Issam Kabbaj und Zohra Bakkoury (2013). „Towards an
approach to improve business process models using risk management techniques“.
In: *Intelligent Systems: Theories and Applications (SITA), 2013 8th International Conference on*.
IEEE, S. 1–8 (siehe S. 22, 63).
- Marin, Beatriz und Jose Quinteros (2014).
„A COSMIC Measurement Procedure for BPMN Diagrams“.
In: *The 26th International Conference on Software Engineering and Knowledge Engineering*
(siehe S. 22, 83).
- Mendling, Jan, Hajo A Reijers und Wil MP van der Aalst (2010).
„Seven process modeling guidelines (7PMG)“.
In: *Information and Software Technology* 52.2, S. 127–136 (siehe S. 22, 50, 67).
- Meziani, Rachid und Imad Saleh (2011).
„Towards a collaborative business process management methodology“.
In: *Multimedia Computing and Systems (ICMCS), 2011 International Conference on*. IEEE,
S. 1–6 (siehe S. 22, 35).
- Milutinovic, Aleksandar (2014).
Konsolidierung mittels Event-Handler kommunizierender BPEL-Prozesse. ger.
URL: <http://elib.uni-stuttgart.de/opus/volltexte/2014/9491> (siehe S. 47).
- Mishra, Shivakant und Chanchal Kumar (2014). „Estimating development size and effort of
business process service-oriented architecture applications“.

- In: *Systems and Informatics (ICSAI), 2014 2nd International Conference on*. IEEE, S. 1006–1011 (siehe S. 22, 26, 35, 83).
- Müller, Sabrina (2014). „Studienarbeit 2451- Situationsanalyse: BPM in Deutschland“. Magisterarb. Universität Stuttgart (siehe S. 10, 11, 15, 17, 21, 28, 84).
- Noth, Thomas und Mathias Kretzschmar (1986). *Aufwandschätzung von DV-Projekten: Darstellung u. Praxisvergleich d. wichtigsten Verfahren*. 2. Aufl. Berlin: Springer, X, 156 S. ISBN: 0-387-16069-8 (siehe S. 37–39, 54).
- OASIS. *BPEL 2.0 Specification*.
URL: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>
(siehe S. 8, 17, 47).
- OMG. *BPMN 2.0 Specification*. URL: <http://www.omg.org/spec/BPMN/2.0/>
(siehe S. 8, 47, 60, 61).
- Posluschny, Peter (2012). *Prozessmanagement*. Konstanz: UVK Verlagsgesellschaft mbH. ISBN: 978-3-8501-2 (siehe S. 10, 11, 22, 24, 28, 29, 32, 62, 67, 83).
- Rolon, Elvira u. a. (2009). „Prediction models for BPMN usability and maintainability“. In: *Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on*. IEEE, S. 383–390 (siehe S. 22, 83).
- Schmelzer, Hermann J. und Wolfgang Sesselmann (2008). *Geschäftsprozessmanagement in der Praxis*. 6. Auflage. München: Hanser. ISBN: 978-3-446-41002-2 (siehe S. 11, 50).
- Skouradaki, Marigianna u. a. (2014). „“BPELanon”: Anonymizing BPEL Processes“. In: *Workshop Proceedings*, S. 5 (siehe S. 74).
- Symons, C.R. (1988). „Function point analysis: difficulties and improvements“. In: *Software Engineering, IEEE Transactions on* 14.1, S. 2–11. ISSN: 0098-5589. DOI: 10.1109/32.4618 (siehe S. 55).
- Thiemich, Christian und Frank Puhlmann (2013). „An Agile BPM Project Methodology“. In: *Business Process Management*. Hrsg. von Florian Daniel, Jianmin Wang und Barbara Weber. Bd. 8094. Springer Berlin Heidelberg, S. 291–306.
URL: http://dx.doi.org/10.1007/978-3-642-40176-3_25 (siehe S. 22, 35).
- Tiemeyer, Ernst (2013). *Handbuch IT-Management*. Hrsg. von Ernst Tiemeyer. 5., überarbeitete und erweiterte Auflage. Hanser. ISBN: 978-3-446-43557-5 (siehe S. 23, 31).
- Van Der Aalst, Wil MP, Arthur HM Ter Hofstede und Mathias Weske (2003). „Business Process Management: A Survey“. In: *Business Process Management*. Springer, S. 1019–1019 (siehe S. 22).
- Weske, Mathias (2012). *Business Process Management - Concepts, Languages, Architectures*. 2nd ed. 2012. Wiesbaden: Springer Berlin Heidelberg. ISBN: 978-3-642-28615-5 (siehe S. 12, 13, 15, 18).
- „Aufwandsschätzung in IT-Projekten“ (2007). In: *Management von IT-Projekten*. Hrsg. von Hans W. Wiczorrek und Peter Mertens. Springer Berlin Heidelberg, S. 205–223. ISBN: 978-3-540-48470-7.
URL: http://dx.doi.org/10.1007/978-3-540-48472-1_8 (siehe S. 24, 25, 56).
- Wiczorrek, Hans W. und Peter Mertens (2010). *Management von IT-Projekten - Von der Planung zur Realisierung*. 4. Aufl.

Berlin Heidelberg New York: Springer-Verlag. ISBN: 978-3-642-16127-8
(siehe S. 23, 24, 73).

Wieschollek, Martin (2013). *Aufwandsschätzung für die Prozessmodellierung*.
URL: <http://www.bpm-plus.de/2013/03/aufwandsschatzung-fur-die-prozessmodellierung/> (besucht am 27.03.2013) (siehe S. 29, 74, 84).

Wikipedia. *Comparison of Business Process Modeling Notation tools*. URL: https://en.wikipedia.org/wiki/Comparison_of_Business_Process_Modeling_Notation_tools
(siehe S. 52, 62).

Alle URLs wurden zuletzt am 2015-09-17 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift