



Technische Universität Graz  
Institut für Maschinenbau- und Betriebsinformatik  
in Kooperation mit  
Siemens Transformers Austria Weiz

Masterarbeit

---

ANFORDERUNGS- UND PROZESSANALYSE IN DER  
PROJEKTVERWALTUNG

---

**Gernot R. Bauer**

Graz, Dezember 2009

*Begutachter:*

Univ.-Prof. Dipl.-Ing. Dr.techn. Siegfried Vössner

*Betreuer:*

Dipl.-Ing. Sabine Hösch  
Dipl.-Ing. Gerald Lichtenegger  
Dipl.-Ing. Martin Kalcher

*Danke für den Tee.*

## **Eidesstattliche Erklärung**

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

## **Statutory Declaration**

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, Dezember 2009

Gernot R. Bauer

# Danksagung

An dieser Stelle möchte ich mich bei all jenen bedanken, die mich im Zuge meines Studiums und bei der Erstellung dieser Arbeit unterstützt haben.

Mein Dank gilt dem Institut für Maschinenbau- und Betriebsinformatik für die Chance, meine Diplomarbeit im Rahmen eines praktischen Projekts durchführen zu können. Ich bedanke mich bei Univ.-Prof. Dipl.-Ing. Dr.techn. Siegfried Vössner und Dipl.-Ing. Gerald Lichtenegger für die Betreuung und insbesondere bei Dipl.-Ing. Sabine Hösch für ihre inhaltlichen Hilfestellungen im Zuge der Betreuung und ihren Blick fürs Detail.

Ich möchte mich auch bei der Siemens Transformers Austria Weiz bedanken, die mir einen Einblick in die Arbeitswelt und in Software-Projekte abseits der grauen Theorie ermöglicht hat. Hier gilt mein besonderer Dank Dipl.-Ing. Martin Kalcher, der meine Arbeit betreut und mit viel Einsatz zur Erstellung der Sollkonzepte beigetragen hat. Weiters möchte ich mich bei Mag. Christoph Hansal von der Firma MCP für die Unterstützung in den Workshops und das Überprüfen meiner Arbeit hinsichtlich Konsistenz und Verständlichkeit bedanken. Für die Einführung in die technischen Aspekte der PV bedanke ich mich bei Michael Gerstl und insbesondere Thomas Kober, der mir den Weg durch das Unkraut des Datengartens PV geebnet hat. Ich bedanke mich auch bei allen Mitarbeitern im Vertrieb der STA Weiz, die mit mir an der Konzeptionierung der Projektverwaltung NEU gearbeitet haben, für die freundliche Zusammenarbeit.

Zu guter Letzt möchte ich meine Studienkollegen, meine Freunde und ganz besonders meine Familie erwähnen, die mich im Zuge dieser Diplomarbeit und meines gesamten bisherigen Werdegangs maßgeblich unterstützt haben – sei es durch fachliche Gespräche bei einem Bier nach getaner Arbeit (manchmal waren's auch zwei), sei es durch eine Ablenkung von den Bits und Bytes, den Prozessen und Strategien, die mir im Laufe meines Studiums begegnet sind oder auch nur durch eine kleine Aufmunterung zur rechten Zeit: Ihr habt einen großen Anteil daran, dass ich nun hier sitze und die letzten Worte meiner Masterarbeit schreibe. Euch allen gebührt mein Dank.

# Kurzfassung

Die Tätigkeit der Anforderungsanalyse (Requirements Engineering) hat in den letzten Jahren sukzessive an Bedeutung gewonnen. Aus gutem Grund: Fehler, die in der Analyse und Definition von Anforderungen gemacht werden, sind mit fast 50% die Hauptursache für Probleme in Softwareentwicklungsprojekten. Die Kosten, Fehler in einer späten Projektphase zu beheben, können die Fehlerbehebungskosten zum Zeitpunkt der Anforderungsanalyse um den Faktor 1000 übersteigen.

Mittels der Modellierung von Informationssystemen lassen sich Geschäftsprozesse aus verschiedenen Blickwinkeln betrachten, um so einerseits die Komplexität der Analyse und Darstellung von Prozessen zu verringern, andererseits aber auch dafür zu sorgen, dass wesentliche Details in der Analyse und weiteren Modellierung ausreichend berücksichtigt werden. ARIS (Architektur integrierter Informationssysteme) ist ein Framework, das dieses Konzept der Aufspaltung in die Perspektiven Organisation, Daten, Funktionen und Prozesse verwendet.

Im Zuge dieser Arbeit wurde das in der Siemens Transformers Austria (STA) Weiz eingesetzte System „Projektverwaltung“ (PV) analysiert. Die PV ist ein System, das seit mehreren Jahren im Vertrieb der STA eingesetzt wird, durch Änderungen in Prozessen und der Organisation allerdings eine große Anzahl von Problemen aufweist. Ziel dieser Arbeit ist ein Sollkonzept für eine Neuentwicklung der PV, die die aktuell existierenden Probleme berücksichtigt. Darauf aufbauend soll ein Pflichtenheft erstellt werden, das von einem externen Softwarelieferanten umgesetzt wird.

Zur Neukonzeptionierung wurden Techniken aus dem Requirements Engineering und der Modellierung von Informationssystemen eingesetzt. Damit wurde der IST-Zustand des Systems und dessen Probleme sowie die Prozesse, die die PV unterstützen soll, analysiert. Anschließend wurde mit den zukünftigen Benutzern ein Sollkonzept für zwölf Module der PV erarbeitet, das eine optimale Unterstützung der Prozesse gewährleisten soll. Im Zuge dieser Konzeptionierung wurden etwa 50 Verbesserungspotentiale in mehreren Bereichen identifiziert.

**Schlüsselwörter:** Anforderungsanalyse, Modellierung von Informationssystemen, Geschäftsprozessmanagement, ARIS, BPMN, ERM, Projektverwaltung

# Abstract

Over the last couple of years, requirements engineering has gained importance for a number of reasons: almost 50 percent of all problems identified in software projects occur due to errors made during the analysis and definition of requirements. The costs of solving problems in advanced stages of software projects can be 1000 times higher compared to the expenses of fixing errors during the analysis of requirements phase.

By applying the methods of modeling of information systems, one has the advantage of looking at business processes from multiple points of view. The complexity of business process analysis is hence reduced, while it is moreover ensured that important details of a business process are considered as well. The ARIS (architecture of integrated information systems) framework is a tool drawing upon this specific concept of looking at a process from several different perspectives (organization, data, functions and processes).

This thesis consists of an analysis of a system called „Projektverwaltung“ (PV, project administration). It is a system that has been used in the sales department of Siemens Transformers Austria (STA) Weiz for several years. However, a number of issues concerning the system which came up because of alterations in business processes and organizational changes now have to be faced. The aim of this thesis is to define a target concept, considering all the current problems of the system. The target concept will then be used to define a requirements specification. Based on the specification, a contractor will develop a new version of the „Projektverwaltung“ system.

Various methods out of the field of requirements engineering and modeling of information systems were used during the definition of the target concept. First, the current situation and the business processes in the sales department were analyzed. Afterwards, the target concept for twelve modules of the PV was defined in cooperation with prospective users of the system. In the course of the conception phase, a catalogue of fifty suggestions for potential improvements in various different branches could be compiled.

**Keywords:** Requirements Engineering, Modeling of Information Systems, Business Process Management, ARIS, BPMN, ERM, Project Administration

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Zielsetzung der Arbeit . . . . .	1
1.2. Aufbau der Arbeit . . . . .	1
<b>I. Grundlagen</b>	<b>3</b>
<b>2. Requirements Engineering</b>	<b>4</b>
2.1. Grundlagen des Requirements Engineerings . . . . .	4
2.1.1. Motivation . . . . .	4
2.1.2. Requirements Engineering als Kostenfaktor . . . . .	7
2.1.3. Definition . . . . .	8
2.2. Requirements Engineering-Prozess . . . . .	9
2.2.1. Requirements Engineering im Softwareentwicklungsprozess . . . . .	9
2.2.2. Stakeholder im Requirements Engineering . . . . .	12
2.2.3. Anforderungsanalyse . . . . .	12
2.2.4. Probleme und Lösungen im Requirements Engineering . . . . .	13
2.2.5. Lasten- und Pflichtenheft . . . . .	14
2.3. Anforderungen . . . . .	15
2.3.1. Definition . . . . .	15
2.3.2. Anforderungsarten . . . . .	15
2.3.3. Ziele . . . . .	16
2.3.4. Szenarien . . . . .	18
2.3.5. Dokumentation von Szenarien . . . . .	19
2.4. Techniken im Requirements Engineering . . . . .	20
2.4.1. Gewinnung von Anforderungen . . . . .	20
2.4.2. Gewinnungstechniken . . . . .	22
2.4.3. Validierungstechniken . . . . .	24
<b>3. Modellierung von Informationssystemen</b>	<b>26</b>
3.1. Modellierung von Geschäftsprozessen . . . . .	26
3.1.1. Motivation . . . . .	26
3.1.2. Geschäftsprozesse . . . . .	27
3.1.3. ARIS . . . . .	29
3.2. Perspektiven und Notationen . . . . .	30
3.2.1. Organisationsperspektive . . . . .	30

3.2.2.	Datenperspektive . . . . .	31
3.2.3.	Funktionsperspektive . . . . .	33
3.2.4.	Prozessperspektive . . . . .	33
<b>II.</b>	<b>Aufgabenstellung und Umsetzung</b>	<b>40</b>
<b>4.</b>	<b>Problemstellung</b>	<b>41</b>
4.1.	Einleitung . . . . .	41
4.2.	Siemens Transformers Austria . . . . .	41
4.3.	Projekt PPS 15.000 MVA+ . . . . .	42
4.3.1.	Projektziele und -vorgaben . . . . .	42
4.3.2.	Zukünftige Systemlandschaft . . . . .	43
4.3.3.	Projektorganisation . . . . .	44
4.4.	Aufgabenstellung . . . . .	44
<b>5.</b>	<b>Projektphasen</b>	<b>47</b>
5.1.	Einarbeitungsphase . . . . .	48
5.1.1.	Zielstellung . . . . .	48
5.1.2.	Vorgangsweise . . . . .	48
5.1.3.	Ergebnis . . . . .	49
5.2.	IST-Analysephase . . . . .	49
5.2.1.	Zielstellung . . . . .	49
5.2.2.	Vorgangsweise . . . . .	50
5.2.3.	Ergebnis . . . . .	50
5.3.	Grobspezifikationsphase . . . . .	50
5.3.1.	Zielstellung . . . . .	50
5.3.2.	Vorgangsweise . . . . .	51
5.3.3.	Ergebnis . . . . .	52
5.4.	Feinspezifikationsphase . . . . .	53
5.4.1.	Zielstellung . . . . .	53
5.4.2.	Vorgangsweise . . . . .	53
5.4.3.	Ergebnis . . . . .	53
5.5.	Nachbereitungsphase . . . . .	53
<b>III.</b>	<b>Ergebnis</b>	<b>55</b>
<b>6.</b>	<b>IST-Analyse</b>	<b>56</b>
6.1.	PV im Systemkontext . . . . .	56
6.1.1.	Technische Eckdaten der PV . . . . .	56
6.1.2.	Schnittstellen zu Drittsystemen . . . . .	57
6.2.	PV im Organisationskontext . . . . .	58
6.3.	Funktionalität der PV . . . . .	58

6.4. Daten innerhalb der PV . . . . .	60
6.5. Prozesse innerhalb der PV . . . . .	60
<b>7. Fallbeispiel: Modul Kalkulation</b>	<b>64</b>
7.1. IST-Zustand . . . . .	64
7.1.1. Übersicht . . . . .	64
7.1.2. Arten der Kalkulation . . . . .	64
7.1.3. Struktur der Kalkulation . . . . .	65
7.1.4. Externe Berechnungen . . . . .	66
7.1.5. Probleme . . . . .	70
7.2. SOLL-Konzept . . . . .	70
7.2.1. Ziele . . . . .	70
7.2.2. Rollen . . . . .	73
7.2.3. Prozess . . . . .	75
7.2.4. Szenarien . . . . .	76
7.2.5. Kalkulation für CRT/CRS/LAL freigeben . . . . .	76
7.2.6. Anforderungen . . . . .	79
7.2.7. Verbesserungspotentiale . . . . .	82
<b>IV. Abschluss</b>	<b>84</b>
<b>8. Fazit und Ausblick</b>	<b>85</b>
<b>V. Anhang</b>	<b>87</b>
<b>A. Literatur</b>	<b>88</b>
<b>B. Abbildungsverzeichnis</b>	<b>92</b>
<b>C. Tabellenverzeichnis</b>	<b>94</b>
<b>D. Diagramme</b>	<b>95</b>
D.1. Funktionalität der PV . . . . .	96
D.2. IST-Analyse der Datenstrukturen der PV . . . . .	97
D.3. Auftragsakquisitionsprozess . . . . .	98

# 1. Einleitung

## 1.1. Zielsetzung der Arbeit

Die Firma Siemens Transformers Austria (STA) verwendet im Sales-Bereich von Großtransformatoren seit vielen Jahren ein selbstentwickeltes Projektverwaltungssystem (PV) mit integriertem Kalkulationsmodul zur Angebotskalkulation/Auftragskalkulation. Durch Änderungen in der Organisation und dem Prozessablauf ergaben sich mit diesem System über die Jahre mehrere Probleme, die von den an dem System beteiligten Personen identifiziert wurden. Mehrere interne Gutachten kamen zu dem Schluss, dass eine Weiterentwicklung der PV zum jetzigen Zeitpunkt nicht mehr sinnvoll und wirtschaftlich ist. Deshalb möchte die STA einen externen Anbieter mit einer Neuentwicklung beauftragen, wobei als Realisierungstermin der 1. April 2010 geplant ist.

Im Rahmen dieser Masterarbeit soll der Funktionsumfang sowie die Struktur der bestehenden Applikation ermittelt und dokumentiert werden. Darauf aufbauend soll ein Sollkonzept erstellt werden, das die Kritikpunkte der Analyse des bisherigen Systems berücksichtigt und eine optimale Benutzerunterstützung gewährleistet. Dieses Sollkonzept dient in weiterer Folge als Basis zur Pflichtenhefterstellung.

## 1.2. Aufbau der Arbeit

Diese Arbeit ist in mehrere Kapitel gegliedert, die unterschiedliche Aspekte des Themas behandeln.

Kapitel I beschreibt die der Arbeit zugrunde liegenden theoretischen Grundlagen. Abschnitt 2 beschreibt den Begriff des Requirements Engineerings und den Requirements Engineering-Prozess sowie Techniken, die bei der Analyse der Anforderungen eingesetzt werden können. Abschnitt 3 beschreibt die Modellierung von Informationssystemen, das ARIS-Framework sowie die eingesetzten Notationen.

Kapitel II befasst sich mit dem praktischen Teil dieser Arbeit. Abschnitt 4 beschreibt das Unternehmen Siemens Transformers Austria, das Projekt, im Zuge dessen diese Arbeit durchgeführt wurde, sowie die detaillierte Aufgabenstellung zur Erstellung des Sollkonzepts. Abschnitt 5 listet die Phasen des Projekts auf und beschreibt die gewählte Vorgangsweise.

## *1.2. Aufbau der Arbeit*

Kapitel III präsentiert Ausschnitte der Ergebnisse. Abschnitt 6 beschreibt das Ergebnis der IST-Analyse, Abschnitt 7 zeigt das Sollkonzept am Fallbeispiel Kalkulation.

In Abschnitt 8 wird die Arbeit schlussendlich zusammengefasst und ein Ausblick gewagt. Danach folgt der Anhang mit Literatur-, Abbildungs-, Tabellen- und Abkürzungsverzeichnis sowie Diagrammen, die im Fließtext keinen Platz gefunden haben.

**Teil I.**

**Grundlagen**

## 2. Requirements Engineering

### 2.1. Grundlagen des Requirements Engineerings

#### 2.1.1. Motivation

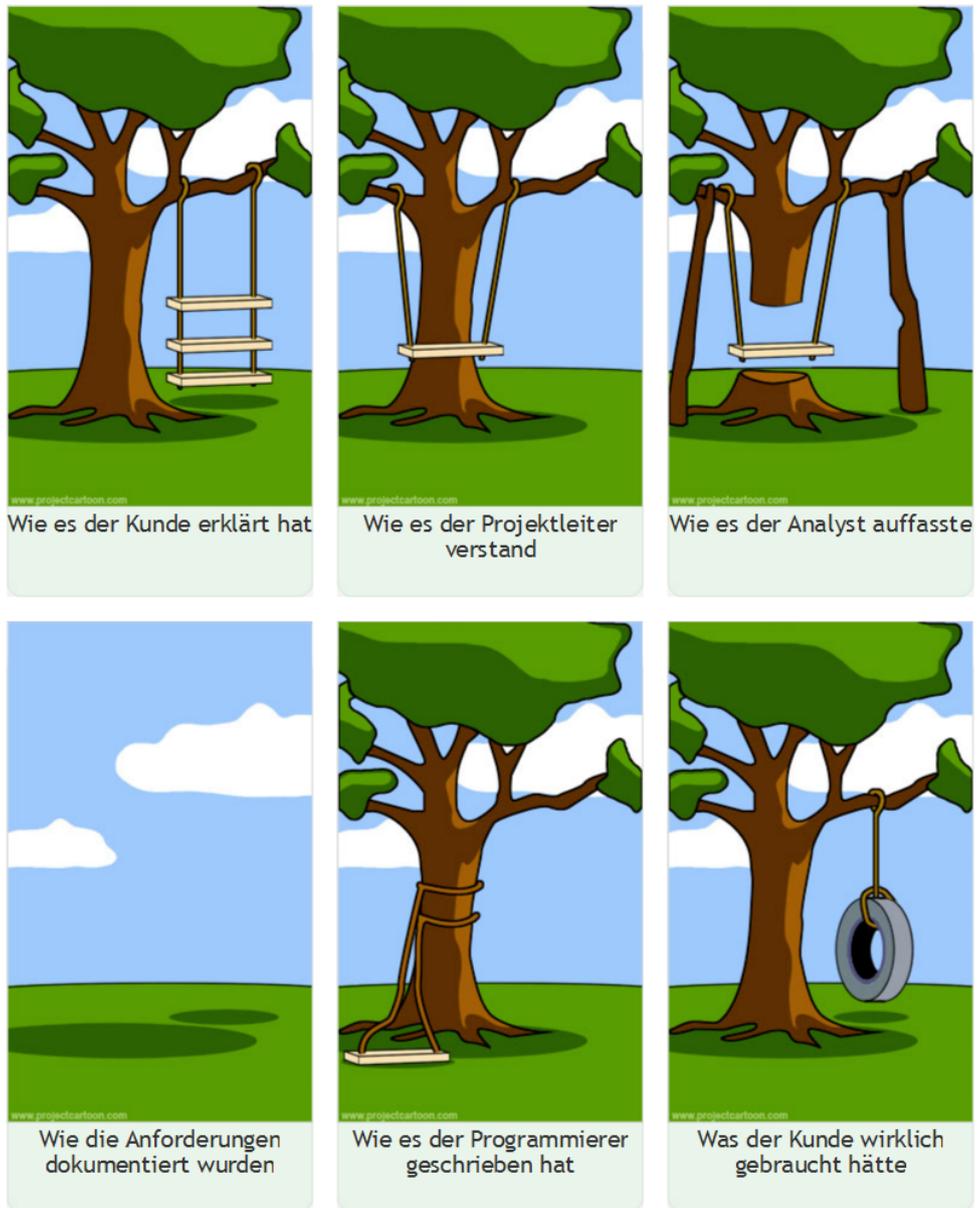
Das Erfassen von Anforderungen ist eine der Grundlagen der Softwareentwicklung. Fehler und Missverständnisse zu Beginn des Softwareentwicklungsprozesses können dazu führen, dass das eigentlich gewünschte Ergebnis erheblich vom realisierten abweicht. Abbildung 2.1 stellt diese Problematik humoristisch dar.

Hierzu ein paar Zahlen: Eine Studie der Standish Group untersuchte die Gründe für das Scheitern von Softwareprojekten und das Nicht-Einhalten von Zeit- und Kostenrahmen (vgl. The Standish Group 1995 S. 4). Dabei zeigte sich, dass über 48 % aller Probleme durch Faktoren entstehen, die dem Bereich Requirements Engineering zuzuordnen sind (siehe Abbildung 2.2). Eine weitere Studie, die in zwölf Software-Unternehmen durchgeführt wurde, führte ebenfalls mit 48 % Unzulänglichkeiten beim Erheben der Anforderungen als häufigstes Problem im Softwareentwicklungsprozess an (vgl. Hall et al. 2002 S. 153 ff.). Eine Aufschlüsselung dieser Faktoren findet sich in Tabelle 2.1.

Der Fall des “London Ambulance Service (LAS) Computer Aided Despatch (CAD) Systems” zeigt die Auswirkungen von Fehlern in der Anforderungsspezifikation besonders deutlich auf: Das System sollte dazu eingesetzt werden, die Bearbeitung von Notrufen durch teilweise Automatisierung zu erleichtern, indem das System feststellt, welche Ambulanzfahrzeuge sich in der Nähe des Unglücksortes befinden. Durch Fehler in der Anforderungsanalyse endete der Einsatz des Systems jedoch in chaotischen Zuständen. In den Anforderungen wurden Funklöcher, in denen Ambulanzfahrzeuge nicht erreichbar waren, nicht berücksichtigt. Stattdessen ging die Spezifikation explizit von nahezu perfekten Statusinformationen von Fahrzeugpositionen und -bereitschaft aus. Darüber hinaus waren die Benutzerschnittstellen in den Ambulanzfahrzeugen fehlerhaft konzipiert.

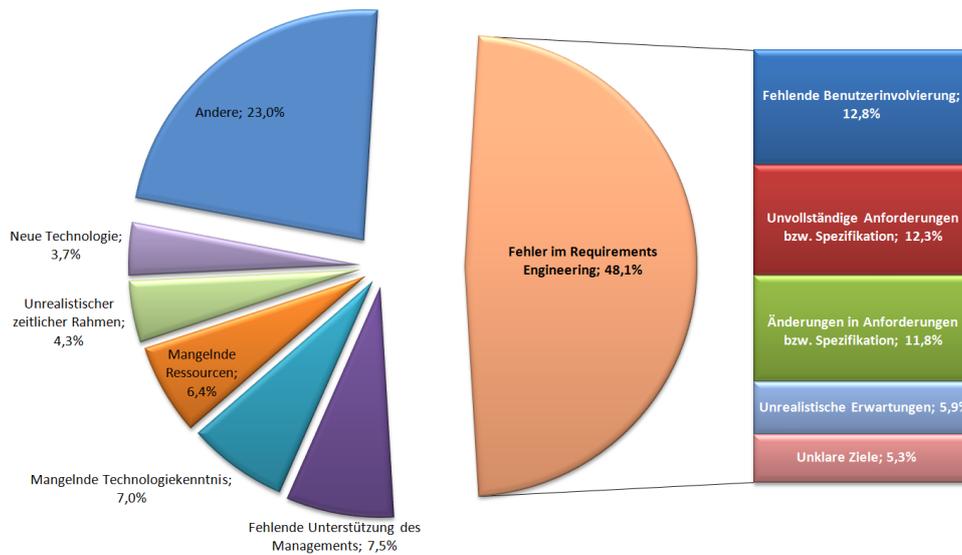
Dies führte zu einer inkonsistenten Datenbasis und in Konsequenz dazu, dass zu manchen Notfällen mehr Ambulanzfahrzeuge als benötigt beordert wurden, während bei anderen Notfällen erst Stunden später ein Rettungswagen eintraf. Durch diese Probleme wurde das System zuerst nur mehr teilweise verwendet, wenig später erfolgte der Rückschritt auf die manuelle Abarbeitung

## 2.1. Grundlagen des Requirements Engineerings



**Abbildung 2.1.:** Eine humoristische Darstellung der Problematik in Softwareprojekten (Project Cartoon und Bauer 2009).

## 2.1. Grundlagen des Requirements Engineerings



**Abbildung 2.2.:** Gründe für das Scheitern oder Nicht-Einhalten von Zielvorgaben in Software-Projekten (Daten aus The Standish Group 1995 S. 4).

von Notrufen (vgl. Finkelstein und Dowell 1996 S. 2 ff.; Pohl 2008 S. 10).

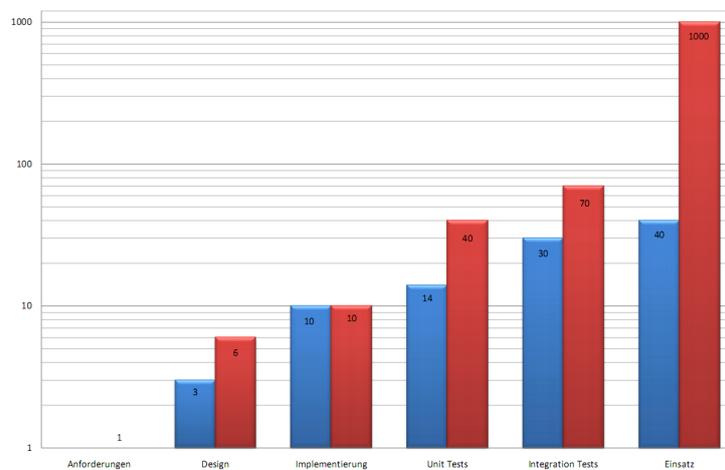
Das Fallbeispiel zeigt die Auswirkungen, die Fehler in der Analyse haben können. Mangelnde oder fehlerhafte Anforderungsspezifikationen können zu signifikanten Kostensteigerungen in Softwareprojekten führen, bis hin zur Einstellung eines Projekts. Die Herausforderung, komplexe und qualitativ hochwertige Softwareprodukte mit großem Funktionsumfang zu entwickeln, bedarf einer genauen Analyse von Anforderungen. Dies führt dazu, dass Requirements Engineering im Softwareentwicklungsprozess steigende Bedeutung zugemessen wird (vgl. Pohl 2008 S. 11).

Kategorie	Prozent
Vage anfängliche Anforderungen	25,0%
Undefinierter Anforderungsprozess	24,2%
Anforderungszuwachs	23,5%
Applikationskomplexität	20,5%
Schlechtes Verständnis des Benutzer	3,8%
Schlechte Rückverfolgbarkeit von Anforderungen	3,0%

**Tabelle 2.1.:** Probleme im Anforderungsprozess (vgl. Hall et al. 2002 S. 155)

### 2.1.2. Requirements Engineering als Kostenfaktor

Mängel in der Anforderungsspezifikation sind ein nicht zu unterschätzender Kostenfaktor im Softwareentwicklungsprozess: Etwa die Hälfte der Fehler im Programmquelltext sind auf fehlerhafte Anforderungen zurückzuführen. Die Kosten, diese Fehler zu beheben, werden umso höher, je weiter die Entwicklung eines Projekts bereits fortgeschritten ist (vgl. Möller 1996; zit. nach Pohl 2008 S. 10). Kleuker spricht in diesem Zusammenhang plakativ von “Garbage in, Garbage out”: Fehler, die bereits in der Anforderungsanalyse gemacht wurden, können in der Regel im weiteren Projektverlauf nicht ausgeglichen werden, auch nicht durch besondere Qualität in späteren Projektphasen (vgl. Kleuker 2009 S. 51).



**Abbildung 2.3.:** Relative Fehlerbehebungskosten in verschiedenen Projektphasen (Daten aus Boehm 1981; zit. nach Banga et al. 2003 S. 123 f.). Das Diagramm stellt den Bereich dar, in dem sich die Fehlerbehebungskosten relativ zur Anforderungsphase befinden. So verursachen Änderungen von Fehlern im Betrieb Kosten, die um den Faktor 40 bis 1000 höher liegen als die Behebung in der Anforderungsphase. Aus Gründen der Darstellung ist die Ordinate logarithmisch skaliert.

Abbildung 2.3 zeigt die relativen Kosten, die die Fehlerbehebung in verschiedenen Projektphasen verursachen können. Wird ein Fehler erst nach der Auslieferung entdeckt, so können die Kosten um den Faktor 1000 höher liegen, als wenn der Fehler schon in der Anforderungsphase erkannt worden wäre (vgl. Boehm 1981; zit. nach Banga et al. 2003 S. 123 f.). Aktuellere Untersuchungen sprechen von einem Faktor um 100:1 bei kritischen Systemen. Bei kleineren und unkritischen Systemen, die mittels iterativer Methoden entwickelt werden, liegt der Faktor mit 5:1 tiefer. Dieser kleinere Faktor kann durch modulare Architekturen auch in großen Systemen erreicht werden (vgl. Boehm 1987 S. 1; Boehm und Basili 2001 S. 1).

Auch agile Entwicklungsmethoden wie das *Extreme Programming (XP)* erkennen dieses Problem an und versuchen, diese Kosteneskalation durch stetiges Prototyping und das zeitnahe Umsetzen neuer Anforderungen so gering wie möglich zu halten und dadurch eine abgeflach-

## 2.1. Grundlagen des Requirements Engineerings

te Kostenkurve zu erreichen. Durch das kontinuierliche Einarbeiten kleiner Änderungen und häufiger Releases wird dabei versucht, Fehler, die durch Mängel in der Anforderung entstehen, frühzeitig erkennen und korrigieren zu können (vgl. Beck 2000 S. 23 ff.). Das Credo von XP, eine frühzeitige und möglichst vollständige Erfassung von Anforderungen durch sogenannte *User Stories* zu ersetzen, die sich erst nach und nach im Entwicklungsprozess ergeben, stößt allerdings auch auf Kritik. Hierbei besteht die Gefahr, dass durch lückenhafte Anforderungen auch das Softwaredesign Lücken aufweist, die in weiterer Folge schwierig zu beheben sind. Darüber hinaus können viele Anforderungen schon in einem frühen Entwicklungsstadium richtig erhoben werden, weshalb eine absichtliche Verzögerung dieser Erhebung oft nicht zielführend ist (vgl. Stephens und Rosenberg 2003 S. 238).

### 2.1.3. Definition

Der Begriff *Requirements Engineering* besitzt in der Literatur keine einheitliche Definition. Partsch unterscheidet zwischen Requirements Engineering im engeren und im weiteren Sinn. *Das Wort [Requirements Engineering, Anm.] steht einmal für alle konkreten Aktivitäten am Beginn einer Systementwicklung, die auf eine Präzisierung der Problemstellung abzielen. Ebenso steht es aber auch für eine ganze Teildisziplin im Grenzbereich zwischen Informatik und Anwendungswissenschaften* (Partsch 1998 S. 17).

Laut Broy et al. zielt Requirements Engineering *auf die systematische Erhebung der Anforderungen für ein zu entwickelndes Produkt oder System auf Basis genereller Geschäftsziele und Vorgaben ab. Angestrebt werden dokumentierte Anforderungen an Produkte oder Systeme, die optimal nutzbar und vermarktbar sind und in jeder Hinsicht möglichst gut den Erwartungen aller Beteiligten entsprechen* (Broy et al. 2007 S. 128)

Pohl verwendet folgende Definition für Requirements Engineering: *Das Requirements Engineering ist ein kooperativer, iterativer, inkrementeller Prozess, dessen Ziel es ist zu gewährleisten, dass*

1. *alle relevanten Anforderungen bekannt und in dem erforderlichen Detaillierungsgrad verstanden sind,*
2. *die involvierten Stakeholder eine ausreichende Übereinstimmung über die bekannten Anforderungen erzielen,*
3. *alle Anforderungen konform zu den Dokumentationsvorschriften dokumentiert bzw. konform zu den Spezifikationsvorschriften spezifiziert sind.* (Pohl 2008 S. 43)

In Anlehnung an diese Definitionen bezeichnet Requirements Engineering im Kontext dieser Arbeit einen Prozess in der Softwareentwicklung zur

## 2.2. Requirements Engineering-Prozess

1. Gewinnung,
2. Dokumentation und
3. Validierung

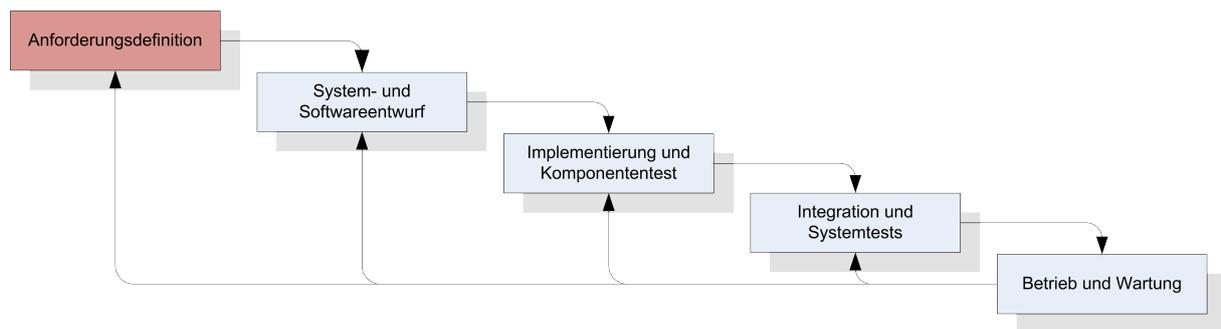
relevanter Anforderungen. Ziel dieses Prozesses ist es, Stakeholdern Anforderungen und dazu gehörende Informationen in einem geeigneten Format zugänglich zu machen.

Der Begriff *Anforderungsanalyse* ist mit dem Begriff Requirements Engineering gleichbedeutend (vgl. Sommerville 2001 S. 131).

## 2.2. Requirements Engineering-Prozess

### 2.2.1. Requirements Engineering im Softwareentwicklungsprozess

#### Wasserfall-Modell



**Abbildung 2.4.:** Das Wasserfall-Modell (nach Sommerville 2001 S. 58). Requirements Engineering lässt sich in diesem Modell am ehesten der ersten Phase der Anforderungsdefinition zuordnen.

Anforderungen stellen den Ausgangspunkt des Softwareentwicklungsprozesses dar und beschreiben die Problemstellung, die es zu lösen gilt.

Im klassischen *Wasserfall-Modell* (auch Softwarelebenszyklus genannt) ist die Analyse und Definition von Anforderungen die erste von fünf Projektphasen (siehe Abbildung 2.4):

**Anforderungsdefinition:** Definition von Anforderungen und Zielen des Systems in Zusammenarbeit mit dem Systembenutzer. In weiterer Folge wird daraus die Systemspezifikation erstellt.

**System- und Softwareentwurf:** Beschreibung der Architektur des Softwaresystems auf Basis der Spezifikation.

**Implementierung und Komponententest:** Übersetzen des Entwurfs in ein oder mehrere Programme. Tests werden auf Basis der Einzelkomponenten durchgeführt.

**Integration und Systemtests:** Zusammenführen der einzelnen Programme und Test des System als Ganzes. Danach erfolgt die Auslieferung an den Kunden.

**Betrieb und Wartung:** Korrigieren von Fehlern, Verbesserung der Implementierung und Integration eventuell entstehender neuer Anforderungen.

Das Wasserfall-Modell basiert auf dem Prinzip, dass aus jeder Phase ein oder mehrere Dokumente hervorgehen, die abgenommen werden. Danach ist die Phase abgeschlossen und die nächste Phase beginnt. In der Praxis wird dieser lineare Prozess generell nicht vollständig eingehalten. Stattdessen überlappen sich Phasen zumeist und wirken auch auf die Vorgängerphase ein. So können Probleme mit Anforderungen, die in der Entwurfsphase entdeckt werden, eine Aktualisierung des Anforderungsdokumentes nach sich ziehen (vgl. Sommerville 2001 S. 57 f.).

### Formale Systementwicklung

Einen Spezialfall des Wasserfall-Modells stellt die *formale Systementwicklung* dar. Dabei werden Anforderungen wie im Wasserfallmodell gewonnen. Diese Anforderungen werden in ein formales, mathematisches Modell übertragen, das in weiterer Folge in ein ausführbares Programm umgewandelt wird. Dieser Ansatz wird hauptsächlich bei Systemen angewandt, die strengen Sicherheits- und Zuverlässigkeitsanforderungen unterliegen. Außerhalb dieser Gebiete bietet dieser Ansatz keine nennenswerten Vorteile (vgl. Sommerville 2001 S. 60 f.).

### Evolutionäre Entwicklung

In der *evolutionären Entwicklung* werden Prototypen entwickelt, mit deren Hilfe weitere Anforderungen gewonnen werden. Mit diesem Ansatz wird die Spezifikation nach und nach auf Basis dieser Prototypen erstellt und können so den unmittelbaren Bedürfnissen des Kunden angepasst werden.

Durch die stetigen Veränderungen besteht allerdings die Gefahr, dass das resultierende System schlecht strukturiert ist. Dieser Ansatz ist geeignet für kleine und mittlere Systeme. Große Systeme mit langer Lebensdauer können von diesem Ansatz allerdings profitieren, indem unklare Teile mittels Wegwerf-Prototypen spezifiziert werden, um so die Anforderungen des Kunden genauer zu erforschen (vgl. Sommerville 2001 S. 58 ff.).

### **Inkrementelle Entwicklung**

Bei der *inkrementellen Entwicklung* wird vom Kunden in einem ersten Schritt grob definiert, welche Leistungen das zu erstellende System bieten soll. Diese Grobdefinition wird in Teilsysteme aufgeteilt, die nach ihrer vom Kunden festgelegten Priorität bearbeitet werden.

Genauere Anforderungen werden erst spezifiziert, wenn das jeweilige Teilsystem bearbeitet wird. Der für ein Teilsystem eingesetzte Entwicklungsprozess kann je nach Problemstellung gewählt werden. Es ist nicht notwendig, für alle Teilsysteme denselben Entwicklungsprozess zu wählen.

Die inkrementelle Entwicklung erzielt schon früh Ergebnisse, da die wichtigsten Teilsysteme zu Beginn ausgeliefert werden. Zudem kann die Erfahrung in der Spezifikation erster Teilsysteme dazu verwendet werden, um Probleme bei späteren Teilsystemen zu vermeiden. Dazu bedarf es relativ kleiner Teilsysteme, die jeweils eine bestimmte Systemfunktion für sich erfüllen. Auch besteht die Gefahr, dass Funktionen, die in allen Teilen benötigt werden, nicht erkannt werden, da zu Beginn keine detaillierte Spezifikation aller Anforderungen erfolgt (vgl. Sommerville 2001 S. 64 f.).

### **Agile Entwicklung**

*Agile Entwicklung* wie das in Abschnitt 2.1.2 erwähnte Extreme Programming stellen eine Weiterentwicklung der inkrementellen Entwicklung dar (vgl. Sommerville 2001 S. 65). In agilen Entwicklungsmethoden verlaufen Anforderungsanalyse, Design und Entwicklung parallel (vgl. Sommerville 2005 S. 19). Hierbei werden kurze Releasezyklen angestrebt. Anforderungen werden dabei in User Stories festgehalten, die in Aufgabenpakete unterteilt und so schnell wie möglich in das System integriert und getestet werden. Durch konstantes Refactoring wird die Systemarchitektur laufend angepasst (vgl. Beck 2000 S. 53 f.).

Anforderungen werden mit einem on-site-customer analysiert. Dabei erfolgt eine Grobplanung im Voraus, Detailplanungen werden allerdings kurzfristig vorgenommen (vgl. Beck 2000 S. 85 f.). Durch seine Besonderheiten und gewisse Einschränkungen ist Extreme Programming hauptsächlich für die Wartungsphase in Softwareprojekten geeignet (vgl. Stephens und Rosenberg 2003 S. 381).

Ein praxisorientiertes Framework zum Einsatz agiler Methoden in Softwareprojekten ist Scrum. Scrum basiert auf den Techniken von Extreme Programming und übernimmt diese in unterschiedlichem Ausmaß. Kern von Scrum sind Sprints, Zyklen von einer Dauer von etwa drei bis sechs Wochen. Die Sprints beginnen mit einem Planungsmeeting, in dem vom Kunden gewünschte Änderungen priorisiert werden und eine Aufwandsabschätzung erfolgt. Auf Basis dieser Planung erfolgt die Implementierung von Änderungen und Erweiterungen. Am Ende des Sprints steht immer ein neuer Release der entwickelten Software (vgl. Kniberg 2007 S. 7 f., 13 f.).

### 2.2.2. Stakeholder im Requirements Engineering

Ein Stakeholder im Rahmen des Requirements Engineering ist jemand, der ein potentielles Interesse am zukünftigen System hat. Daher sind Stakeholder nicht nur zukünftige Systembenutzer, sondern auch solche, die nur indirekt mit dem System zu tun haben werden, aber die Anforderungen von Systemen beeinflussen können. So kann der IT-Strategieverantwortliche bestimmte Anforderungen an das System stellen, ohne direkt in die Verwendung involviert zu sein. Ein Stakeholder verfügt in der Regel über Wissen, das im Rahmen des Projekts relevant sein kann (vgl. Pohl 2008 S. 65 f.). *Typische Beispiele für Stakeholder sind Auftraggeber, Systementwickler, Systemnutzer, Architekten, Domänenexperten, Tester oder Wartungspersonal* (Pohl 2008 S. 65).

### 2.2.3. Anforderungsanalyse

Die Anforderungsanalyse ist ein Vorgang, der alle notwendigen Aufgaben der Erstellung und Wartung eines Systempflichtenhefts umfasst. Es gibt vier allgemeine, auf einer höheren Ebene stattfindenden Aktivitäten bei der Anforderungsanalyse (Sommerville 2001 S. 131). Diese vier Aktivitäten sind in Abbildung 2.5 dargestellt.

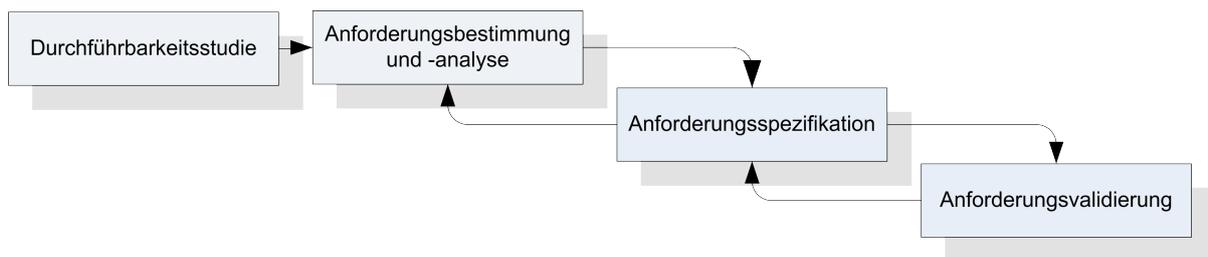


Abbildung 2.5.: Der Ablauf der Anforderungsanalyse (nach Sommerville 2001 S. 132).

Nach Sommerville besteht die Anforderungsanalyse aus vier Aktivitäten:

**Durchführbarkeitsstudie** Die Durchführbarkeitsstudie umfasst eine Grobbeschreibung des geplanten Systems sowie dessen Verwendung. Hierbei werden sowohl wirtschaftliche Aspekte als auch technische Realisierungsmöglichkeiten betrachtet. Ziel der Durchführbarkeitsstudie ist es, auf Basis dieser Betrachtungen eine Empfehlung über die Weiterentwicklung des Systems zu geben sowie eventuelle Veränderungen am zukünftigen System vorzuschlagen.

**Anforderungsbestimmung** In dieser Phase werden in Zusammenarbeit mit den zukünftigen Benutzern des Systems der Anwendungsbereich und die zu erfüllende Funktionalität des Systems bestimmt. An dieser Phase sind auch Personen beteiligt, die mit dem zukünftigen System nur indirekt zu tun haben. Es existieren mehrere unterstützende Techniken für diese Phase, wie zum

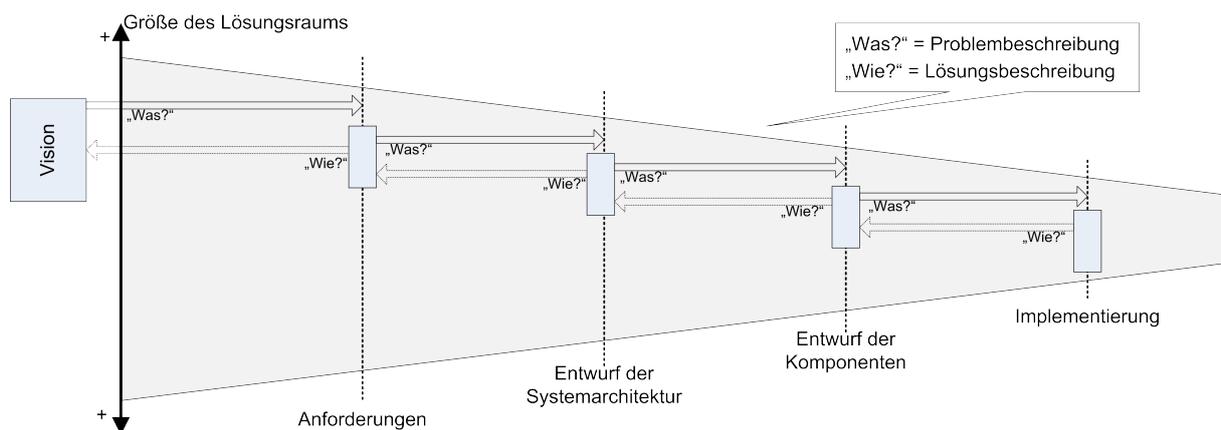
## 2.2. Requirements Engineering-Prozess

Beispiel der Einsatz von Szenarien (Abschnitt 2.3.4) oder die Viewpoint-orientierte Bestimmung. Für Details zur Bestimmung der Anforderungen siehe auch Abschnitt 2.4.1.

**Anforderungsspezifikation** Diese Phase beschreibt die Ausarbeitung und Dokumentation der Anforderungen, die sich bei der Anforderungsbestimmung ergeben haben.

**Validierung** Mit der Validierung wird überprüft, ob die spezifizierten Anforderungen den Wünschen des Kunden entsprechen. Hierzu können verschiedene Aspekte betrachtet werden, wie beispielsweise die Konsistenz und Vollständigkeit von Anforderung. Zur Durchführung der Validierung existieren verschiedene Techniken: Beim *Anforderungsreview* werden die Anforderungen von mehreren Projektteilnehmern auf Fehler überprüft. Dieser Prozess kann informell durch Diskussion mit mehreren Projektteilnehmern durchgeführt werden oder formell mit einem Walkthrough durch die Anforderungen (vgl. Sommerville 2001 S. 132–149). Für Details zur Validierung, siehe auch Abschnitt 2.4.3.

### 2.2.4. Probleme und Lösungen im Requirements Engineering



**Abbildung 2.6.:** Probleme und Lösungen im Requirements Engineering (nach Pohl 2008 S. 21). Die Lösung eines Prozessschrittes dient als Problem-Input für den nächsten Prozessschritt. Durch die Festlegung auf eine bestimmte Lösung verringert sich dadurch die Größe des Lösungsraumes sukzessive.

Ein typischer Softwareentwicklungsprozess alterniert zwischen Problemen und Lösungen. Die Lösung eines Prozessschrittes dient als Problem-Input für den nächsten Prozessschritt (siehe Abbildung 2.6). Die Unterscheidung zwischen Problem und Lösung muss aus der Sichtweise eines bestimmten Stakeholders getroffen werden: Für einen Requirements Engineer stellen die gesammelten und dokumentierten Anforderung die Lösung dar. Für einen Softwarearchitekten

## 2.2. Requirements Engineering-Prozess

sind diese Anforderungen wiederum das Problem, das er mittels eines geeigneten Systementwurfs zu lösen versucht.

Mit jedem Schritt in diesem Prozess verringert sich der Abstraktionsgrad. Aus einer abstrakten Systemvision entstehen konkretere Anforderungen, diese werden durch die Systemarchitektur weiter konkretisiert. In jedem Schritt verkleinert sich durch das Festlegen auf eine bestimmte Lösung auch der Lösungsraum. Am Ende des Prozesses erfolgt durch die Implementierung des Systems die finale Konkretisierung.

Derartige Konkretisierungsschritte können auch innerhalb des Requirements Engineerings stattfinden. So kann zum Beispiel die Anforderung “Das System soll sicher sein” einen erste Lösung darstellen. In weiterer Folge wird diese Anforderung als Problem aufgefasst und durch zusätzliche Anforderungen konkretisiert, z.B. “Benutzer müssen sich für das System authentifizieren”, “Der Datenaustausch soll verschlüsselt erfolgen” (vgl. Pohl 2008 S. 20 ff.).

### 2.2.5. Lasten- und Pflichtenheft

Das Lastenheft enthält die Systemvision und beschreibt das gewünschte System auf einer hohen Abstraktionsebene. Das Lastenheft beschreibt die gewünschten Basisanforderungen des Systems, wobei die Konzentration auf die fundamentalen Eigenschaften des Systems gelegt wird. Darüber hinaus enthält es oft eine Auflistung relevanter Stakeholder und Dokumente. Es wird als Grundlage zur Erstellung des Pflichtenheftes verwendet und kann auch als Basis von Ausschreibungen und Verhandlungen verwendet werden (vgl. Balzert 1996 S. 57 f.; Pohl 2008 S. 232 f.).

Das Pflichtenheft ist das Ergebnis der Definition und Dokumentation von Anforderungen und beschreibt die Anforderungen und Rahmenbedingungen, die vom Entwickler des Systems gefordert werden. Andere Namen für das Pflichtenheft sind Produktdefinition oder Produktspezifikation. Das Pflichtenheft richtet sich an verschiedene Benutzergruppen wie z.B. die Endbenutzer und Softwareentwickler und -warter.

Das Pflichtenheft ist die Grundlage für den weiteren Entwicklungsprozess. Ausgehend von diesem Dokument wird das Projekt geplant und die grundlegende Software-Architektur erstellt. Es schlüsselt auf, was implementiert werden muss und dient so auch als Basis für Tests und die Systemabnahme (vgl. Balzert 1996 S. 104 ff.; Sommerville 2001 S. 125 ff.; Pohl 2008 S. 234 ff.).

Zur Gliederung eines Pflichtenheftes existieren verschiedene Vorschläge und Standards, so zum Beispiel der IEEE/ANSI-Standard 830-1998. In der Regel entwickeln Unternehmen und Organisationen selbst interne Vorschriften, die weniger allgemein gehalten sind, sich allerdings vielfach an diesem Standard orientieren (vgl. Sommerville 2001 S. 127).

### 2.3. Anforderungen

#### 2.3.1. Definition

Der Begriff “Anforderungen” bezeichnet einerseits die Dienste, die ein Benutzer von einem System benötigt, andererseits auch die Einschränkungen und Eigenschaften des Systems.

Anforderungen stellen eine Beschreibung dieser Eigenschaften dar. Die Gewinnung, Dokumentation und Validierung von Anforderungen erfolgt im Zuge der in Abschnitt 2.2.3 beschriebenen Anforderungsanalyse (vgl. Pohl 2008 S. 14; Sommerville 2001 S. 107).

#### 2.3.2. Anforderungsarten

Anforderungen können in drei Gruppen gegliedert werden:

***Funktionale Anforderungen*** Diese Anforderungen spezifizieren die Dienste, die ein System zur Verfügung stellen soll. Sie können aus drei komplementären Perspektiven betrachtet werden, die sich mit Funktionen, Datenstrukturen und dem Systemverhalten beschäftigen (zu den verschiedenen Perspektiven auf Informationssysteme siehe auch Abschnitt 3). Ziel ist es, dass die Spezifikation funktionaler Anforderungen sowohl vollständig als auch konsistent ist. Dies ist in der Praxis allerdings zumeist nicht in vollem Umfang zu erreichen. Probleme, die bei der Validation oder in späteren Projektphasen erkannt werden, müssen in der Spezifikation korrigiert werden (vgl. Pohl 2008 S. 15; Sommerville 2001 S. 110 f.).

***Nichtfunktionale Anforderungen*** Diese Art von Anforderung beschreibt Eigenschaften, die die Funktionalität eines Systems nicht direkt betreffen, wie beispielsweise Reaktionszeit oder Speicherbedarf des Systems. Auch qualitative Einschränkungen des Systems werden durch diese Art von Anforderungen spezifiziert. Zumeist beziehen sich diese Eigenschaften auf das Gesamtsystem. Nichtfunktionale Anforderungen sollten anhand von Metriken objektiv überprüfbar sein, auch wenn dies in der Praxis nicht immer erreichbar ist und daher Interpretationsspielraum bietet (vgl. Sommerville 2001 S. 111 ff.).

Pohl unterteilt die nichtfunktionalen Anforderungen in unterspezifizierte funktionale Anforderungen sowie *Qualitätsanforderungen*. Diese Einteilung soll das Problem verhindern, dass nichtfunktionale Anforderungen in der Praxis oft nicht hinterfragt werden und dadurch unterspezifiziert sein können. Diese Anforderungen sollen im Zuge des Analyseprozesses durch detailliertere funktionale Anforderungen ersetzt werden. Beispielsweise kann „Das System soll sicher sein“ als nichtfunktionale Anforderung eingeteilt werden, da dies eine Eigenschaft des Systems beschreibt. Eine solche Anforderung ist allerdings nicht sinnvoll überprüfbar. Durch Verfeinerung können

## 2.3. Anforderungen

daraus verschiedene funktionale Anforderungen gewonnen werden, wie „Der Benutzer soll sich für das System authentifizieren müssen“, „Der Datenaustausch zwischen System und Computer des Benutzers soll verschlüsselt erfolgen“ (vgl. Pohl 2008 S. 16 f.).

**Problembereichsanforderungen** Diese Anforderungen ergeben sich aus dem Anwendungsbereich des Systems. Dabei kann es sich um folgende Arten von Anforderungen handeln:

- Neue funktionale Anforderungen,
- solche Anforderungen, die existierende funktionale Anforderungen einschränken oder
- nichtfunktionale Anforderungen.

Ein Beispiel für eine Problembereichsanforderung stellt eine anwendungsspezifische Berechnungsanweisung dar (vgl. Sommerville 2001 S. 115).

### 2.3.3. Ziele

Ein Ziel im Sinne des Requirements Engineering ist *die intentionale Beschreibung eines charakteristischen Merkmals des zu entwickelnden Systems bzw. des zugehörigen Entwicklungsprozesses* (Pohl 2008 S. 91).

Ziele dienen der Betrachtung der Intention von Stakeholdern. Bei tendenziell geringem Aufwand ist eine zielorientierte Vorgehensweise aus mehreren Gesichtspunkten vorteilhaft:

- Explizit formulierte Ziele dienen dem besseren Verständnis des Systems und zeigen Mehrwerte auf. Dabei können auch irrelevante Anforderungen aufgedeckt werden, die ein Systemverhalten ohne Mehrwert spezifizieren.
- Ziele können als Grundlage für die Gewinnung von Anforderungen dienen und mit diesen verknüpft sein. Durch diese Verknüpfung existiert auch eine Rechtfertigung für Anforderungen, warum sie für das Erreichen des Systemzwecks benötigt werden.
- Ziele dienen als stabile Basis für Anforderungen, die sich in der Regel nicht mit dem Lösungsansatz ändern. Dabei können Ziele auch als Basis zur Bewertung von Lösungsansätzen verwendet werden (vgl. Pohl 2008 S. 89 f.).
- Ziele unterstützen die Fokussierung auf den Problembereich und helfen dem Requirements Engineer dabei, in Problemen und nicht in Lösungen zu denken (vgl. Nuseibeh und Easterbrook 2000 S. 39).

### 2.3. Anforderungen

- Ziele können im Zuge der Validierung verwendet werden. Dabei werden Anforderungen im Kontext der Ziele, denen sie zugeordnet sind, betrachtet. Anschließend wird überprüft, ob das Ziel durch die Menge dieser Anforderungen vollständig erfüllt werden kann (vgl. Pohl 2008 S. 169).

Ziele können natürlichsprachig dokumentiert werden, wobei auch Schablonen zum Einsatz kommen können. Darüber hinaus können Ziele über Modellierungstools wie Graphen oder i\* dargestellt werden.

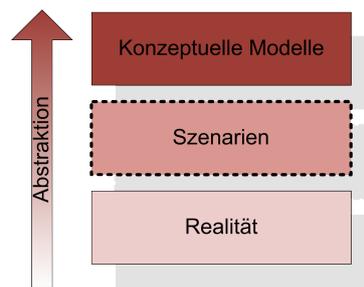
Für natürlichsprachige Ziele sollten folgende Regeln beachtet werden:

1. Ziele sollten kurz und prägnant formuliert sein.
2. Ziele sollten in aktiver Form formuliert sein.
3. Ziele sollten, wenn möglich, in einer überprüfbaren Form formuliert sein.
4. Nicht überprüfbare Ziele sollten verfeinert werden.
5. Der Mehrwert eines Zieles soll sich aus der Zieldefinition ergeben.
6. Ziele sollten, wenn möglich, kurz begründet werden.
7. Lösungsansätze sollten vermieden werden (vgl. Pohl 2008 S. 100–108)

Ein wichtiger Schritt des Requirements Engineering ist das Priorisieren von Anforderungen. Diese Priorisierung soll sich an den Bedürfnissen der Benutzer orientieren. Durch das Betrachten von höher priorisierten Anforderungen vor niedriger priorisierten können sowohl Zeitaufwand als auch Kosten in der Anforderungsanalyse verringert werden (vgl. Hofmann und Lehner 2001 S. 59). Zur Priorisierung sind Ziele als grundlegende Anforderungen gut geeignet.

Zu beachten ist in diesem Zusammenhang auch, dass die Ziele von Benutzern nicht unbedingt sofort ersichtlich sind. Ein Buchhalter kann scheinbar zum Ziel haben, Rechnungen schneller bearbeiten zu können. Dies ist jedoch eher das Ziel des Arbeitgebers und nicht des Mitarbeiters selbst. Der Mitarbeiter hat wahrscheinlich das Ziel, bei seiner Arbeit einen kompetenten Eindruck zu hinterlassen und auch bei repetitiven Routineaufgaben motiviert zu bleiben. Im Zuge des Requirements Engineerings sollten auch diese Faktoren berücksichtigt werden (vgl. Cooper und Reimann 2003 S. 11).

## 2.3. Anforderungen



**Abbildung 2.7.:** Szenarien als Middle-Level-Abstraktion. Szenarien fungieren als Bindeglied zwischen abstrakten Modellen (Ziele und Anforderungen) und der Realität, auf denen diese Modelle basieren (nach Pohl 2008 S. 119 f.).

### 2.3.4. Szenarien

*Ein Szenario ist ein konkretes Beispiel für die Erfüllung bzw. Nichterfüllung eines oder mehrerer Ziele. Es konkretisiert dadurch eines oder mehrere Ziele. Ein Szenario enthält typischerweise eine Folge von Interaktionsschritten und setzt diese in Bezug zum Systemkontext (Pohl 2008 S. 123).*

Szenarien beschreiben ein System auf einer weniger abstrakten Ebene als Ziele und Anforderungen, indem sie die Interaktionsfolge zwischen Benutzer und System beschreiben. Dadurch fungieren sie als Bindeglied zwischen den abstrakten Modellen und der abzubildenden Realität (siehe auch Abbildung 2.7).

Szenarien bieten eine intuitive Möglichkeit, Anforderung in konkreter Form zu beschreiben. Sie können im Rahmen des Requirements Engineerings dazu eingesetzt werden, beschreibend das Auffinden von Anforderungen und Zielen zu unterstützen, explorativ die Bewertung unterschiedlicher Lösungsmöglichkeiten zu erleichtern oder erklärend die Hintergründe von Interaktionen zu liefern (vgl. Pohl 2008 S. 119–123; S. 313).

Szenarien können eine Vielzahl von Kontextinformationen beinhalten. Typische Informationen sind:

**Agenten:** Personen oder Systeme, die mit dem System interagieren

**Rollen:** Benutzerklassen, dienen der Klassifikation von Agenten

**Ziele:** Ziele, die zu einem Szenario zugehörig sind und dadurch konkretisiert werden

**Vor- und Nachbedingungen:** Bedingungen, die zu Beginn des Szenarios bzw. nach dessen Beendigung erfüllt sein müssen (vgl. Pohl 2008 S. 124 f.)

### 2.3.5. Dokumentation von Szenarien

Szenarien können auf verschiedene Arten dokumentiert werden. Dabei kann sowohl eine textuelle Repräsentation verwendet werden als auch Diagramme.

**Narrative Szenarien** Diese Szenarien stellen eine einfache, natürlichsprachige Beschreibung eines Szenarios dar. Ein narratives Szenario beschreibt einen Interaktionsablauf in erzählender Form. Es umfasst und vermischt Informationen mit unterschiedlichen Abstraktionsgraden, so zum Beispiel strukturelle, erklärende und funktionale Aspekte.

Narrative Szenarien sind durch ihre erzählende, nichttechnische Repräsentation für Stakeholder einfach verständlich und unterstützen die Gewinnung von Kontextinformationen. Diese Art von Szenarien wird zu Beginn des Entwicklungsprozess verwendet und kann später durch eine strukturierte Darstellung abgelöst werden. Dieser Schritt soll jedoch nicht zu früh erfolgen, um die Vorteile narrativer Szenarien in der Gewinnung länger nutzen zu können (vgl. Pohl 2008 S. 142 ff.).

**Strukturierte textuelle Dokumentation** Eine strukturierte Darstellung von Szenarien verbessert die Verständlichkeit und verdeutlicht die einzelnen Interaktionsschritte. Diese Darstellung kann in Form einer einfachen Nummerierung vorgenommen werden oder in Tabellenspalten, wobei jede Tabellenspalte einem Akteur zugeordnet ist. Im Gegensatz zur nummerierten Darstellung müssen hierbei die Akteure nicht explizit in den einzelnen Szenarioschritten aufgeführt werden. Eine zusätzliche Verwendung einer *Schablone* erweitert diese Repräsentationsformen um eine definierte Darstellung zugehöriger Attribute wie zum Beispiel den Szenariobezeichner und -namen, beteiligte Autoren, Versionen, Akteure sowie Vor- und Nachbedingungen (vgl. Pohl 2008 S. 142–146).

Bei der Beschreibung von Szenarien in natürlicher Sprache sollten folgende Regeln beachtet werden:

1. Verwenden der Präsensform
2. Verwenden der Aktivform
3. Formulierung in der Form „Subjekt-Prädikat-Objekt-Präpositionalgruppe“
4. Vermeidung von Modalverben
5. Beschreibung von einer Interaktion pro Satz
6. Nummerierung von Szenarioschritten

## 2.4. Techniken im Requirements Engineering

7. Eine Interaktionsfolge pro Szenario
8. Verwendung des Blickwinkels eines Außenstehenden (keine zu detaillierte Beschreibung)
9. Explizites Benennen der beteiligten Akteure in den Szenarioschritten
10. Explizites Benennen des Ziels eines Szenarios
11. Fokussierung der Beschreibung auf das Szenarioziel (vgl. Pohl 2008 S. 151–155)

**Dokumentation als konzeptuelles Modell** Eine weitere Möglichkeit, Szenarien darzustellen, ist das Verwenden von Diagrammen. Dabei können verschiedene Arten von Diagrammen zum Einsatz kommen. Greift man auf Elemente der UML-Notation (Unified Modeling Language) zurück, können dabei folgende Arten von Diagrammen verwendet werden:

**Sequenzdiagramme:** Diese Art von Diagrammen kann dazu verwendet werden, um die Interaktion zwischen Akteuren strukturiert darzustellen.

**Aktivitätsdiagramme:** Aktivitätsdiagramme werden dazu verwendet, um den Kontrollfluss *zwischen* Szenarien zu dokumentieren.

**Use-Case-Diagramme:** Diese Diagramme bieten einen Überblick über die Beziehungen von Use Cases und den zugehörigen Akteuren (vgl. Pohl 2008 S. 156–165).

Eine kombinierte Darstellung von textuellen bzw. schablonenbasierten Szenarien sowie einer modellbasierten Darstellung ist ebenfalls möglich und sinnvoll (vgl. Pohl 2008 S. 165). Tabelle 2.2 stellt einen Überblick über die verschiedenen Dokumentationsarten sowie ihre Eignung in der Analyse- und Spezifikationsphase dar.

## 2.4. Techniken im Requirements Engineering

### 2.4.1. Gewinnung von Anforderungen

*Das Ziel der Aktivität „Gewinnung von Anforderungen“ ist (...) ein möglichst vollständiges inhaltliches Verständnis aller Anforderungen an das geplante System.*

*Ideen sowie konkrete Anforderungen an das geplante System stecken in den Köpfen von Stakeholdern, stehen in Dokumenten oder sind u.a. bereits in existierenden Systemen realisiert. (...) Dieses Wissen muss (...) aus den Anforderungsquellen gewonnen werden (Pohl 2008 S. 311).*

Die Aktivität der Gewinnung besteht aus drei Teilaktivitäten. Techniken, die in den jeweiligen Phasen zum Einsatz kommen können, finden sich in Abschnitt 2.4.2.

## 2.4. Techniken im Requirements Engineering

Dokumentationsart	Textuell	Schablonenbasiert	Sequenzdiagramme	Aktivitätsdiagramme	Use-Case-Diagramme
Verwendung Analyse	***	**	**	–	**
Verwendung Spezifikation	–	***	***	*	**

**Tabelle 2.2.:** Die Eignung verschiedener Szenariodokumentationsarten in der Analyse- und Spezifikationsphase (nach Pohl 2008 S. 166). Die Skala reicht von „–“ (nicht geeignet) bis hin zu „\*\*\*“ (besonders geeignet).

**Identifikation von Anforderungsquellen** In einem ersten Schritt müssen potentielle Anforderungsquellen identifiziert werden. Bereits bekannte Stakeholder werden nach möglichen Anforderungsquellen befragt. Dabei sollten verschiedene Facetten möglichst vollständig berücksichtigt werden:

**Gegenstandsfacette** Ein Softwaresystem bildet Gegenstände aus der realen Welt ab und verarbeitet diese Informationen weiter. Anforderungsquellen dieser Facette können Informationen zu diesen Gegenständen liefern.

**Nutzungsfacette** Diese Facette beinhaltet die zukünftigen Benutzer des Systems.

**IT-Systemfacette** Anforderungsquellen dieser Facette liefern Eigenschaften der IT-Umgebung, in der sich das zukünftige System befinden wird.

**Entwicklungsfacette** Diese Facette beinhaltet jene Quellen, die den Entwicklungsprozess des zukünftigen Systems umfassen.

In einem weiteren Schritt wird die Relevanz einzelner Anforderungsquellen für die weitere Analyse bestimmt (vgl. Pohl 2008 S. 40; S. 314 ff.).

**Gewinnung existierender Anforderungen** Im Zuge dieser Tätigkeit werden Anforderungen aus zuvor definierten relevanten Anforderungsquellen gewonnen. Bei natürlichen Personen können unterschiedliche Gewinnungstechniken zum Einsatz kommen wie beispielsweise Beobachtung oder Gespräche mit diesen Stakeholdern. Beobachtungen können auch bei der Gewinnung

## 2.4. Techniken im Requirements Engineering

von Anforderungen aus existierenden Systemen hilfreich sein. Die Gewinnung aus Dokumenten geschieht durch Lesen und darauf folgende Analyse (vgl. Pohl 2008 S. 318 ff.).

**Entwicklung innovativer Anforderungen** Innovative Anforderungen können nicht direkt aus bereits existierenden Quellen gewonnen werden, sondern müssen in einem kreativen Prozess entwickelt werden. In der Praxis wird diese Phase aus Zeit- und Kostengründen oft nicht umgesetzt (vgl. Pohl 2008 S. 320 f.).

Ein wesentlicher Aspekt bei der Gewinnung von Anforderungen ist der Humanfaktor. Kommunikation ist eine zentrale Komponente im Requirements Engineering. Hierbei liegt der Fokus unter anderem auf der Lösung von Konflikten und unterschiedlichen Ansichten der Stakeholder im Requirements-Prozess, da die zukünftigen Lösungen eine breite Unterstützung benötigen. Durch die Einbindung von Stakeholdern kann diesen das Gefühl vermittelt werden, direkt am Entwurf des zukünftigen Systems beteiligt zu sein. Dies ist einerseits essentiell für das Verständnis des vom Benutzer gewünschten Systems durch die Berücksichtigung der Ziele von Anwendern (siehe Abschnitt 2.3.3), andererseits auch für dessen Akzeptanz und damit für die Zufriedenheit der Stakeholder (vgl. Holtzblatt und Beyer 1998 S. 31 f.).

### 2.4.2. Gewinnungstechniken

Zur Gewinnung von Anforderungen stehen unterschiedliche Techniken zur Auswahl, die sich in Aufwand und Nutzen für einzelne Teilaktivitäten der Anforderungsgewinnung unterscheiden. Tabelle 2.3 bietet eine Übersicht über die einzelnen Gewinnungstechniken.

Technik	Aufwand	Teilaktivität		
		Identifikation	Gewinnung	Entwicklung
<i>Interview</i>	hoch/mittel	✓	✓	✓
<i>Workshop</i>	sehr hoch	✓	✓	✓
<i>Beobachtung</i>	sehr hoch/hoch	–	✓	–
<i>Schriftliche Befragung</i>	gering	✓	✓	✓
<i>Perspektivenbas. Lesen</i>	hoch/mittel	–	✓	–

**Tabelle 2.3.:** Übersicht der Gewinnungstechniken im Requirements Engineering (nach Pohl 2008 S. 324). Die Teilaktivitäten entsprechen den in Abschnitt 2.4.1 besprochenen.

**Interview** Interviews werden dazu benutzt, um Informationen von Stakeholdern zu erhalten. Sie können sowohl explorativ als auch standardisiert geführt werden (vgl. Oppenheim 1999; zit.

## 2.4. Techniken im Requirements Engineering

nach Pohl 2008 S. 325). Ein exploratives Interview ist ein offenes Gespräch, das – im Gegensatz zu standardisierten Interview – nicht in einer zuvor exakt definierten Folge an Fragen durchgeführt wird. Interviews können sowohl mit Einzelpersonen als auch in Gruppen geführt werden.

Zur Identifikation von Anforderungsquellen kann ein standardisiertes Interview mit unterschiedlichen Stakeholdern geführt werden. Für die Anforderungsgewinnung eignen sich offene Interviews, die auch als Basis für weitere Workshops verwendet werden können. Eingeschränkt nützlich sind offene Interviews für die Entwicklung innovativer Anforderungen (vgl. Pohl 2008 S. 325–335). Diese Tätigkeit kann *durch geschicktes Nachfragen und durch Konfrontieren der Interviewten mit unausgereiften Lösungsideen unterstützt werden. In Gruppeninterviews können sich innovative Anforderungen zudem aus Synergien zwischen den Teilnehmern ergeben* (Pohl 2008 S. 334).

**Workshop** Workshops werden als die mächtigste Technik der Anforderungsgewinnung bezeichnet (vgl. Leffingwell und Widrig 2000; zit. nach Pohl 2008 S. 336). Workshops unterscheiden sich dadurch von Interviews, dass eine Gruppe von Stakeholdern Anforderungen erarbeitet, anstatt dass diese von den Teilnehmern lediglich erfragt werden. Die Ergebnisse dieser Technik werden in einem Protokoll festgehalten.

Workshops eignen sich sehr gut für alle drei Teilaktivitäten der Anforderungsgewinnung. Assistenztechniken wie Brainstorming können die Ergebnisse eines Workshops positiv beeinflussen. Workshops sollten in Kleingruppen von drei bis vier Teilnehmern geführt werden, die entweder die gleichen Ausgangsziele erhalten oder aber unterschiedliche Ziele. Abschließend sollten erzielte Ergebnisse mit allen Teilgruppen diskutiert werden. Durch die aktive Beteiligung mehrerer Stakeholder ist diese Technik sehr gewinnbringend, verursacht allerdings auch einen sehr hohen Zeitaufwand (vgl. Pohl 2008 S. 336–345).

**Beobachtung** Mithilfe dieser Technik werden Anforderungen aus der Beobachtung von Benutzern oder Systemen gewonnen. Beobachtungen können direkt durchgeführt werden, indem ein Beobachter dem Benutzer bei der Durchführung von Aufgaben zusieht und dabei Fragen stellt. Bei ethnografischen Beobachtungen erlernt der Beobachter aktiv Arbeitsweisen und führt die Tätigkeiten der Stakeholder auch selbst durch.

Diese Technik ist hauptsächlich in der Gewinnung existierender Anforderungen von Vorteil. Indirekt hilfreich ist die ethnografische Beobachtung bei der Entwicklung innovativer Anforderungen, da sie dem Problembereichsverständnis des Beobachters dient (vgl. Pohl 2008 S. 346–352).

## 2.4. Techniken im Requirements Engineering

**Schriftliche Befragung** Die schriftliche Befragung wird von Stakeholdern selbständig durchgeführt. Ausgehend von initialen Zielen beantworten Stakeholder Fragen, die sowohl offen als auch geschlossen als Auswahl zwischen verschiedenen Alternativen gestellt werden können. Die Schriftliche Befragung eignet sich sowohl für die Identifikation von Anforderungsquellen als auch zur Gewinnung existierender Anforderungen (vgl. Pohl 2008 S. 352–356).

**Perspektivenbasiertes Lesen** Das perspektivenbasierte Lesen zielt darauf ab, Dokumente aus der Perspektive eines zuvor festgelegten Stakeholders zu lesen und für diesen irrelevante Informationen zu ignorieren. Diese Technik eignet sich zur Gewinnung existierender Anforderungen aus Dokumenten (vgl. Pohl 2008 S. 356–360).

### 2.4.3. Validierungstechniken

Die Validierung von Anforderungen ist wesentlich für den Softwareentwicklungsprozess. Abschnitt 2.1.1 führt beispielhaft auf, welche Probleme durch fehlerhafte und unvalidierte Anforderungen entstehen können. Durch die Validierung können Fehler frühzeitig im Projektverlauf erkannt werden (vgl. Pohl 2008 S. 419 ff.).

Die Validierung kann auf mehreren Ebenen erfolgen. Folgende Punkte sollen als Minimalziel im Validierungsprozess betrachtet werden:

1. Sind die Anforderungen nachvollziehbar?
2. Sind die Anforderungen konsistent?
3. Sind die Anforderungen korrekt?
4. Sind die Anforderungen vollständig?
5. Sind die Anforderungen im richtigen Dokumentationsformat?
6. Sind die Anforderungen verständlich?
7. Sind die Anforderungen abgestimmt?
8. Wurden alle relevanten Stakeholder in die Anforderungsgewinnung einbezogen?

Darüber hinaus lassen sich noch zusätzliche Qualitätsziele definieren, die verschiedene Aspekte der Anforderungen im Detail betrachten (vgl. Pohl 2008 S. 432 f.).

Wesentlichen Nutzen in der Validierung hat die Verwendung von Zielen und Szenarien. Da Ziele als Grundlage für die weitere Anforderungsgewinnung dienen, müssen sie bereits frühzeitig

## 2.4. Techniken im Requirements Engineering

validiert werden, um Folgefehler zu vermeiden. Zugehörige Szenarien nehmen dabei einen wichtigen Stellenwert ein, weil sie abstrakte Ziele mit konkreten Beispielen verknüpfen. Sie erlauben die Validierung aus unterschiedlichen Blickwinkeln auf Basis von Interaktionsschritten (vgl. Pohl 2008 S. 434 f.).

Zur Validierung können unterschiedliche Techniken eingesetzt werden.

**Inspektion** Bei der Inspektion sollen Fehler systematisch aufgedeckt werden. Dabei wird zu-  
meist auf ein striktes Prozessschema zurückgegriffen, das die Phasen Planung, Übersicht, Fehler-  
suche und Fehlersammlung beinhaltet. Dieser Ansatz eignet sich für eine detaillierte Überprüfung  
von Anforderungen (vgl. Pohl 2008 S. 444–450).

**Reviews** Reviews sind eine vereinfachte Form von Inspektionen, die weniger formal durch-  
geführt werden. Sie konzentrieren sich auf eine quantitativ größere Anzahl von Anforderungen,  
haben allerdings eine geringere qualitative Detailliertheit. Reviews können von Prüfern entweder  
unabhängig durchgeführt werden oder innerhalb einer Gruppe. Reviews haben den Vorteil, in  
kürzerer Zeit einen umfassenden Überblick zu bieten, erreichen aber nicht die Genauigkeit von  
Inspektionen (Pohl 2008 S. 450–452).

**Walkthroughs** Bei einem Walkthrough führt ein Moderator die Teilnehmer durch die zu  
prüfenden Artefakte. Dabei werden die Artefakte erläutert und auf kritische Stellen hingewiesen.  
Diese Technik ist vor allem für frühe Anforderungsstadien von Vorteil und kann neue Ideen und  
Einsichten für Anforderungen liefern (vgl. Pohl 2008 S. 450–454).

Neben diesen Validierungsarten kann auch das in Abschnitt 2.4.2 beschriebene perspektiven-  
basierte Lesen eingesetzt werden. Desweiteren können Prototypen eingesetzt werden, die es Sta-  
keholdern als Weiterentwicklung von Szenarien erlauben, Aktivitäten in einer Systemumgebung  
durchzuführen (vgl. Pohl 2008 S. 456–464).

## 3. Modellierung von Informationssystemen

### 3.1. Modellierung von Geschäftsprozessen

#### 3.1.1. Motivation

Das Ziel des Einsatzes von IT-Systemen in Unternehmen ist es, die Geschäftsprozesse des Unternehmens optimal zu unterstützen. Fehler in der Anforderungsdefinition können in Systemen resultieren, die den Unternehmensprozess nicht oder nicht hinreichend unterstützen und nur durch Workarounds sinnvoll einsetzbar sind. Deshalb ist gerade bei der Planung und Spezifikation das Verständnis des Kontextes und des Unternehmens, in dem eine Software eingesetzt wird, unerlässlich.

Dieses Verständnis kann durch ein Modell des Geschäfts erlangt werden. Ein Modell ist eine vereinfachte, abstrakte Sicht auf die Realität. Durch diese Abstraktion lassen sich bestimmte, für den jeweiligen Kontext irrelevante Details ausblenden und stattdessen zentrale Punkte in den Mittelpunkt rücken. Modellierung erlaubt es Fachleuten auf dem jeweiligen Gebiet über eine Software zu diskutieren und sich an deren Entwicklung zu beteiligen, ohne selbst Informatiker zu sein (vgl. Mielke 2002 S. 1 f.). Der Einsatz von Modellen ist bereits in frühen Phasen des Requirements Engineerings sinnvoll, da gerade in diesen Phasen das Verständnis des „Warums“ für Ziele und Anforderungen von großer Bedeutung ist. Modelle tragen dazu bei, zu verstehen, was der Benutzer eines Systems eigentlich möchte und benötigt (vgl. Yu 1997 S. 1 f.).

Modelle können informell oder semiformell (z.B. durch den Einsatz von Schablonen) in Textform beschrieben werden. Diese Darstellung hat allerdings den Nachteil, dass durch Mehrdeutigkeiten oder ungenaue Formulierungen ein gewisser Freiraum bei der Implementierung des Zielsystems herrscht. Dies kann dazu führen, dass das Zielsystem nicht oder nicht in vollem Umfang den Bedürfnissen und Anforderungen des Unternehmens entspricht. Formelle Modelle können diesen Spielraum verringern. Mittels eindeutig definierter Transformationsregeln kann aus Prozessmodellen auch automatisch Software generiert werden. Die auf dem ARIS-Framework (siehe Abschnitt 3.1.3) basierende ARIS-Plattform ist ein Beispiel für ein System, das auf Modellen basiert.

Für die Modellierung von Geschäftsprozessen ist nicht nur ein Ablaufmodell notwendig; auch andere Aspekte wie die Organisation eines Unternehmens oder die Informationsstruktur müssen

### 3.1. Modellierung von Geschäftsprozessen

modelliert werden. Dies erlaubt die Abdeckung verschiedener Aspekte des Geschäftsprozesses (vgl. Mielke 2002 S. 2 f.; Sommerville 2005 S. 17).

#### 3.1.2. Geschäftsprozesse

##### Definition

Der Begriff *Geschäftsprozess* hat in der Literatur unterschiedliche Definitionen. Ein Geschäftsprozess kann die Menge an Aktivitäten sein, die aus unterschiedlichen Inputs einen für den Kunden nutzbringenden Output erzeugt. Er kann auch eine durch informationstechnologische Anwendungen unterstützte Abfolge von Aufgaben darstellen (vgl. Gadatsch 2008 S. 45). Ebenso gibt es den Ansatz, Geschäftsprozesse als *Transaktion oder eine planmäßige Folge von Transaktionen zwischen betrieblichen Objekten im Rahmen der Leistungserzeugung* (Mielke 2002 S. 10) zu definieren.

Eine weitere Definition eines Geschäftsprozesses ist die *modellhafte Beschreibung der in einem Unternehmen durchzuführenden Funktionen in ihrer inhaltlichen und zeitlichen Abhängigkeit* (vgl. Gadatsch 2008 S. 45). Funktionen sind Aufgaben und Tätigkeiten, die über Ereignisse verknüpft sind (vgl. Scheer 1990; Scheer und Jost 1996; zit. nach Gadatsch 2008 S. 45).

Im Rahmen dieser Arbeit werden die Begriffe Prozess und Geschäftsprozess synonym verwendet, wenn nicht anders angegeben.

Ein *Workflow* ist ein automatisierter Geschäftsprozess (vgl. Gadatsch 2008 S. 11). Hierbei werden *Dokumente, Informationen oder Aufgaben von einem Teilnehmer an einen anderen zur Ausführung entsprechend einer Menge von prozeduralen Regeln übergeben* (Gadatsch 2008 S. 52). Eine *Workflow-Instanz* ist eine konkrete Ausführung eines Workflows (vgl. Gadatsch 2008 S. 53). Im Gegensatz zu einem Geschäftsprozess, der auf konzeptioneller Ebene die Analyse und Gestaltung von Abläufen zum Ziel hat, beschreibt ein Workflow auf einer operativen Ebene die technische Ausführung dieser Abläufe. Die Detaillierung reicht bei Geschäftsprozessen bis zu dem Grad, an dem ein Mitarbeiter Arbeitsschritte in einem Zug und an einem Arbeitsplatz ausführen kann. Ein Workflow kann diese Arbeitsschritte noch weiter verfeinern (vgl. Gadatsch 2008 S. 59).

##### Geschäftsprozessmanagement

*Geschäftsprozessmanagement (Business Process Management, BPM)* ist ein *Konzept zur modellbasierten Gestaltung, Koordination und Ausführung von Geschäftsprozessen* (Scheer et al. 1995 S. 7). Das Geschäftsprozessmanagement hat zum Ziel, Prozesse einerseits an die Unternehmensstrategie anzupassen, andererseits dient es auch der Organisation von Prozessen sowie

### 3.1. Modellierung von Geschäftsprozessen

deren technischer Unterstützung und Umsetzung durch IT-Systeme (vgl. Gadatsch 2008 S. 1 f.). Der Lebenszyklus von BPM besteht aus vier Phasen (vgl. Davis und Brabänder 2007 S. 7).

**Strategiephase** In dieser Phase werden kritische Erfolgsfaktoren des Unternehmens sowie Kernkompetenzen und andere, für die Unternehmensstrategie wesentlichen Punkte analysiert und ausgewertet. Basierend auf dieser Auswertung wird die Unternehmensstrategie festgelegt. (vgl. Gadatsch 2008 S. 1 f.; Scheer et al. 1995 S. 9). Nach dem Prinzip „Structure follows strategy“ sollten Geschäftsprozesse an der Unternehmensstrategie ausgerichtet werden, um sicherzustellen, dass die gewählte Strategie auch optimal umgesetzt werden kann. Die Unternehmensstrategie beeinflusst deshalb die Geschäftsprozesse in einem Unternehmen wesentlich; jede Änderung in der Unternehmensstrategie führt auch zu Änderungen in Geschäftsprozessen (vgl. Davis und Brabänder 2007 S. 7 f.).

**Modellierungsphase** In dieser Phase werden Geschäftsprozesse an die Unternehmensstrategie und die Erfordernisse des Zielmarktes angepasst. Dazu werden Geschäftsprozesse analysiert, auf Schwächen untersucht und optimiert. Hierbei können auch weitere Facetten des Unternehmens wie beispielsweise die Systemlandschaft analysiert und optimiert werden (vgl. Davis und Brabänder 2007 S. 9).

Für eine Neumodellierung von Prozessen müssen zuerst Prozesskandidaten anhand der Unternehmensstrategie identifiziert werden. Als nächster Schritt werden die Prozesse gestaltet und modelliert. Dies kann auf Basis von Referenzmodellen geschehen, die als Basismodell für Prozesse entweder komplett übernommen oder aber adaptiert werden. Zu Prozessen werden Messgrößen definiert, die in der Controllingphase zur Performanceüberprüfung verwendet werden (vgl. Gadatsch 2008 S. 2 f.; Scheer et al. 1995 S. 8 f.).

**Implementierungsphase** Nach der Modellierung der Geschäftsprozesse müssen diese implementiert werden. Die Implementierungsphase konzentriert sich auf die Abbildung von Geschäftsprozessen in einem Softwaresystem. Dies kann über ERP-Systeme wie beispielsweise SAP geschehen oder über eine Service-oriented Architecture (SOA), die einzelne und eigenständige Dienste anbietet. Eine dritte Möglichkeit ist die Entwicklung von Individualsoftware, die die zu unterstützenden Geschäftsprozesse abbildet (vgl. Davis und Brabänder 2007 S. 9 ff.).

**Controllingphase** In dieser Phase werden Prozesse anhand der in der Modellierungsphase definierten Kennzahlen bewertet. Diese Kennzahlen werden in der Regel vom eingesetzten IT-System automatisch ermittelt. Sie dienen dazu, Schwachstellen bei der Ausführung der Prozesse

### 3.1. Modellierung von Geschäftsprozessen

zu erkennen. Basierend auf dieser Analyse können Prozesse optimiert werden (vgl. Davis und Brabänder 2007 S. 11 f.).

#### Workflow-Management

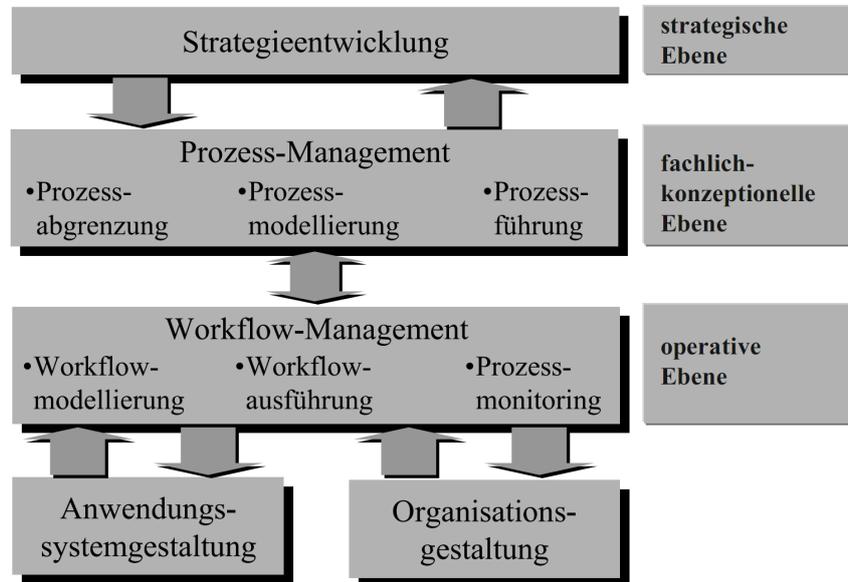


Abbildung 3.1.: Integriertes Prozess- und Workflow-Management (aus Gadatsch 2008 S. 2).

Das Workflow-Management beschäftigt sich mit der Workflow-Modellierung. Diese Tätigkeit basiert auf der Modellierung der Geschäftsprozesse. Der Geschäftsprozess wird in diesem Schritt um detaillierte Spezifikationen erweitert, die eine (teil-)automatisierte Ausführung des Prozesses erlauben. Hierbei fließen detaillierte Betrachtungen von Datenstrukturen und Rollen aus der Organisation in das Modell ein.

In der Workflow-Ausführung werden Prozessobjekte durchlaufen. Die so gewonnenen IST-Daten können in der Controllingphase mit SOLL-Daten verglichen werden. Auch können basierend auf diesen Daten Simulationen und Abweichungsanalysen durchgeführt werden (vgl. Gadatsch 2008 S. 3; Scheer et al. 1995 S. 10 f.).

Abbildung 3.1 zeigt den Zusammenhang zwischen Unternehmensstrategie, BPM und Workflow-Management.

#### 3.1.3. ARIS

ARIS („Architektur integrierter Informationssysteme“) ist ein Framework, um Geschäftsprozesse zu gestalten und zu optimieren und diese in weiterer Folge durch IT-Systeme zu unterstützen

## 3.2. Perspektiven und Notationen

(vgl. Scheer 1996 S. 2). Zu diesem Zweck definiert ARIS mehrere Modelle, die zur Beschreibung, Strukturierung und Analyse von Prozessen eingesetzt werden können. Diese Modelle repräsentieren unterschiedliche Perspektiven eines Geschäftsprozesses.

Ziel der Verwendung mehrere Perspektiven ist die Reduktion der Komplexität der Geschäftsprozessmodellierung. Durch die verschiedenen Blickwinkel konzentrieren sich die Modelle auf Teilaspekte eines Geschäftsprozesses, anstatt den gesamten Prozess mitsamt Informationen der Organisation und Datenmodelle in allen Details darstellen zu müssen. Das in Abbildung 3.2 dargestellte ARIS-Haus basiert auf dieser Aufteilung (vgl. Davis und Brabänder 2007 S. 27–31) Die im ARIS-Framework verwendeten Perspektiven sind:

**Organisationsperspektive** Statische Modelle der Organisationsstruktur.

**Datenperspektive** Statische Modelle von Informationen innerhalb der Organisation wie zum Beispiel Datenmodelle.

**Funktionsperspektive** Statische Modelle von Aufgaben, die in Prozessen ausgeführt werden.

**Prozessperspektive** Dynamisches Prozessmodell, das das Verhalten von Prozessen und ihre Beziehung zur Organisation, Daten und Funktionen zeigt. Dieser auch als Steuerungsperspektive bekannte Blickwinkel betrachtet die Beziehungen zwischen den anderen Perspektiven.

Neuere Versionen des ARIS-Frameworks haben darüber hinaus noch eine Produktperspektive, die ein statisches Modell der Struktur von Produkten beinhaltet (vgl. Scheer 1996 S. 10; Davis und Brabänder 2007 S. 32 f.).

Das ARIS-Konzept basiert auf einer hierarchischen Architektur, die es erlaubt, Prozessmodelle in verschiedenen Ebenen zu betrachten. Von einer abstrakten Makroebene wird der Detaillierungsgrad von Ebene zu Ebene vergrößert und erlaubt so genauere Einblicke in Geschäftsprozesse. Eine typische Prozessarchitektur besteht aus vier bis sechs solcher Ebenen (vgl. Davis und Brabänder 2007 S. 50 f.).

Für ARIS existiert ein kommerzielles Softwaresystem, das die Anwendung des Frameworks unterstützt.

## 3.2. Perspektiven und Notationen

### 3.2.1. Organisationsperspektive

Die Organisationsperspektive beschreibt die Strukturierung einer Organisation. Dabei wird die Hierarchie von Abteilungen, Gruppen oder Rollen dargestellt (vgl. Davis und Brabänder 2007

### 3.2. Perspektiven und Notationen

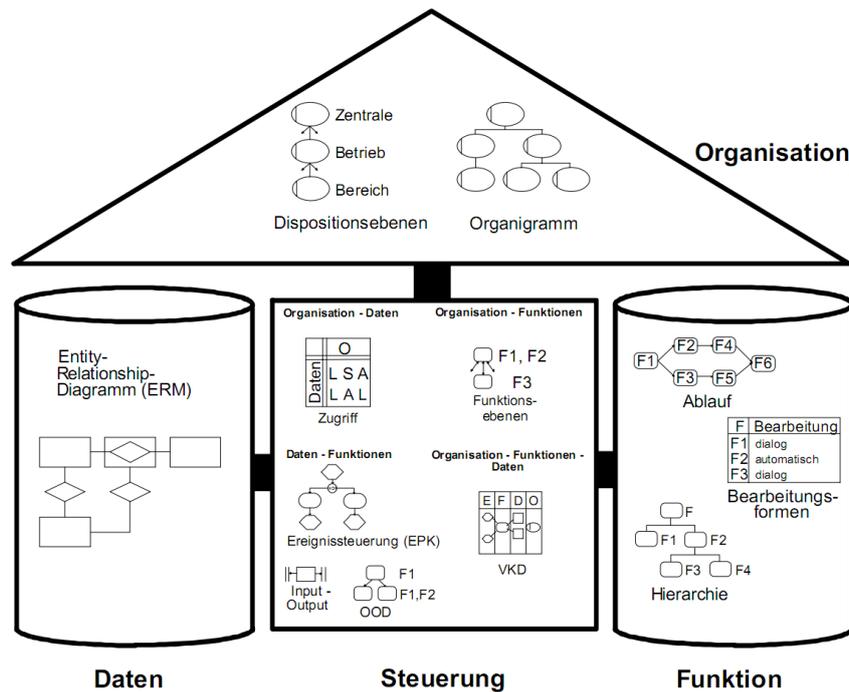


Abbildung 3.2.: Aris-Haus des Business Engineerings (aus Scheer 1996 S. 10).

S. 56 f.). Eine Darstellungsmöglichkeit für die Organisation ist das Organigramm, das die Gliederung in organisatorische Einheiten als Baum darstellt (vgl. Balzert 1996 S. 117).

#### 3.2.2. Datenperspektive

Daten in einem IT-System unterliegen in der Regel einem bestimmten Datenmodell. Dieses Datenmodell dient dazu, den für eine Aufgabe relevanten Teil der realen Welt im System abzubilden. Hierzu gehört sowohl die Bedeutung der Daten als auch ihre Struktur und Beziehungen untereinander (vgl. Mylopoulos 1998 S. 1 f.). Die Datenperspektive ist die statische Sichtweise auf dieses Datenmodell (vgl. Pohl 2008 S. 184). Einer der bekanntesten Ansätze zur Datenmodellierung ist das *Entity-Relationship Modell* von Chen. Dieser Ansatz betrachtet Entitäten und ihre Beziehung untereinander sowie die Struktur und Repräsentation dieser Informationen als Daten (vgl. Chen 1976 S. 10). Das ER-Modell wird im ARIS-Framework zur Datenmodellierung eingesetzt (vgl. Davis und Brabänder 2007 S. 57).

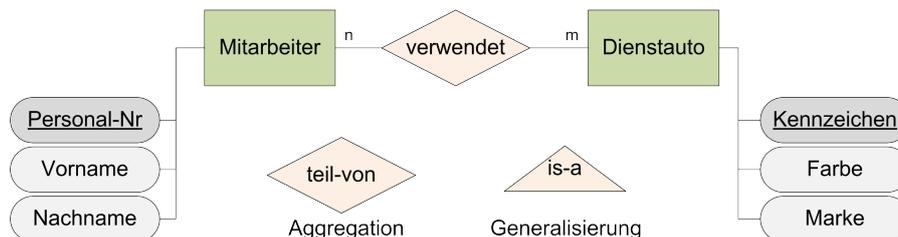
*Ausgangspunkt des ER-Modelles ist der Begriff der Entität. Eine Entität (entity) ist ein individuelles und identifizierbares Exemplar von Dingen, Personen oder Begriffen der realen oder der Vorstellungswelt und wird durch Eigenschaften beschrieben. Für Entität wird synonym oft der Begriff Objekt verwendet* (Balzert 1996 S. 138). Ein Beispiel für eine Entität ist der Mitar-

### 3.2. Perspektiven und Notationen

beiter „Hans Müller“ mit der eindeutigen Personalnummer 21; ein anderes ein schwarzes Auto der Marke Skoda mit dem eindeutigen Kennzeichen 313. Die Menge gleichartiger Entitäten (z.B. „Mitarbeiter“, „Dienstfahrzeug“) ist die Entitätsmenge oder der Entitätstyp (*entity set*) (vgl. Chen 1976 S. 11).

Eine Beziehung (relation) ist die Assoziation zwischen zwei Entitäten (vgl. Chen 1976 S. 10). Beispielsweise ist „wird-verwendet-von“ die Beziehung zwischen dem Mitarbeiter Müller und dem schwarzen Dienstauto. Analog zu Entitäten existiert auch für Beziehungen eine Beziehungsmenge oder Beziehungstyp (relationship set). *Ein Beziehungstyp klassifiziert (..) eine Menge gleichartiger Beziehungen zwischen Entitätstypen* (Pohl 2008 S. 188). Die Kardinalität einer Beziehung definiert die minimale und maximale Anzahl an Beziehungsinstanzen für eine Entität (vgl. Balzert 1996 S. 140; Pohl 2008 S. 188).

Ein Attribut (attribute) ist eine Eigenschaft für alle Entitäten einer Entitätsmenge. Ein Attribut besitzt eine bestimmte Bedeutung und einen bestimmten Wertebereich. Der Entitätstyp „Mitarbeiter“ hat beispielsweise die Eigenschaften „Personalnummer“, „Vorname“ und „Nachname“. Attribute (oder Gruppen von Attributen), die innerhalb aller Entitäten eines Entitätstyps eindeutig sind und diese somit identifizieren können, werden als Primärschlüssel (primary key) bezeichnet (vgl. Chen 1976 S. 12 ff.; Balzert 1996 S. 139). Primärschlüssel für Mitarbeiter ist in diesem Beispiel die Personalnummer, für Dienstfahrzeuge das Kennzeichen.



**Abbildung 3.3.:** Notation des ER-Diagramms. Rechtecke kennzeichnen Entitäten. Attribute werden durch Ovale dargestellt, Primärschlüssel sind unterstrichen. Das Symbol für Beziehung ist eine Raute, die Kardinalität wird an den der Linie gekennzeichnet. Aggregation und Generalisierung stellen Erweiterungen der Notation dar.

Chen definiert neben dem Modell auch eine grafische Notation, das so genannte ER-Diagramm (vgl. Chen 1976 S. 19). Abbildung 3.3 bietet eine Aufstellung der darin vorkommenden Elemente anhand des obigen Beispiels.

Für das Designen einer Datenbank unter Zuhilfenahme von ERM schlägt Chen folgende Vorgehensweise vor:

1. Identifizieren von Entitätstypen und deren Beziehungstypen.
2. Identifizieren semantischer Informationen in den Beziehungstypen, z.B. Kardinalität.

### 3.2. Perspektiven und Notationen

3. Definition der Attribute und Identifikation von Primärschlüsseln (vgl. Chen 1976 S. 20 f.).

ERM ist im Vergleich zu anderen Methoden der Datenmodellierung sowohl für das Design eines Datenmodells als auch für dessen Analyse sehr gut geeignet (vgl. Mylopoulos 1998 S. 25).

Eine Erweiterung des ERM ist die semantische Datenmodellierung, die ERM um die Elemente Aggregation und Generalisierung erweitert. Die Aggregation stellt im Gegensatz zu den gleichberechtigten Beziehungen des ERM eine Über-/Unterordnungsbeziehung dar („ist Teil von“). Die Generalisierung entspricht dem Konzept der Vererbung („ist ein“). Subtypen erben von ihren Supertypen alle Attribute. Aggregationen können in der ER-Notation durch Beziehungen dargestellt werden, für Generalisierungen werden oft Dreiecke verwendet (vgl. Balzert 1996 S. 147 ff.).

Der klassische ER-Ansatz wird im Requirements Engineering zunehmend von objektorientierten Modellierungsmethoden verdrängt, die auf der Verwendung von Klassen und Objekthierarchien basieren (vgl. Nuseibeh und Easterbrook 2000 S. 41 f.).

#### 3.2.3. Funktionsperspektive

*Eine Funktion beschreibt eine Tätigkeit oder eine klar umrissene Aufgabe innerhalb eines größeren Zusammenhangs. In der Software-Technik ermittelt eine Funktion aus Eingabedaten Ausgabedaten oder bewirkt eine Veränderung des Inhalts oder der Struktur von Informationen (Balzert 1996 S. 116).*

Durch eine hierarchische Anordnung von Funktionen in Form eines Baumes kann ein Funktionsbaum erstellt werden. Eine übergeordnete Funktion nennt sich Vaterfunktion, eine untergeordnete Kindfunktion. Eine hierarchische Ordnung kann mehrere Bedeutungen haben:

- Die übergeordnete Funktion besteht aus den untergeordneten.
- Die übergeordnete Funktion ruft die untergeordneten auf.

Vaterfunktionen sollten in Funktionsbäumen nur solchen Funktionen als Kinder haben, die eng zusammenhängende Tätigkeiten beschreiben. Darüber hinaus sollen Funktionen, die auf einer Hierarchieebene liegen, den gleichen Abstraktionsgrad aufweisen.

Funktionsbäume sind ein bewährtes System zur Gliederung von Funktionen und können auch für die Dialoggestaltung Hinweise liefern (vgl. Balzert 1996 S. 116 f.)

#### 3.2.4. Prozessperspektive

Die Prozessperspektive stellt die Geschäftsprozesse eines Unternehmens dar. Je nach Größe des Unternehmens kann diese Betrachtung in mehreren Abstraktionsgraden durchgeführt werden,

### 3.2. Perspektiven und Notationen

die sich von einer Makrosicht bis hin zu elementaren Prozessschritten erstrecken können (vgl. Gadatsch 2008 S. 48). Abbildung 3.4 veranschaulicht dieses Prinzip.

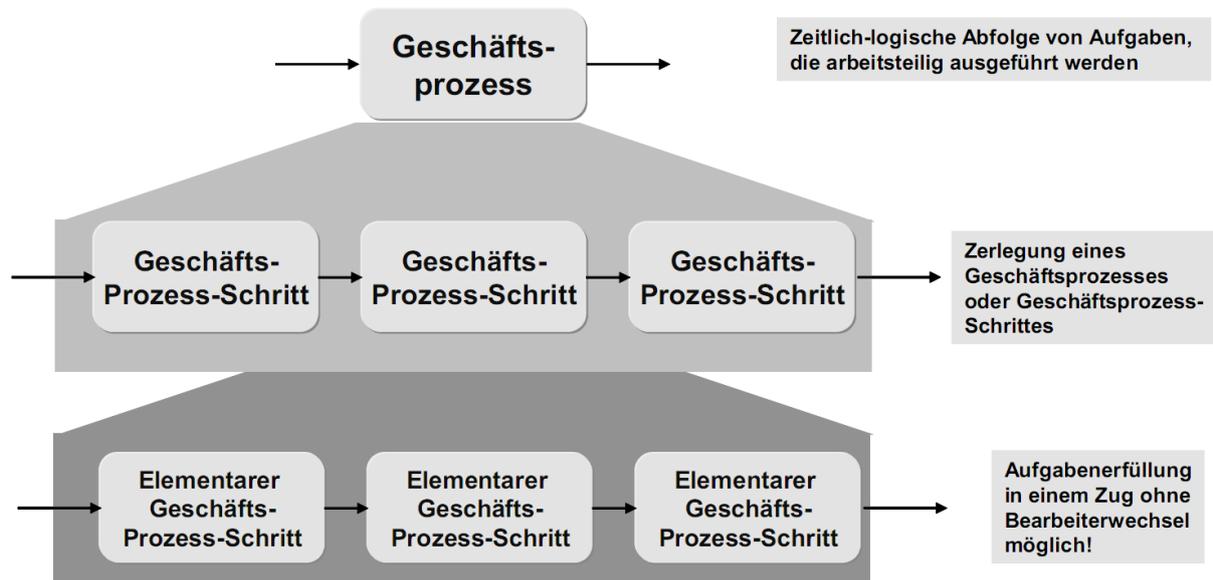


Abbildung 3.4.: Zerlegung eines Prozesses auf unterschiedlichen Ebenen (aus Gadatsch 2008 S. 48).

#### Modellierungswerkzeuge im ARIS-Framework

Im ARIS-Framework werden zur Modellierung von Prozessen zwei Konzepte verwendet: Die Wertkette und die Ereignisgesteuerte Prozesskette.

Die *Wertkette (Value Chain)* geht zurück auf Porter. Eine Wertkette zerlegt die Aktivitäten eines Unternehmens in strategisch relevante Teile. Diese Aktivitäten erzeugen für den Kunden einen bestimmten Wert, beispielsweise ein Produkt oder eine Dienstleistung (vgl. Porter 1985 S. 33–38). Die Aktivitäten stehen für jeweils eigene Prozesse, die untereinander sequentiell verknüpft sind. Die Value Chain wird hauptsächlich für die Darstellung von Geschäftsprozessen auf einer hohen Abstraktionsebene verwendet und zeigt die Schnittstellen zwischen grundlegenden Geschäftsprozessen. Durch weitere Unterteilungen kann auf diesem Weg auch eine Hierarchie von Geschäftsprozessen erzeugt werden (vgl. Davis und Brabänder 2007 S. 53 f.).

Die *Ereignisgesteuerte Prozesskette EPK (Event-driven Process Chains EPC)* ist das Hauptmodell von ARIS, um Prozesse darzustellen. Dieses dynamische Modell kombiniert Teile der statischen Modelle anderer Perspektiven (wie Organisation, Daten) mit Funktionen und stellt sie als Sequenz von Aktivitäten dar. Die zwei Hauptelemente dieses Modells sind das Ereignis und die Funktion. Ein Ereignis ist das Eintreten eines definierten Zustandes. Ereignis-

### 3.2. Perspektiven und Notationen

nisse können Funktionen auslösen, die wiederum bestimmte Aktivitäten darstellen. In einem EPC-Modell wechseln Ereignisse und Funktionen ab. Durch Regeln (AND, OR, XOR) können Verzweigungen im Ablauf dargestellt werden. EPC bietet die Möglichkeit, Funktionen mit Modellen aus anderen Perspektiven zu verknüpfen, um so zum Beispiel Daten, die in einer Funktion benötigt werden, in das Modell einarbeiten zu können (vgl. Keller et al. 1992 S. 7–12; Davis und Brabänder 2007 S. 105–108).

#### **BPMN**

Die *Business Process Modeling Notation* (BPMN) ist eine standardisierte Notation zur Beschreibung von Geschäftsprozessen. Die Version 1.0 wurde 2004 veröffentlicht, die Version 1.2 im Jänner 2009. Eine Version 2.0 befindet sich aktuell im Beta-Status. Der Standard wurde von der Business Process Management Initiative (BPMI) entwickelt. Mittlerweile wird der Standard von der Object Management Group (OMG) weiterentwickelt, die auch für den UML-Standard (Unified Modeling Language) verantwortlich ist (vgl. White 2004 S. 1, 11; OMG 2009a S. 1; OMG 2009b S. 1). Im Rahmen dieser Arbeit wurde die aktuelle Version 1.2 des Standards verwendet.

Ein Prozess im Kontext von BPMN ist eine Aktivität, die innerhalb einer Organisation oder über Organisationsgrenzen hinweg durchgeführt wird. Graphisch wird dieser Prozess über sogenannte Flow Objects und Verbindungen untereinander dargestellt. Der Abstraktionsgrad für einen Prozess kann dabei je nach Aufgabengebiet frei gewählt werden. Die graphische Repräsentation von Geschäftsprozessen wird mittels des Business Process Diagrams (BPD) durchgeführt. Darüber hinaus definiert der Standard ein Mapping zwischen BPD und BPEL4WS, einer Ausführungssprache für automatisierte BPM-Systeme. Die graphische Repräsentation basiert auf traditionellen Flowchart-Techniken. Die Modellierung von Daten-, Funktions- und Organisationsmodellen sowie der Geschäftsstrategie ist kein Teil von BPMN (vgl. OMG 2009a S. 11 f., 32).

BPMN unterteilt die Elemente der Notation in „Core Elements“, die die grundlegenden Notationselemente für BPD beschreibt. Diese Elemente können durch das „Extended Element Set“ erweitert werden, die auf der Einteilung der Core-Elemente basiert, diese jedoch mit zusätzlichen semantischen Informationen erweitert. Die folgende Beschreibung der Elemente von BPD basiert auf der Unterteilung der Core-Elemente (vgl. OMG 2009a S. 17). Erweiterte Elemente sind beschrieben, soweit sie in dieser Arbeit verwendet werden. Eine vollständige Auflistung der Elemente findet sich in (OMG 2009a S. 18–29).

**Flow Objects** Flow Objects sind die elementaren Bestandteile, um das Verhalten eines Geschäftsprozesses zu beschreiben. Abbildung 3.5 stellt diese Elemente dar. Die Flow Objects

### 3.2. Perspektiven und Notationen

werden in drei Kategorien unterteilt:

**Events** Ein Event ist etwas, das im Verlaufe eines Prozesses „passiert“. Events werden in drei Kategorien eingeteilt: Start Events stellen den Beginn eines Prozesses dar. Intermediate Events passieren im Verlauf eines Prozesses und beeinflussen diesen, stellen aber weder Start noch Ende dar. Dieses Ende des Prozesses wird durch ein End Event gekennzeichnet. Durch das „Extended Element Set“ können diese Events noch weiter präzisiert werden. BPD schlägt zehn solcher Präzisierungen vor. Im Zuge dieser Arbeit wird das Message Event verwendet, das ein durch eine Nachricht verursachtes Event darstellt. Hierbei wird zwischen gefangenen („Catching“) und gesendeten („Throwing“) Nachrichten unterschieden.

**Activities** Activities stellen die elementaren Aktivitäten eines Geschäftsprozesses dar. Zusätzlich existiert eine Notation für Subprozesse, die eine Sammlung von Aktivitäten darstellt.

**Gateways** Ein Gateway definiert das Verzweigen und Zusammenführen von Sequenzen. Ein Gateway kann durch Symbole noch weiter präzisiert werden. In dieser Arbeit wird das Parallel Gateway verwendet, das für das Verzweigen und Zusammenführen paralleler Sequenzen zuständig ist.

(vgl. OMG 2009a S. 17–25)

**Connecting Objects** Connecting Objects stellen Verbindungen zwischen Flow Objects untereinander oder zu Informationen her (Abbildung 3.6). Es existieren drei Typen von Connecting Objects:

**Sequence Flow** Ein Sequence Flow beschreibt die Reihenfolge der Aktivitäten in einem Prozess. Jeder Sequence Flow hat genau einen Ursprung und ein Ziel.

**Message Flow** Ein Message Flow beschreibt den Nachrichtenfluss zwischen Entitäten. Da in BPMN diese Entitäten durch Pools (s.u.) dargestellt werden, muss diese Verbindung zwischen zwei Pools erfolgen. Nachrichten können zwischen Pools ausgetauscht werden, ohne sich auf konkrete Objekte zu beziehen (insbesondere bei „Black Boxes“, s.u.). Sie können auch zwischen konkreten Objekten ausgetauscht werden. Ebenfalls möglich ist eine Kombination aus beiden.

**Association** Eine Association verbindet Informationsartefakte mit Flow Objects. Die Assoziation kann gerichtet oder ungerichtet sein (vgl. OMG 2009a S. 17–19, 97–103; Owen und Raj 2003 S. 17 f.).

### 3.2. Perspektiven und Notationen

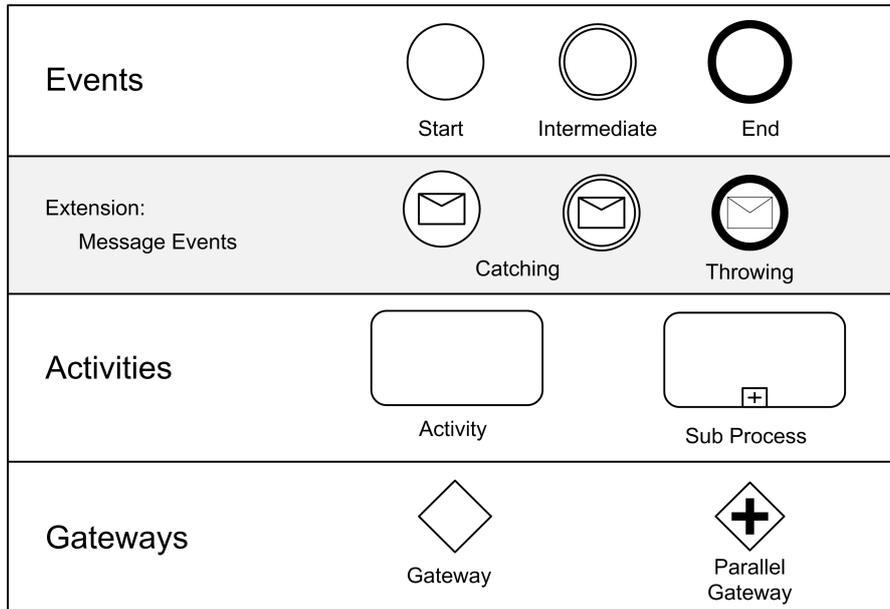


Abbildung 3.5.: Flow Objects in BPMN.

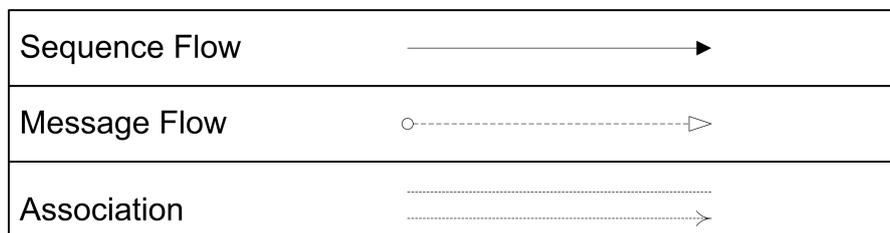


Abbildung 3.6.: Connecting Objects in BPMN.

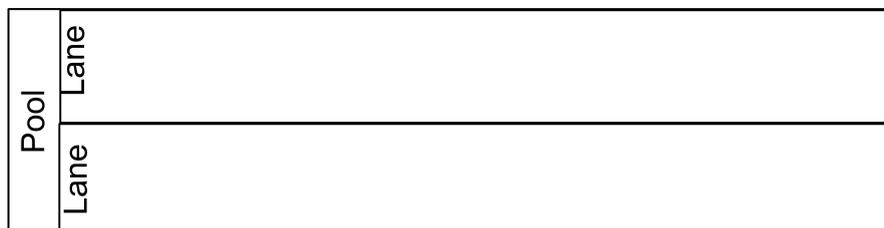


Abbildung 3.7.: Swimlanes in BPMN.

### 3.2. Perspektiven und Notationen

**Swimlanes** Swimlanes (Abbildung 3.7) dienen zur Gruppierung der Modellierungselemente in BPMN. Dazu existieren zwei Elemente:

**Pools** Ein Pool stellt einen Teilnehmer (Entität) in einem Prozess dar. Dieser Teilnehmer kann entweder eine konkrete (z.B. Firma XYZ) oder eine abstrakte (z.B. Kunde) Entität repräsentieren. Ein Pool dient als grafische Methode zur Gruppierung von Elementen („White Box“). Er kann aber auch eine „Black Box“ sein, die keine Elemente aufnimmt. Dies kann zum Beispiel bei einer externen Entität wie einem Kunden der Fall sein, dessen interne Prozesse für die Prozessmodellierung eines Unternehmens nicht relevant sind, der aber dennoch durch den Austausch von Nachrichten an einem Prozess teilnimmt.

**Lanes** Eine Lane unterteilt ein Pool in Untereinheiten. Die Unterteilung in Lanes kann nach mehreren Gesichtspunkten erfolgen, z.B. nach Rollen oder Abteilungen (vgl. OMG 2009a S. 18, 29, 87–92; Owen und Raj 2003 S. 17)

Data Object	
Group	
Annotation	 Text

Abbildung 3.8.: Artifacts in BPMN.

**Artifacts** Artefakte (Abbildung 3.8) stellen zusätzliche Informationen über einen Prozess dar. BPMN definiert drei Arten von Artefakten:

**Data Object** Ein Datenobjekt steht für eine Informationseinheit. Es kann eine Vielzahl von elektronischen oder physikalischen Objekten darstellen.

**Group** Eine Gruppe ist ein grafischer Mechanismus zur informellen Gruppierung verschiedener Objekte innerhalb eines Diagramms. Eine Gruppe kann sich über mehrere Swimlanes und Pools erstrecken. Sie beeinflusst den Ablauf des Prozesses nicht.

**Annotation** Annotationen liefern zusätzliche Informationen über den Prozess bzw. einzelne Objekte dieses Prozesses, beeinflussen diesen allerdings nicht. (vgl. OMG 2009a S. 18, 93–96)

### 3.2. *Perspektiven und Notationen*

BPMN erlaubt explizit die Verwendung unterschiedlicher Farben für Objekte, so sie sinnvoll zur Modellierung beitragen, z.B. zur Hervorhebung bestimmter Aspekte (vgl. OMG 2009a S. 29 f.).

Durch die Möglichkeiten der Gruppierung durch Pools und Lanes erleichtert BPMN im Vergleich zu EPK die Darstellung bestimmter Konzepte wie Rollen oder Abteilungen. Schnittstellen zwischen diesen Konzepten können direkt aus dem Diagramm gelesen werden. Schwächen hat BPMN beim Darstellen von Zuständen, die in EPK direkter Teil der Notation sind (vgl. Recker et al. 2005 S. 6 f.).

**Teil II.**

# **Aufgabenstellung und Umsetzung**

## 4. Problemstellung

### 4.1. Einleitung

Die *Siemens Transformers Austria* (STA) in Weiz ist ein Unternehmen, das Transformatoren produziert. Diese Tätigkeit wird durch IT-Systeme unterstützt. Neben SAP werden auch mehrere individualprogrammierte Systeme in Produktion und Vertrieb eingesetzt, die teilweise über Schnittstellen und Hilfsprogramme miteinander verbunden sind. Daneben werden Systeme verwendet, die vom Mutterkonzern Siemens vorgegeben sind und für Prozessmanagement und Projektreporting verwendet werden.

Eines der in der STA verwendeten Systeme ist die *Projektverwaltung* (PV). Dieses System wird im Bereich Vertrieb eingesetzt, um Projekte und zugehörige Daten zu verwalten. Die PV wurde als Individualsoftware in-house entwickelt.

Mitarbeiter aus dem Bereich Vertrieb identifizierten mehrere Probleme mit der PV, die teilweise aus Veränderungen in der Organisation, teilweise aus Fehlern in der Implementierung resultierten. Zwei interne Gutachten kamen zu dem Schluss, dass einer vollständigen Neuentwicklung der Vorzug gegenüber einer Überarbeitung des existierenden Systems zu geben ist.

Im Zuge dieser geplanten Neuentwicklung wurden zwei Diplomarbeiten ausgeschrieben, die mit der Analyse des IST-Zustandes und einer SOLL-Konzeption befassen sollten. In weiterer Folge wurde diese Analyse in das Projekt *PPS 15.000 MVA+* integriert, dessen Ziel eine grundlegende Erneuerung der Systemlandschaft innerhalb der STA Weiz darstellt, wobei die neu entwickelte PV (*PV-NEU*) auch zukünftig einen Teil der Systemlandschaft darstellen soll.

### 4.2. Siemens Transformers Austria

Die STA erzeugt Verteil- und Großtransformatoren. Sowohl die technische Auslegung als auch Produktion und Vertrieb der Transformatoren erfolgt am Standort Weiz. Diverse zusätzliche Dienstleistungen (z.B. Transportplanung) werden an anderen Standorten der Sparte *Siemens Energy* erbracht.

Im Gegensatz zu Verteiltransformatoren, die durch hohe Stückzahlen und die Wiederverwendung bereits existierender technischer Auslegungen dem Produktgeschäft zuzuordnen sind,

### 4.3. Projekt PPS 15.000 MVA+

zählt der Bereich der Großtransformatoren zum Projektgeschäft. Großtransformatoren werden in geringer Stückzahl geordert, oft als Einzelstücke; die technische Auslegung erfolgt nach den Vorstellungen des Kunden und kann in der Angebotsphase auch überarbeitet werden.

Projekte im Bereich Großtrafo werden von Sachbearbeitern der Abteilung *Customer Relations Offers* (CRO) behandelt. Die Sachbearbeiter erstellen auf Basis von technischen Trafoauslegungen und technischen Kalkulationen eine kaufmännische Kalkulation und in weiterer Folge ein Angebot. Der Vertrieb wird über eigenständige Länderorganisationen durchgeführt, die als rechtlicher Kunde in Erscheinung treten und als Proxy zum Käufer des Transformators dienen. Die Tätigkeit in diesem Bereich wird durch die Projektverwaltung unterstützt, die projektrelevante Daten und die kaufmännische Kalkulation verwaltet sowie Berichte und verschiedene Formen von Ansichten auf Projektdaten zur Verfügung stellt.

## 4.3. Projekt PPS 15.000 MVA+

### 4.3.1. Projektziele und -vorgaben

Im Zuge der Diplomarbeit wurde die IST- und SOLL-Analyse der PV in das Projekt PPS 15.000 MVA+ integriert. Ziel dieses Projekts ist eine Neukonzeptionierung und -implementierung der Systemlandschaft in der STA Weiz, wobei als Resultat ein integriertes Individualsystem erreicht werden soll.

Das Projekt definiert mehrere Zielvorgaben, die im Zuge der Spezifikation zu beachten waren:

- Das System soll die Benutzer unterstützen (Usability, Effizienz und Performance).
- Das System soll als integriertes System entwickelt werden und dadurch Insellösungen vermeiden.
- Compliance-Vorgaben seitens des Mutterkonzerns müssen erfüllt werden.
- Das System soll aus mehreren Teilsystemen bestehen, wobei pro Aufgabengebiet ein System führend ist.
- Die SOLL-Prozesse sollen sich stark an den IST-Prozessen orientieren und diese ohne Workarounds unterstützen, wobei etwaige Schwächen im Prozess im Zuge der Spezifikation beseitigt werden sollen.
- Das System soll stark parametrisierbar sein, um als Template-Lösung an Schwesterunternehmen verkauft werden zu können.

### 4.3. Projekt PPS 15.000 MVA+

Die Entscheidung, ob Aufgaben auf Basis des bereits eingesetzten SAP durchgeführt oder individuell entwickelt werden, soll auf Grundlage der dokumentierten Anforderungen getroffen werden.

#### 4.3.2. Zukünftige Systemlandschaft

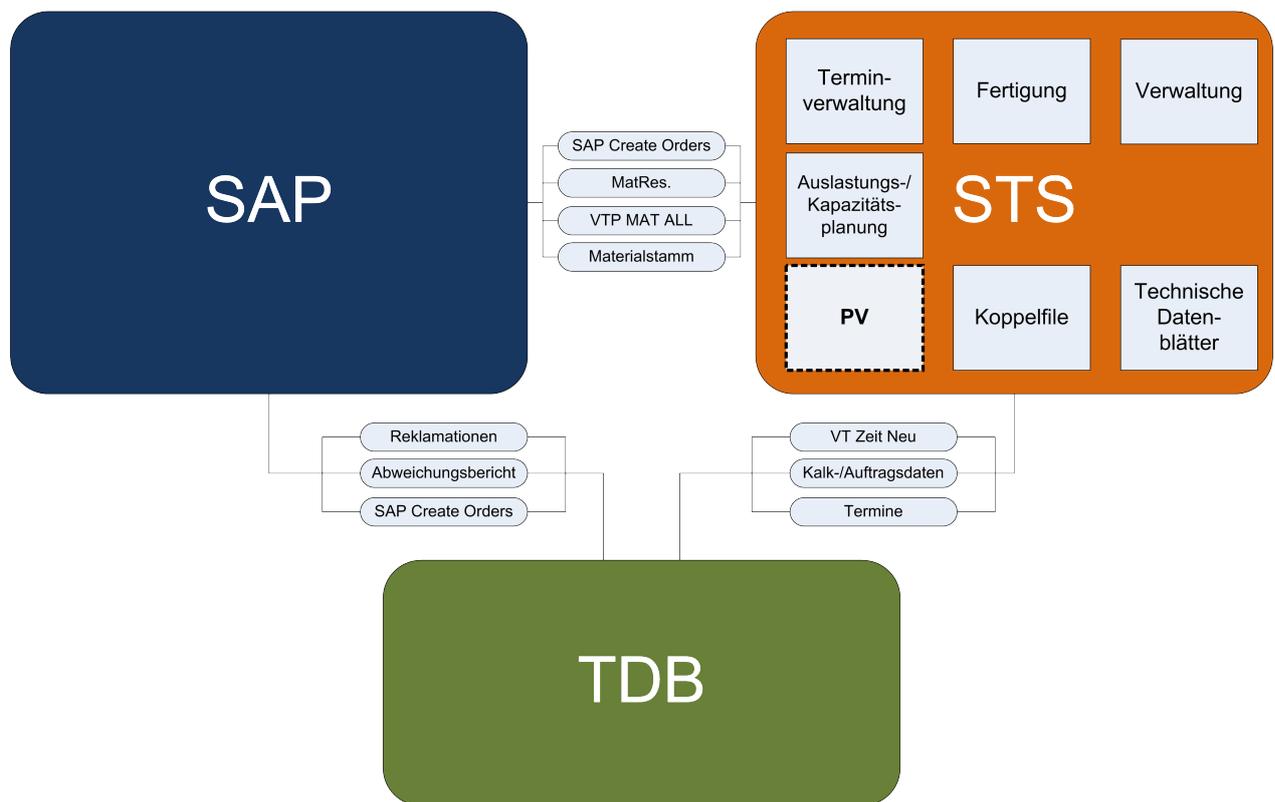


Abbildung 4.1.: Geplante Systemlandschaft innerhalb der STA Weiz (nach MCP 2009b S. 3).

Ziel ist eine dreigeteilte Systemlandschaft, die aus folgenden Komponenten bestehen soll (Abbildung 4.1):

**SAP:** Bereits in der STA eingesetzte Standard-Software.

**STS:** Siemens Transformers System, enthält Trafo-spezifische Business Intelligence

**TDB:** Technische Datenbank als gemeinsame Datenbasis

Die PV ist in dieser Systemlandschaft im Bereich STS angesiedelt, der als Kernsystem die Trafo-spezifischen Prozesse unterstützen soll.

## 4.4. Aufgabenstellung

### 4.3.3. Projektorganisation

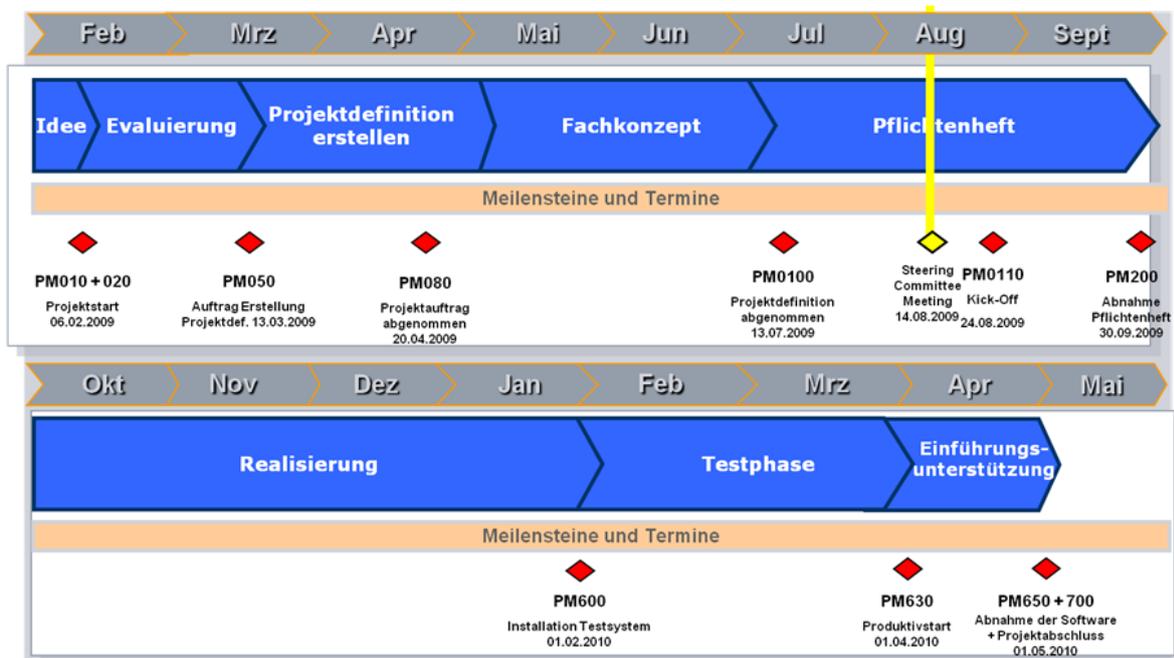


Abbildung 4.2.: Projektplan Projekt PPS 15.000 MVA+ (MCP 2009a S. 3).

Mit der Projektleitung wurde die externe Firma Management Consulting Partners (MCP) beauftragt, die in Zusammenarbeit mit der STA Weiz die Projektleitung übernahm. Die Siemens-Tochter AMC war darüber hinaus beratend tätig.

Der Projektplan sieht als Einführungstermin der neuen Softwaresysteme den 1. April 2010 vor, an dem das System abgenommen wird und in weiterer Folge im Produktiveinsatz verwendet wird (siehe auch Abbildung 4.2). Die Entwicklung des Systems wird von einem externen Lieferanten durchgeführt.

Die Analyse und Spezifikation einzelner Subsysteme erfolgt aus den zeitlichen Rahmenbedingungen parallel. Die Zusammenführung und Integration der einzelnen Systeme wird von der MCP durchgeführt. Geplant ist, dass der Softwarelieferant einzelne Teilsysteme, die bereits fertig spezifiziert und klar abgegrenzt sind, im Zuge eines iterativen Vorgehensmodells entwickelt (siehe Abschnitt 2.2.1).

## 4.4. Aufgabenstellung

Aufgabe der Diplomarbeit war die Erstellung eines Fachkonzeptes für insgesamt zwölf Module der PV, das in weiterer Folge als Basis für die Pflichtenhefterstellung gemeinsam mit dem Soft-

#### 4.4. Aufgabenstellung

warelieferanten dient. Ziel war es, relevante Punkte aus der IST-Analyse zu dokumentieren und Probleme zu identifizieren und darauf aufbauend ein SOLL-Konzept für die Projektverwaltung zu erstellen. Folgende Punkte waren im Rahmen dieser Tätigkeit zu betrachten:

- Analyse des IST-Zustandes
  - Systemlandschaft
  - Organisation
  - Prozesse
  - Datenstrukturen
  - Funktionalität
- Spezifikation des SOLL-Zustands
  - Ziele
  - Anforderungen
  - Szenarien und Testfälle
- Dokumentation der Ergebnisse

Nach einer anfänglichen Problemanalyse wurde die Projektverwaltung in 22 Module (“Arbeitspakete”) unterteilt. Die im Zuge dieser Diplomarbeit zu behandelnden Module wurden festgelegt als:

1. JourFixe-Liste
2. PM@Transformers
3. PM Vorakquisition
4. PM Projektakquisition
5. PM Angebotserstellung
6. Kalkulation
7. Scorecard/Kundenpräsente
8. Kurssicherung
9. Gleitpreis

#### *4.4. Aufgabenstellung*

10. Cashplan

11. Suchfunktion

12. Dokumentenmanagement light

## 5. Projektphasen

Die Tätigkeit im Zuge der Erstellung des Fachkonzepts für die PV wurde in mehrere Projektphasen unterteilt, die sich hinsichtlich Vorgehensweise, Zielsetzung, beteiligter Personen und Ergebnisse voneinander unterscheiden. Jede Projektphase nahm die Ergebnisse der vorhergehenden Phase als Input.

Phase	Beginn	Ende	Dauer
<i>Einarbeitungsphase</i>	06.04.2009	01.05.2009	4w
<i>IST-Analyse</i>	04.05.2009	29.05.2009	4w
<i>Grobspezifikationsphase</i>	01.06.2009	26.06.2009	4w
<i>Feinspezifikationsphase</i>	29.06.2009	07.08.2009	6w
<i>Nachbereitungsphase</i>	10.08.2009	28.08.2009	3w

**Tabelle 5.1.:** Projektphasen im Zuge der Spezifikation der PV.

Tabelle 5.1 bietet eine tabellarische Auflistung der Projektphasen samt dem Zeitraum, in dem sie durchgeführt wurden. Prinzipiell wurden je Projektphase Blöcke zu jeweils vier Wochen als zeitlicher Rahmen gewählt. Ausnahme hierzu ist die Feinspezifikationsphase mit einem Zeitrahmen von sechs Wochen. Dies war notwendig, da ein Großteil dieser Phase in die Haupturlaubszeit fiel und Gruppen von Stakeholdern nicht immer als Ganzes greifbar waren. Die Nachbereitungsphase, die für Rückfragen und Verbesserungen vorgesehen war, wurde mit drei Wochen kürzer angesetzt.

In der *Einarbeitungsphase* erfolgte das Kennenlernen des Gesamtprojekts, des Systems sowie der beteiligten Stakeholder. Die in weiterer Folge zu bearbeitenden Punkte wurden grob erfasst. In der *IST-Analysephase* wurde die PV von der Makroperspektive in verschiedenen Kontexten betrachtet. Diese Vorarbeit diente als Grundlage für die *Grobspezifikationsphase*, die den zentralen Punkt der SOLL-Spezifikation darstellte. In dieser Phase wurden die Kernpunkte der einzelnen Sollkonzepte erarbeitet und dokumentiert. Eine detaillierte Ausarbeitung dieser Punkte erfolgte in der *Feinspezifikationsphase*, an deren Ende die Fertigstellung der Sollkonzepte stand. Die *Nachbereitungsphase* wurde dazu genutzt, um missverständliche Anforderungen oder aufgetretene Inkonsistenzen informell zu klären.

## 5.1. Einarbeitungsphase

Folgend werden die jeweiligen Projektphasen im Detail vorgestellt. Tabelle 5.2 gibt einen Überblick über die Methoden, die in den jeweiligen Projektphasen eingesetzt wurden.

	Interview	Workshop	Beobachtung	Persp.bas. Lesen
<i>Einarbeitungsphase</i>	✓	✓	✓	
<i>IST-Analysephase</i>	✓		✓	✓
<i>Grobspezifikationsphase</i>	✓	✓	✓	
<i>Feinspezifikationsphase</i>	✓	✓		
<i>Nachbereitungsphase</i>	✓			

Tabelle 5.2.: Hauptsächlich eingesetzte Methoden in den jeweiligen Projektphasen.

## 5.1. Einarbeitungsphase

### 5.1.1. Zielstellung

Die Einarbeitungsphase diente dem Kennenlernen des Projekts, der Zielsetzung sowie der beteiligten Stakeholder. Ziel war es, einen groben Überblick über die PV zu erhalten und die Funktionen sowie Probleme der PV überblicksartig zu analysieren. Darüber hinaus hatte diese Phase zum Ziel, Anforderungsquellen für eine weitergehende Analyse sowie die Spezifikationsphase zu gewinnen.

### 5.1.2. Vorgangsweise

In dieser Phase wurden anfänglich hauptsächlich Besprechungen mit Stakeholdern aus dem Bereich Vertrieb abgehalten. Diese Besprechungen wurden teilweise im Workshop-Format, teilweise auch im offenen Interview-Format (oder einer Mischform aus beiden) geführt. Innerhalb dieser Meetings wurden die grundlegenden Eckpunkte des Teilprojekts PV besprochen. Im Mittelpunkt standen dabei die bisherige PV sowie die grundlegenden Abläufe und Tätigkeiten, die mit diesem System durchgeführt werden. Darüber hinaus wurden Problempunkte der jetzigen PV sowie weitere Stakeholder, die im Laufe des Projekts als Anforderungsquellen dienen können, identifiziert.

In weiterer Folge wurden mit diesen Stakeholdern in kleinen Gruppen oder als Einzelperson offene Interviews geführt, mit denen dringende Probleme und essentiell benötigte Funktionalität identifiziert wurden. In kleinerem Rahmen wurden direkte Beobachtungen durchgeführt, um ein besseres Verständnis für die Arbeit mit der PV zu erreichen.

## 5.2. IST-Analysephase

Als besonders effektiv erwiesen sich in dieser Phase das Workshop-Format bzw. offene Interviews, da beide Methoden direktes Feedback auf Fragen ermöglichten. Missverständnisse bzw. Verständnisprobleme konnten auf diesem Weg schnell ausgeräumt werden. Durch das Einbinden einer größeren Anzahl von Stakeholdern in mehreren Etappen konnte auch der projektbezogene Zeitaufwand für einzelne Stakeholder verringert werden.

### 5.1.3. Ergebnis

Ergebnis dieser Phase war eine 17-seitige Dokumentation der Kernpunkte der PV sowie deren grundlegende Probleme und die daraus resultierenden Änderungswünsche. Diese Dokumentation entstand aus den Protokollen der Workshops und Interviews. Darüber hinaus wurden in dieser Projektphase die wesentlichen Anforderungsquellen für die weiteren Projektschritte identifiziert. Diese Anforderungsquellen sind in Abbildung 5.1 dargestellt.

<b>Gegenstandsfacette</b> Martin Kalcher (Vertrieb) Emanuel Zechner (Vertrieb) Steinort Eberhard (Cashplanung) Werner Huber (Cashplanung) Christian Nell (Controlling) Herbert Thier (technische Kalkulation) Dokumentation PM@Transformers Dokumentation OO4T-Schnittstelle Problemanalyse PV Bedienungsanleitung PV	<b>Nutzungsfacette</b> Martin Kalcher (Vertrieb) Emanuel Zechner (Vertrieb) Steinort Eberhard (Cashplanung) Mitarbeiter Vertrieb
<b>IT-Systemfacette</b> Thomas Kober (IT) Michael Gerstl (IT)	<b>Entwicklungsfacette</b> Thomas Kober (IT) Michael Gerstl (IT) AMC

Abbildung 5.1.: Identifizierte Anforderungsquellen.

## 5.2. IST-Analysephase

### 5.2.1. Zielstellung

Ziel der IST-Analyse war es, einen möglichst umfassenden Überblick über die PV auf Makroebene zu gewinnen. Dabei sollte die PV aus mehreren Perspektiven betrachtet werden. Zusätzlich sollten innerbetrieblich vorhandene Dokumente, die zuvor als potentielle Anforderungsquellen definiert wurden, nach relevanten Informationen gesichtet werden.

### 5.3. Grobspezifikationsphase

Die PV sollte in folgenden Kontexten betrachtet werden:

- Systemkontext
- Organisationskontext
- Funktionskontext
- Datenkontext
- Prozesskontext

#### 5.2.2. Vorgangsweise

In der IST-Analyse wurden mehrfach offene Interviews eingesetzt. Diese Interviews fanden hauptsächlich in Einzelgesprächen statt. Für den Daten- und Systemkontext wurden Stakeholder aus der IT-Systemfacette befragt, für den Organisations- und Prozesskontext Personen aus der Gegenstandsfacette.

Der Funktionskontext wurde hauptsächlich durch Beobachtungen erforscht. Die schon in der Einarbeitungsphase eingesetzten direkten Beobachtungen wurden in weiterer Folge durch ethnografische Beobachtungen abgelöst. Dabei wurden auf einem Testsystem typische Aufgaben eines Vertriebsmitarbeiters in der PV durchgeführt. Zusätzlich erfolgte eine strukturierte Abarbeitung in der PV vorhandener Menüpunkte, die protokolliert und anschließend in Themengebiete gruppiert wurden.

Für das Sichten der Dokumente erwies sich perspektivenbasiertes Lesen als zielführende Vorgangsweise. Hierbei wurden die Dokumente, die in der Einarbeitungsphase bzw. in dieser Phase identifiziert wurden, aus verschiedenen Blickwinkeln gelesen, um daraus die Informationen, die für das weitere Projekt relevant sein könnten, zu extrahieren.

#### 5.2.3. Ergebnis

Das Ergebnis dieser Phase war eine Beschreibung des IST-Systems aus verschiedenen Blickwinkeln. Einige Ergebnisse dieser Projektphase sind in Abschnitt 6 beschrieben.

## 5.3. Grobspezifikationsphase

### 5.3.1. Zielstellung

In dieser Phase erfolgte eine Aufteilung der PV in 22 Module, die eigenständig zu bearbeiten waren. Die zwölf Module, die im Rahmen dieser Arbeit behandelt wurden, sind in Abschnitt 4.4 aufgelistet.

### 5.3. Grobspezifikationsphase

Ziel dieser Phase war es, die einzelnen Module grob zu spezifizieren. Dazu gehörte eine detaillierte IST-Analyse, in der die bisher vorhandene Funktionalität des jeweiligen Moduls erfasst wurde und deren Probleme identifiziert wurden. Darüber hinaus sollten die zugehörigen von der PV unterstützten bzw. zu unterstützenden Prozesse betrachtet werden. Die Ziele und grundlegenden Bedingungen (Anforderungen) an das SOLL-System sollten ebenso erfasst werden wie die geplanten Interaktionen mit dem System (Szenarien).

#### 5.3.2. Vorgangsweise

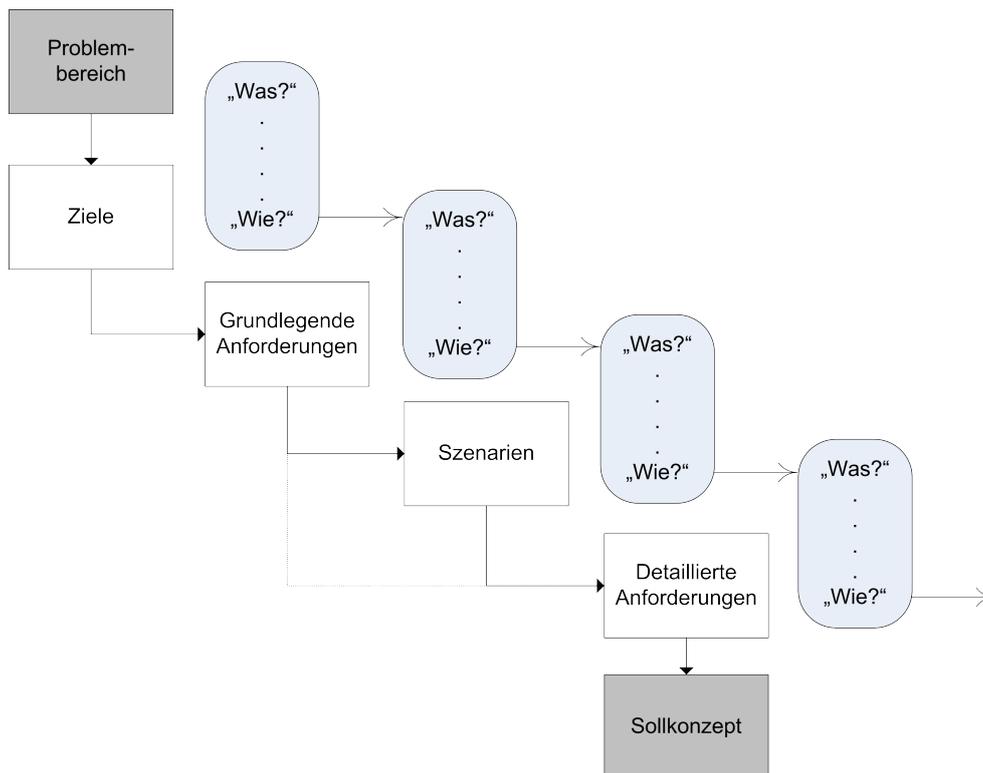


Abbildung 5.2.: Iterative Vorgangsweise zur Erarbeitung des Sollkonzeptes.

Am Beginn dieser Phase erfolgte die Aufteilung der PV in 22 Module (Arbeitspakete). Diese Aufteilung basierte auf den bisher gewonnenen Informationen. In einem Workshop mit den Stakeholdern aus der Nutzungs- und Gegenstandsfacette (siehe Abbildung 5.1), die in der Regel auch die Endbenutzer des System darstellten, wurde diese Aufteilung besprochen und die zu bearbeitenden Punkte der einzelnen Module diskutiert und verfeinert. Im Anschluss erfolgte die Zuweisung von zwei bis drei Stakeholdern pro Modul, die die Spezifikation gemeinsam mit dem Diplomanden erarbeiten sollten. Ebenfalls erfolgte eine grobe Priorisierung der einzelnen

### 5.3. Grobspezifikationsphase

Module in die Prioritätsklassen 1, 2 und 3, wobei 1 die höchste Prioritätsstufe darstellte.

Die zentrale Methodik dieser Projektphase war die Durchführung von Workshops in Kleingruppen. Für jedes Modul wurde ein eigener Workshop abgehalten, in dem gemeinsam mit den zugewiesenen Stakeholdern ein Sollkonzept erarbeitet wurde. Die Erstellung der Konzepte erfolgte in iterativen Schritten (siehe auch Abschnitt 2.2.4), wobei die “Lösung” eines Schrittes das zu lösende Problem für den folgenden Schritt darstellte. Abbildung 5.2 illustriert diese Vorgangsweise.

Ausgehend vom Problembereich des jeweiligen Moduls wurde in den Workshops besprochen, welche Aufgaben die einzelnen Module in Zukunft erfüllen sollen. Diese Aufgaben wurden in Form von Zielen formuliert, aus denen im nächsten Schritt die grundlegenden Anforderungen an das jeweilige Modul abgeleitet wurden. Um den Mehrwert des Sollkonzepts im Vergleich zur jetzigen PV zu identifizieren, wurde die IST-Situation ebenfalls in diese Betrachtungen einbezogen. Die abstrakte Definition von Zielen und Anforderungen wurden durch wesentliche Interaktionen mit dem System in Form von deskriptiven Szenarien konkretisiert. Diese SOLL-Abläufe dienten auch als Basis für Diskussionen.

Die Ergebnisse der Workshops wurden protokolliert und in weiterer Folge ohne Einbeziehung der Stakeholder analysiert, ausgearbeitet und dokumentiert. Aus den besprochenen Zielen, Szenarien und grundlegenden Anforderungen wurden verfeinerte Anforderungen an das zukünftige System abgeleitet. Offene Punkte wurden identifiziert und, sofern sie sich nicht durch informelles Nachfragen klären ließen, als solche markiert. Mittels Einzelinterviews wurden Details des zugrunde liegenden Prozesses von den Stakeholdern erfragt.

Für eine Validierung wurden die in dieser Form ausgearbeiteten Sollkonzepte den beteiligten Stakeholder vorgelegt, die ihrerseits einzeln und selbstständig Reviews durchführten. Die nicht am jeweiligen Modul beteiligten Stakeholder wurden durch ein Meeting in der Großgruppe an der Validierung beteiligt. Hierbei wurden (ähnlich einem Walkthrough) für jedes Modul die identifizierten Ziele und wesentlichen Anforderungen vorgestellt. In einer Diskussion konnten die Stakeholder zu diesen Punkten Stellung nehmen und Fragen stellen. Die einzelnen Sollkonzepte wurden auf Basis dieses Meetings aktualisiert.

#### 5.3.3. Ergebnis

Ergebnis dieser Phase waren zwölf Sollkonzepte, die die IST-Situation des jeweiligen Moduls identifizieren sowie Ziele, Szenarien und Anforderungen an das SOLL-System beschreiben. Auszugsweise ist dieses Ergebnis in Abschnitt 7 zu finden.

## 5.4. Feinspezifikationsphase

### 5.4.1. Zielstellung

Diese Phase hatte die Finalisierung der Sollkonzepte zum Ziel. Dies bedeutete die Verfeinerung der Ziele und Anforderungen sowie die Umwandlung der bisher deskriptiven Szenarien in ein strukturiertes Format. Darüber hinaus hatte diese Projektphase zum Ziel, bisher offene Punkte zu klären. In Absprache mit der Projektleitung wurden einzelne offene Punkte zur Ausarbeitung im Zuge des weiteren Projektverlaufs vermerkt.

### 5.4.2. Vorgangsweise

Auch in dieser Projektphase waren Kleingruppen-Workshops die zentrale Methodik. Diese Workshops hatten zweierlei Ziel. Einerseits dienten sie der Validierung der ausgearbeiteten Sollspezifikationen. Andererseits wurden offene Punkte diskutiert und spezifiziert.

Als Einleitung dieser Workshops dienten die bereits ausgearbeiteten Szenarien. Diese wurden einerseits eingesetzt, um die Workshop-Teilnehmer auf den aktuellen Stand des Konzepts zu bringen. Andererseits wurden dadurch die SOLL-Abläufe überprüft, um sicherzustellen, dass diese auch dem Wunsch der zukünftigen Benutzer des Systems entsprechen. In weiterer Folge wurden die zu den jeweiligen Szenarien zugehörigen Ziele und Anforderungen betrachtet und offene Punkte geklärt.

Besprochene Veränderungen wurden vermerkt und anschließend in die Sollkonzepte eingearbeitet. Danach erfolgte eine Übertragung der Szenarien in ein strukturiertes Format. Details zu einzelnen Modulen wurden mit Stakeholdern in Einzelinterviews geklärt. Schlussendlich wurden die Sollkonzepte den Stakeholdern zur Durchsicht zur Verfügung gestellt. In weiterer Folge wurden diese Konzepte an die Projektleitung zur Integration in ein Gesamtdokument übergeben.

### 5.4.3. Ergebnis

Resultat dieser Phase war die finalisierte Version der Sollkonzepte, die sowohl die Beschreibung der IST-Situation als auch eine detaillierte Ausarbeitung des von den Stakeholdern gewünschten SOLL-Zustandes beinhaltet. Diese Konzepte werden in weiterer Folge zur Erstellung des Pflichtenheftes verwendet. Auszugsweise sind die Ergebnisse dieser Phase in Abschnitt 7 zu finden.

## 5.5. Nachbereitungsphase

In der Nachbereitungsphase erfolgte die Integration der 22 Sollkonzepte zu einem Gesamtdokument. Diese Tätigkeit wurde von der Projektleitung durchgeführt. Im Zuge dieser Integration

### *5.5. Nachbereitungsphase*

erfolgte ein Review der Konzepte. Unklarheiten und Inkonsistenzen in einzelnen Konzepten wurden informell mit der Projektleitung geklärt.

**Teil III.**

**Ergebnis**

## 6. IST-Analyse

Dieser Abschnitt beinhaltet das Ergebnis der IST-Analyse aus verschiedenen Blickwinkeln.

### 6.1. PV im Systemkontext

#### 6.1.1. Technische Eckdaten der PV

Die PV ist ein innerhalb der STA Weiz entwickeltes und gewartetes System. Die Software ist in VisualBasic 6.0 geschrieben und umfasst etwa 160.000 Codezeilen. Das System basiert auf einer zweischichtigen Architektur. Darstellung und Geschäftslogik sind in der Client-Applikation implementiert und werden auf den Workstations der Benutzer ausgeführt. Die Clientsoftware ist über eine ODBC-Schnittstelle mit einer zentralen Datenbank verbunden.

Zentrale Datenbank ist die *Transformatordatenbank (TRDB)*. Als Datenbanksystem wird eine Oracle Rdb verwendet. Das relationale Datenbanksystem Rdb wurde 1994 von Oracle gekauft und ist nicht verwandt mit anderen Datenbankprodukten von Oracle (vgl. Oracle 2009). Die Datenbank läuft auf Rechnern mit einer VAX-Architektur. Als Abfragesprache wird SQL verwendet. Die Verwendung eines grafischen Frontends zur Datenbankverwaltung ist nicht möglich (auch nicht über die ODBC-Schnittstelle). Dafür bietet das System die Möglichkeit, konsolenbasierte Abfragemasken (*Rally-Masken*) zu erzeugen, mit denen Daten manipuliert werden können. Diese Masken sind in der STA noch vereinzelt im Einsatz, unter anderem, weil sie gegenüber anderen Abfragen höher priorisiert sind und deshalb eine schnellere Antwortzeit bieten.

Aufgrund der Tatsache, dass die Produktion von VAX-Systemen seit einigen Jahren eingestellt ist, plant die STA die Migration der Datenbank auf einen MS Access-Server. Die Migration soll parallel zur Neuentwicklung der PV erfolgen, sodass die PV-NEU auf dem neuen Datenbanksystem basieren wird.

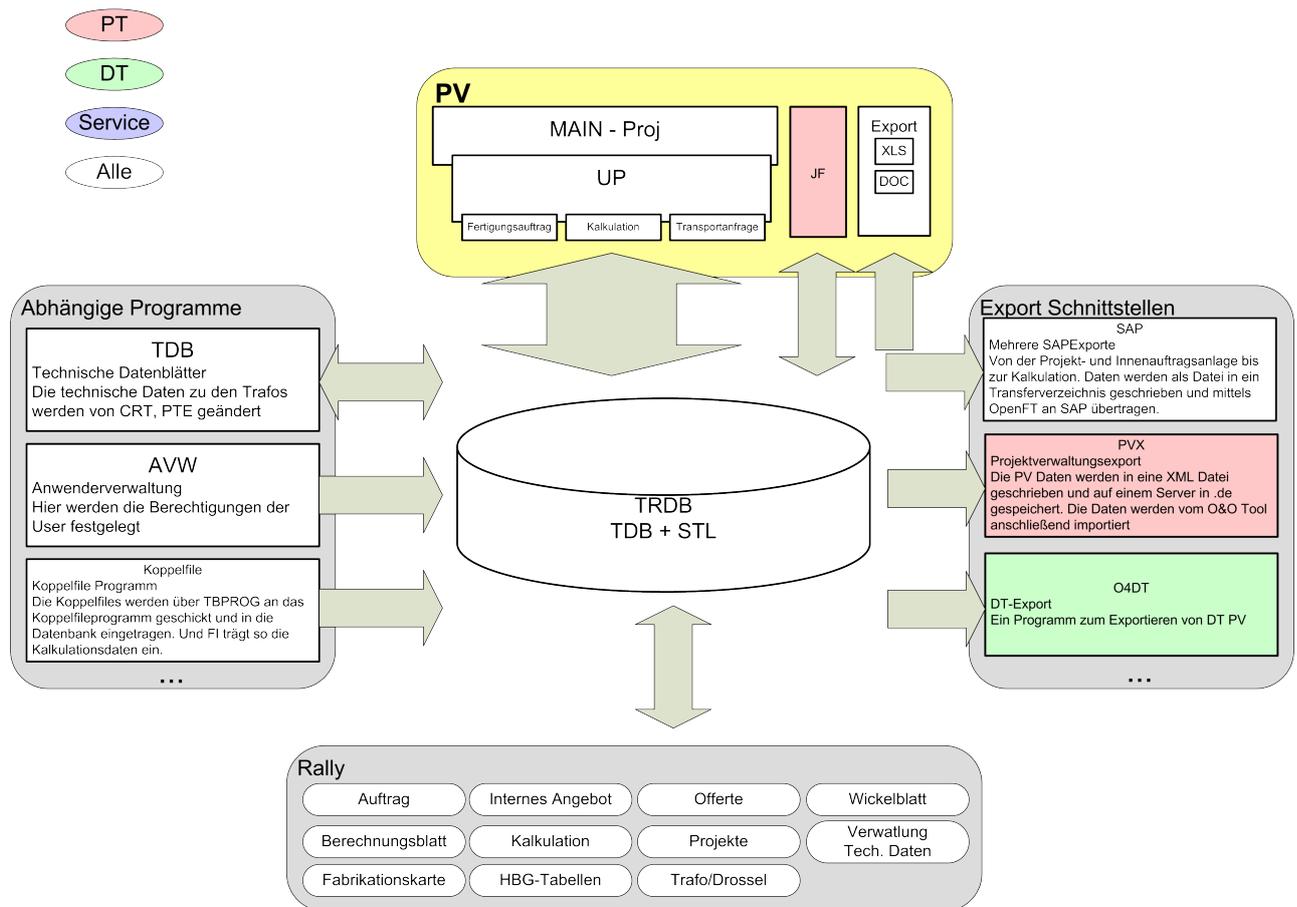
Der Code der PV hat einige Probleme, die eine Weiterentwicklung nicht sinnvoll erscheinen lassen. Dazu gehören einerseits Probleme mit der Performance, andererseits strukturelle Probleme innerhalb der Software, wodurch Änderungen und Fehlerbehebungen in bestimmten Bereichen mit sehr großem Aufwand verbunden sind. Ursache der Probleme ist unter anderem das mehrmalige Wechseln von Entwicklern. Unklare und sich oft ändernde Benutzeranforderungen (von den Entwicklern der PV auch als „Wurlitzerprogrammierung“ bezeichnet) sind ebenfalls

## 6.1. PV im Systemkontext

mit ein Grund der Probleme. Darüber hinaus existieren in der Systemlandschaft Altlasten, die Designentscheidungen bei der Entwicklung der PV einschränken und somit Workarounds verursachen.

### 6.1.2. Schnittstellen zu Drittsystemen

Die PV ist über Schnittstellen mit anderen Systemen der STA Weiz verbunden. Abbildung 6.1 veranschaulicht diese Systemlandschaft.



**Abbildung 6.1.:** Die PV im Kontext der Systemlandschaft der STA Weiz (aus Siemens Transformers Austria 2009 S. 1).

**Exportschnittstellen** Eine für die PV wesentliche Schnittstelle ist die Schnittstelle *PVX*, die Projektdaten in das *Offers&Orders Tool for Transformers (OO4T)* exportiert. OO4T ist ein von der Siemens Energy vorgegebenes System, das Projektdaten aggregiert und für deren Auswertungen und Analyse zuständig ist. Diese XML-Schnittstelle ist für die PV deshalb von

## 6.2. PV im Organisationskontext

Bedeutung, da sie Anforderungen an eine Projektstruktur vorgibt, insbesondere Pflichtfelder in Abhängigkeit von Projektmeilensteinen. In weiterer Folge wird aus dem OO4T das System *PM@Transformers* mit einigen aus der PV übernommenen Daten befüllt. *PM@Transformers* dient als Ablage relevanter Projektdaten und -dokumente zu Auditzwecken. Für die Abteilung Verteiltrafo wird eine eigene Schnittstelle verwendet, die bereits aggregierte Projektdaten exportiert.

Die Schnittstelle zu SAP überträgt Projektdaten in dieses System. Der Export an SAP erfolgt, sobald ein Angebot zu einem Auftrag wurde. Übertragene Daten sind beispielsweise allgemeine Projektinformationen und die Projektkalkulation.

**Importschnittstellen** Die Anwenderverwaltung verwaltet als externes System die Anwendungsberechtigungen für Systeme innerhalb der STA Weiz und damit auch für die PV. Der Login funktioniert über den Windows-Benutzernamen. Administratoren haben einen speziellen Login und können damit auch die Rolle eines anderen Benutzers einnehmen.

Technische Daten werden nicht direkt in der PV verwaltet, sondern im technischen Datenblattprogramm. Die Verknüpfung zu diesen Daten erfolgt über die sogenannte TBVA-Nummer. Technische Kalkulationen, die ebenfalls extern erstellt werden, werden über das Koppelfileprogramm importiert. Dies geschieht über ein spezielles Dateiformat, das Koppelfile.

## 6.2. PV im Organisationskontext

Die STA Weiz ist in mehrere organisatorische Einheiten aufgeteilt. Wesentlich für die PV ist die Abteilung Großtrafo (PT), dem hauptsächlichen Einsatzgebiet der PV. In dieser Abteilung werden von den Mitarbeitern der Abteilung CRO Angebote aufgenommen und im Zuge dessen Verhandlungen mit Kunden geführt. Das Projektcontrolling (CRC) ist für die Wirtschaftlichkeit von Trafoprojekten verantwortlich. In der Technik (CRT) werden Trafos ausgelegt, in weiterer Folge führen Mitarbeiter der Abteilung CAE die technische Kalkulation durch. CRS ist für Montage und Service von Trafos verantwortlich. Die Logistik-Abteilung Linz ist für den Transport von Transformatoren zuständig.

## 6.3. Funktionalität der PV

Die PV bietet unterschiedliche Funktionen zur Unterstützung der Arbeit von Vertriebsmitarbeitern. Seite 96 enthält eine grafische Darstellung.

Die wesentliche Funktionalität der PV ist, wie der Name des Systems, impliziert, das Verwalten von Projekten. Mithilfe der PV lassen sich neue Projekte anlegen, darüber hinaus kann nach

### 6.3. Funktionalität der PV

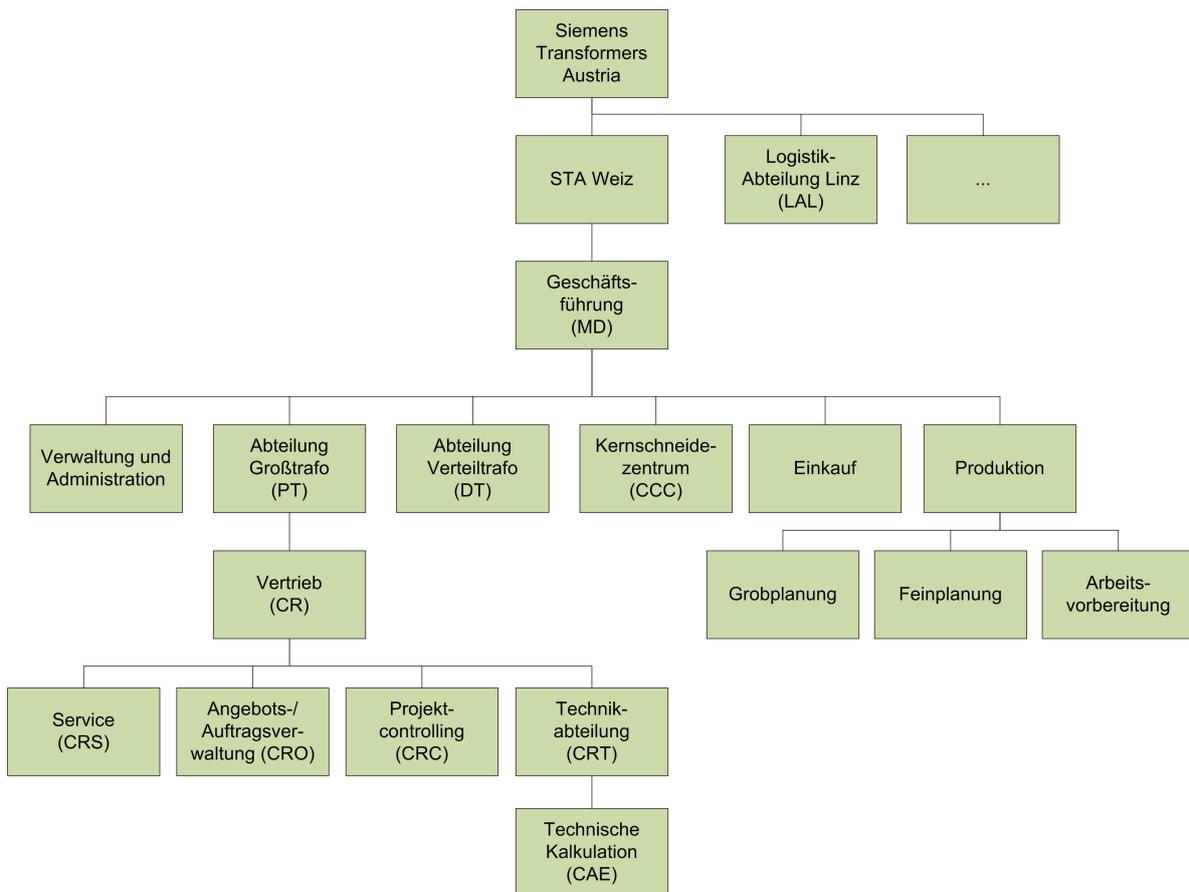


Abbildung 6.2.: Organisation der STA.

bereits existierenden Projekten mithilfe mehrerer Parameter gesucht werden. In diesen Projekten können zentrale Projektdaten bearbeitet werden: Der Endkunde und der rechtliche Kunde sowie Due Dates und der Projektfortschritt (in Form von Milestones und als Projektwahrscheinlichkeit). Projekte können ebenfalls kopiert werden. In einem Projekt lassen sich Subprojekte anlegen, die entweder neu erstellt oder aus einem bereits existierenden Subprojekt kopiert werden. Zu diesem Subprojekt lassen sich Daten wie der Trafotyp, Versandbedingungen sowie eine Kalkulation ablegen.

Zusätzlich bietet die PV Funktionalität zur Verwaltung von Kunden. Der Benutzer kann neue Kunden anlegen und nach existierenden suchen. Die Umsetzung dieser Funktionalität hat allerdings zu Mehrfacheinträgen vieler Kunden geführt, was automatische Auswertungen stark erschwert. Ein weiterer zentraler Punkt ist die JourFixe-Liste. In dieser Liste lassen sich Subprojekte, die bestimmte Filterbedingungen erfüllen, tabellarisch auflisten. Darüber hinaus bietet diese Ansicht die Möglichkeit, bestimmte Felder der angezeigten Subprojekte direkt zu editieren.

## 6.4. Daten innerhalb der PV

Schlussendlich erlaubt die PV den Export einer Anzahl von Berichten, die aus den Projektdaten generiert werden. Die Berichte können im PDF-, Word- oder Excel-Format exportiert werden.

### 6.4. Daten innerhalb der PV

Die Daten innerhalb der PV sind in der in Abschnitt 6.1 beschriebenen Datenbank TRDB abgelegt. Das Datenmodell in dieser Datenbank existiert in Teilen bereits seit mehreren Jahrzehnten. Abbildung 6.3 zeigt eine Skizze einiger Datenbanktabellen, die im Zuge der Erarbeitung der Datenstrukturen entstanden ist.

Seite 97 zeigt das Datenmodell in Form eines ER-Diagramms. Die Notation wurde um Aggregation und Generalisierung (siehe Abschnitt 3.2.2) erweitert, die die grafische Beschreibung des Datenmodells erleichtern. Die Beschreibung des Datenmodells basiert nicht direkt auf der Datenstruktur der Datenbank, die teilweise auf Vorgängersystemen basiert und Probleme wie regelmäßig überlaufende Schlüsselfelder sowie fehlende Absicherung referentieller Integrität aufweist. Stattdessen bietet das Modell eine abstraktere Sicht auf die Daten, die der Präsentation dieser Daten durch die PV entspricht. Aus Gründen der Übersichtlichkeit wurde zusätzlich auf einige Attribute der Objekte verzichtet.

Das zentrale Objekt der PV ist das Projekt. Ein Projekt wird von einem Mitarbeiter (Benutzer) verwaltet, hat einen Namen, einen Kunden und eine Länderorganisation als rechtlichen Kunden. Ein Projekt besteht aus einem oder mehreren Subprojekten, die entweder Angebot oder Auftrag darstellen können. Dieses Subprojekt fasst alle relevanten Daten für dieses Angebot oder den Auftrag zusammen. Dies sind Informationen wie der Subprojektstatus, Milestone und Versandbedingungen. Darüber hinaus beinhaltet ein Subprojekt ein oder mehrere Trafos. Zu diesen Trafos gehören technische Daten des Trafos (die technische Trafoauslegung selbst ist nicht mehr Teil der PV) sowie die Kalkulation.

Die Kalkulation besteht einerseits aus den Herstellkosten (technische Kalkulation, wird in einem externen System berechnet), andererseits aus der kaufmännischen Kalkulation. Die kaufmännische Kalkulation beinhaltet im Vertrieb zu ermittelnde Positionen wie Lieferkosten, Risikoaufschlag, Versicherungskosten und Kommission.

### 6.5. Prozesse innerhalb der PV

Der Prozess innerhalb der STA Weiz entspricht im Wesentlichen der Vorgabe des Mutterkonzerns Siemens, die in Abbildung 6.4 als Value Chain Diagram dargestellt ist. Diese Vorgabe entspricht auch der Struktur des in der STA Weiz eingesetzten Programms PM@Transformers. Sie definiert

## 6.5. Prozesse innerhalb der PV

die Projektschritte zur Akquisition und Abwicklung eines Trafoprojektes und die zugehörigen Meilensteine.

Die für die PV relevanten Schritte reichen von der Vorakquisition bis hin zur Projekteröffnung und Klärung. Die nachfolgenden Schritte werden von anderen Systemen (hauptsächlich SAP) unterstützt. Im Folgenden sind die für die PV relevanten Schritte beschrieben:

**Vorakquisition** In diesem Schritt wird die Geschäftsmöglichkeit identifiziert und analysiert sowie die Geschäftserwartung bewertet. Anschließend wird eine Go/No Go-Entscheidung zur weiteren Abwicklung des Projektes getroffen.

**Projekt-Akquisition** Hierbei wird die Geschäftserwartung analysiert sowie die Realisierbarkeit geprüft. Anschließend wird eine Bid/No Bid-Entscheidung getroffen.

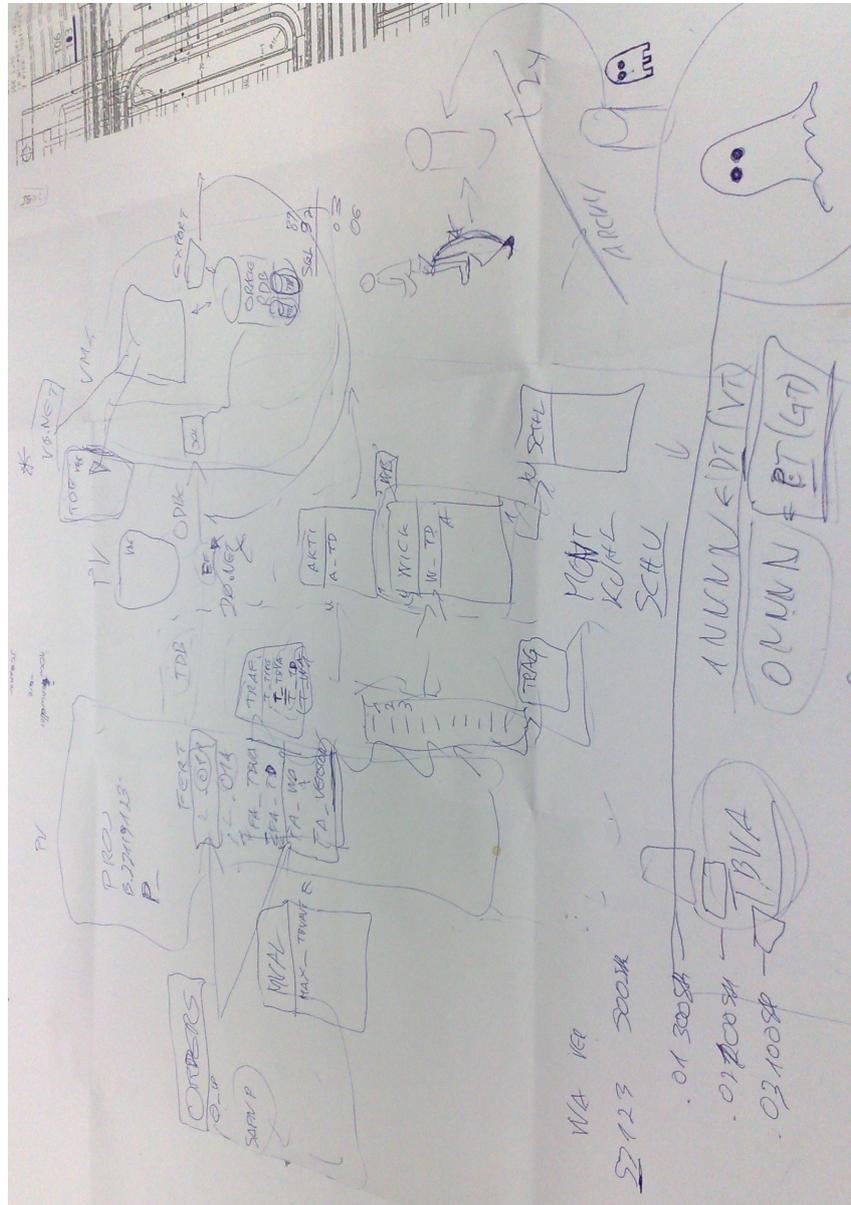
**Angebotserstellung** Bei der Angebotserstellung wird die technische Konstruktion durchgeführt, darauf basierend werden die Herstellkosten ermittelt. Zusammen mit der Kalkulation der Transportkosten dient dies als Basis für die Angebotskalkulation, die ebenfalls in diesem Schritt erstellt wird.

**Vertragsverhandlung** In diesem Schritt wird das Angebot übergeben, darauf basierend werden Vertragsverhandlungen geführt. Schlussendlich wird ein abschließender Vertrag erstellt.

**Projektübergabe** Hierbei werden Übergabedokumente erstellt und das Projekt wird an einen Projektleiter übergeben.

**Projekteröffnung & Klärung** Der Schritt der Projekteröffnung und Klärung umfasst das Erstellen eines Projektplans sowie eine Analyse des Vertrags. Außerdem wird eine Auftragseingangskalkulation durchgeführt (vgl. Siemens 2008 S. 22–28).

## 6.5. Prozesse innerhalb der PV



**Abbildung 6.3.:** Diese Skizze der Struktur der Daten in der STA entstand bei der Erarbeitung des Datenmodells, die mit einem IT-Experten der STA durchgeführt wurde. Die Skizze zeigt einige Datenbanktabellen aus dem Bereich PV sowie zugehörige Tabellen aus anderen Systemen.

## 6.5. Prozesse innerhalb der PV

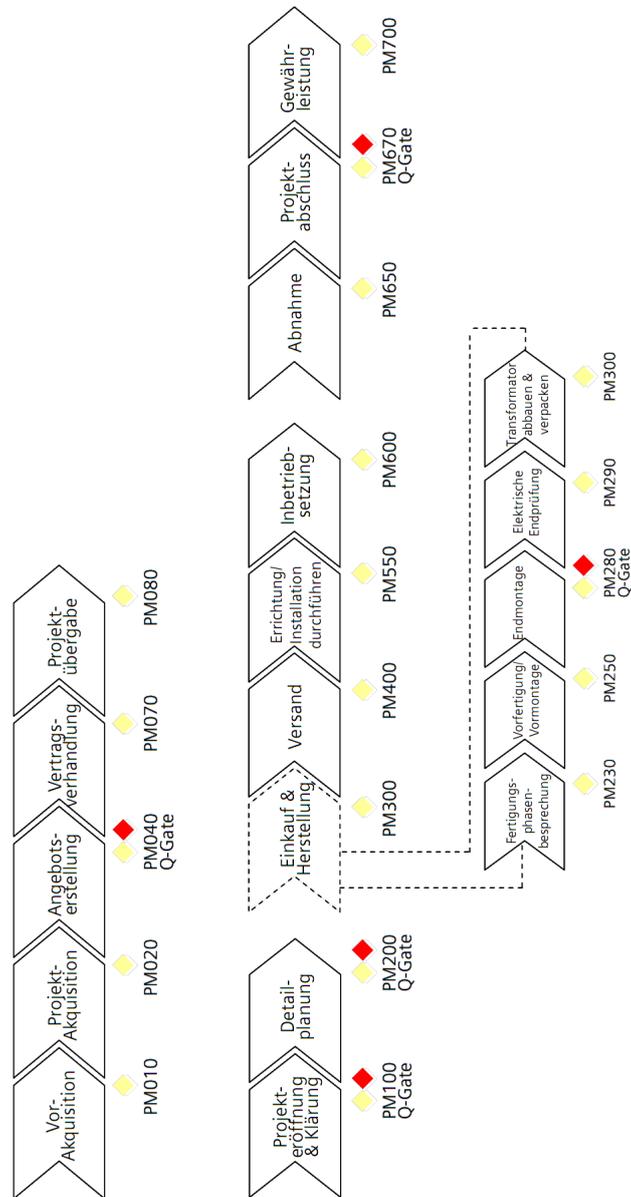


Abbildung 6.4.: Akquisitions- und Projektabwicklungsprozess (aus Siemens 2008 S. 21).

## 7. Fallbeispiel: Modul Kalkulation

Dieser Abschnitt beschreibt das Modul Kalkulation, eines der zwölf konzeptionierten Module. Die Ausarbeitung des Moduls ist in eine Analyse des IST-Zustandes (Abschnitt 7.1) sowie ein SOLL-Konzept (Abschnitt 7.2) unterteilt.

Dieses Modul hat in Teilen Abhängigkeiten zu anderen, in dieser Arbeit nicht präsentierten Modulen. Nicht Teil dieses Moduls ist die Gleitpreiskalkulation, die hier dargestellt Kalkulation um bestimmte variable Teile ergänzt. Die Gleitpreiskalkulation wurde als selbständiges Modul konzeptioniert und basiert auf der Kalkulation.

Aus Datenschutzgründen wurden die Abbildungen in diesem Abschnitt in Teilen geschwärzt.

### 7.1. IST-Zustand

#### 7.1.1. Übersicht

Die Kalkulation von Trafo-Projekten ist ein wesentlicher Bestandteil der PV. Damit werden die Kosten eines Trafo-Projektes geschätzt und ein Angebotspreis ermittelt, der sich aus den Kosten sowie einem Gewinnzuschlag ergibt.

Eine Kalkulation ist dabei einem *Subprojekt* zugeordnet. Sie wird bis zum Auftragsfall in der PV gewartet. Eine nachträgliche Kalkulation zur Ermittlung der tatsächlichen Kosten findet in SAP statt.

#### 7.1.2. Arten der Kalkulation

Die Kalkulation in der PV besteht aus zwei Teilen: Einer technischen Kalkulation und einer kaufmännischen Kalkulation. Aus diesen zwei Teilen werden die Gesamtkosten eines Großtrafo-Projektes berechnet. Zuzüglich eines einberechneten Gewinnes ergibt sich daraus der anvisierte Verkaufspreis.

#### Technische Kalkulation

Die technische Kalkulation wird von einem Mitarbeiter von CRT durchgeführt und enthält Herstellkosten wie:

## 7.1. IST-Zustand

- Produktionskosten
- Materialkosten
- Verpackung

### **Kaufmännische Kalkulation**

Die kaufmännische Kalkulation wird von einem Mitarbeiter von CRO durchgeführt und enthält Positionen wie:

- Risikozuschläge
- Versicherung
- Lieferkosten

### **Ablauf der Kalkulation**

Im Zuge des Vertriebsprozesses wird in der PV eine Reihe von Kalkulationen zu unterschiedlichen Zeitpunkten durchgeführt.

**Budget Calculation** Erste, grobe Abschätzung der Werksherstellkosten auf Basis ähnlicher Projekte. Dieser Schritt wird nicht bei jedem Projekt durchgeführt, sondern hängt von der Kundenabfrage ab.

**Angebotskalkulation** Gilt als Grundlage zur Angebotslegung.

**Auftragseingangskalkulation** Nachkalkulation zum Zeitpunkt des Auftragseingangs.

### **7.1.3. Struktur der Kalkulation**

Die aktuelle Kalkulation wird in der PV mittels drei Formularblättern abgebildet:

- Factory Production Cost (Abbildungen 7.1 und 7.2)
- Factory Prime Cost (Abbildung 7.3)
- Total Cost (Abbildung 7.4)

Die ersten zwei Blätter enthalten dabei die Herstellkosten. Das dritte Blatt repräsentiert die kaufmännische Kalkulation sowie den anvisierten Verkaufspreis.

Neben diesen in der PV durchgeführten Berechnungen werden die zwei Berechnungsschritte Risikokalkulation und On-/Off-Shore-Split extern durchgeführt und anschließend in die PV übertragen. Diese zwei Schritte sind in Abschnitt 7.1.4 beschrieben.

## 7.1. IST-Zustand

### **Herstellkosten**

Die Blätter „Factory Production Cost“ und „Factory Prime Cost“ enthalten die Herstellkosten eines Subprojektes (ein oder mehrere Stück einer bestimmten Trafo-Auslegung). Diese extern berechneten Kosten werden auf Wunsch des bearbeitenden CRO-Mitarbeiters aus GROTWI importiert.

Die Herstellkosten sind in variable und in fixe Anteile aufgliedert. Darüber hinaus hat der Benutzer die Möglichkeit, die Herstellkosten um einen prozentuellen Betrag zu reduzieren (oder zu erhöhen) sowie einzelne Positionen absolut zu überschreiben. Außerdem wird neben den Gesamtherstellkosten auch eine stückweise Aufgliederung angeboten.

### **Kaufmännische Kalkulation**

Das Blatt „Total Costs“ (Abbildung 7.4) enthält die kaufmännische Kalkulation. Dabei kann der Benutzer die einzelnen Positionen entweder als Absolutwerte oder relative Werte bearbeiten sowie Bemerkungen hinzufügen.

Der Verkaufspreis kann entweder pro Einheit oder als Gesamtpreis angegeben werden. Bei Fremdwährungsgeschäften kann der Benutzer die Fremdwährung angeben sowie einen dazugehörigen Wechselkurs.

Das System berechnet Vertriebs- und Verwaltungsgemeinkosten ein. Dieser Prozentsatz wird vom Administrator aktualisiert, in der Regel einmal jährlich.

Das System berechnet Gewinn/Verlust laufend automatisch und stellt dies dem Benutzer dar.

#### **7.1.4. Externe Berechnungen**

##### **Risikokalkulation**

Die Risikokalkulation (Abbildung 7.5) wird über eine Excel-Datei durchgeführt. Dabei sind Art und Größe des Risikos angegeben. Der Benutzer gibt zu jeder benötigten Position das Risikopotential prozentuell an. Außerdem kann er risikominimierende Maßnahmen und den Faktor der Risikoreduktion durch diese Maßnahmen angeben. Die Gesamtsumme aller Risiken wird automatisch berechnet und danach manuell in die Kalkulation in der PV eingefügt.

##### **On-/Off-Shore-Split**

Bei Geschäften, die über Landesgesellschaften abgewickelt werden, muss die STA Weiz der Landesgesellschaft einen gewissen Anteil des Verkaufspreises für die entstandenen Kosten (Anteil an Lieferkosten, Zoll) sowie einen Gewinnanteil abliefern. Dieser Anteil am Verkaufspreis ist der

## 7.1. IST-Zustand

The screenshot shows the 'Calculation' dialog box in SAP. The 'Factory Production Cost' section is active, displaying a table with columns: Total, Portion, Var, Fix, Red %, and per Unit. The table contains several rows of data, with some cells redacted with black boxes. Below the table, there are options for 'Accept TEO - Calc', a checkbox for 'convert budgetcalc to standardcalc', and radio buttons for 'Calculation of Order (TD:09651)' and 'Calculation of Offer (TBVA:09651)'. At the bottom, there are fields for 'Sales Price EUR', 'Sales Price USD', 'FOC', and 'Calc Unit', along with buttons for 'Excelexport T&D', 'internal profit (T&D)', 'external profit', 'Internal technical Offer', 'Delete', 'Print', 'Save', and 'Cancel'.

Abbildung 7.1.: Kalkulationsblatt „Factory Production Cost“. Aus Datenschutzgründen wurde diese Abbildung in Teilen geschwärzt.

The screenshot shows the 'Manufacturing Costs Detail' dialog box. It features a large blacked-out area on the left side and a smaller table on the right side. At the bottom of the dialog, there are two buttons: 'Accept' and 'Cancel'.

Abbildung 7.2.: Detailformular „Manufacturing Costs Detail“. Aus Datenschutzgründen wurden diese Abbildung in Teilen geschwärzt.

## 7.1. IST-Zustand

The screenshot shows a software window titled "Calculation" with a close button (X) in the top right corner. The window is divided into several sections:

- Sub Project Data:** A field for "SPNR" is present, with its content redacted.
- TED Data:** Fields for "Type" and "Unit" are present, with their contents redacted.
- Buttons:** Three buttons labeled "Factory Production Cost", "Factory Prime Cost", and "Total Cost" are located below the TED data. A "SAP Export OK" button is on the right.
- Factory Prime Cost Table:** A table with the following columns: "Total", "Portion", "Var", "Fix", and "perUnit". The table contains multiple rows of data, all of which are redacted.
- Input Fields:** Below the table, there are several input fields, some of which are redacted. These include fields for "Sales Price EUR", "Sales Price USD", "FOC", and "Calc Unit" (which has the value "1" entered).
- Bottom Controls:** A row of buttons and options at the bottom includes "Excelexport T&D", a radio button selection for "internal profit (T&D)" (selected) and "external profit", "Internal technical Offer", "Delete", "Print", "Save", and "Cancel".

**Abbildung 7.3.:** Kalkulationsblatt „Factory Prime Cost“. Aus Datenschutzgründen wurden diese Abbildung in Teilen geschwärzt.

## 7.1. IST-Zustand

The screenshot shows the SAP Calculation window with the following components:

- Sub Project Data:** SPNR [redacted]
- TEO Data:** Type [redacted], Unit [redacted]
- Buttons:** Factory Production Cost, Factory Prime Cost, Total Cost, SAP Export OK
- Total Cost Section:**
  - Factory Prime Cost: Total [redacted], Fix [redacted], per Unit [redacted]
  - Table with columns: ? Amount, in %, Remark
  - per Unit: [redacted]
  - Remark: [redacted] (Edit button)
  - Calc - Date: 05.06.2009
  - Exch. Rate: 1 EUR = [redacted] USD
  - Contr. Currency: USD
  - FOC: [redacted]
  - Sales Price: USD [redacted]
  - enter per unit
  - PriceBase: Firm
- Bottom Bar:**
  - Excelexport T&D
  - internal profit (T&D)
  - external profit
  - Internal technical Offer
  - Delete
  - Print
  - Save
  - Cancel

Abbildung 7.4.: Kalkulationsblatt „Total Cost“. Aus Datenschutzgründen wurden diese Abbildung in Teilen geschwärzt.

## 7.2. SOLL-Konzept

On-Shore-Anteil, der für die STA verbleibende Teil ist der Off-Shore-Anteil. Dies wird erst im Auftragsfall schlagend.

Die Berechnung dieser Anteile erfolgt über ein Excel-Formular (siehe Abbildung 7.6). Der Benutzer kann ein mit den wichtigsten Projektdaten gefülltes Formular aus der PV exportieren.

### 7.1.5. Probleme

Die momentane Umsetzung der Kalkulation in der PV stellt den Benutzer vor mehrere Probleme:

- Es gibt bisher keine Möglichkeit, ohne Nachfrage in der Technik die Stückzahl eines bestimmten Trafos zu erhöhen, sodass die Kalkulation entsprechend aktualisiert wird. Mengeneffekte werden in der PV nicht berücksichtigt.
- Das in der PV abgebildete Kalkulationsschema entspricht nicht mehr den Anforderungen der STA Weiz.
- Die On-/Off-Shore-Berechnung muss über eine externe Excel-Datei durchgeführt werden. Aufgrund von Zirkelbezügen muss dies in iterativen Schritten manuell durchgeführt werden.
- Ersatzteile (spare parts) können von Technik nicht direkt in die Kalkulation eingetragen werden.
- Es ist nicht möglich, eine Deckungsbeitragsrechnung in der PV durchzuführen.
- Aus Vertriebsicht zusammengehörende Positionen (Trafos bestehend aus mehreren Teilen, z.B. Autotrafo plus Einbaudrossel) können nicht gemeinsam, sondern müssen getrennt kalkuliert werden.
- Es gibt keine Möglichkeiten, Kalkulationen einzufrieren bzw. zu versionieren. Dadurch ist es zu einem späteren Zeitpunkt nicht ersichtlich, wie die Kalkulation z.B. zum Zeitpunkt der Angebotsabgabe ausgesehen hat.
- Die Risikokalkulation muss manuell über eine externe Excel-Datei durchgeführt werden.

## 7.2. SOLL-Konzept

### 7.2.1. Ziele

- Der Benutzer soll die kaufmännische Kalkulation mit der PV durchführen können, da sich dieses Konzept bewährt hat.

## 7.2. SOLL-Konzept

	A	B	C	D	E	F	G	H	I	J	K	L	M	
1														
2	<b>Risk- /Opportunity-Management STA, Werk Weiz</b>			<b>Q3 / FY2009</b>										
3	Proj. Name:													
4	Proj. No.:													
5	Client:													
6	Country:												Total Order Value	
7														
8														
9	No.	<b>Cost risks</b>		worst case										
10	1													
11	2													
12	3													
13	4													
14	5													
15	6													
16	7													
17	8													
18	9													
19	10													
20				Total						Total				
21														
22				worst case										
23	No.	<b>Contractual risks</b>												
24	11													
25	12													
26	13													

Abbildung 7.5.: Excel-Tabelle zur Durchführung der Risikokalkulation. Aus Datenschutzgründen wurden diese Abbildung in Teilen geschwärzt.

	A	B	C	D	E	F
1	<b>On- and Off-Shore Split</b>					#####
2						
3	Foreign Exchange Rate (2007-07-04):		1 EUR =			USD
4	<b>Total Selling Price to Customer</b>			\$		
5				\$		
6	<b>Subtotal without total erection</b>			\$		
7			EUR		\$	Basis: Subtotal
8				\$		
9				\$		
10			- gal	\$	\$	\$/gal
11				\$		
12				\$		
13	<b>Factory Price (CIF incoming port basis)</b>		EUR		\$	
14					\$	
15			EUR		\$	
16	<b>For Siemens internal use only!</b>					
17	Calculation of U.S. Customs Duty:					
18				\$		
19				\$		
20				\$		
21				\$		
22				\$		
23				\$		
24				\$		
25				\$		
26				\$	Shunt Reactor <input type="checkbox"/>	
27						
28	Sign. GF	Date	Sign. AL	Date	Sign. SB	Date

Abbildung 7.6.: Excel-Tabelle zur Berechnung des On-/Off-Shore-Splits. Aus Datenschutzgründen wurden diese Abbildung in Teilen geschwärzt.

## 7.2. SOLL-Konzept

A	B	C	D	E	F	G	H	I
2	<b>Projectname</b>				<b>Date/TEO</b>		20.07.2007	09.07.2007
3	<b>Customer</b>				<b>Dep.</b>		CRO	
4	<b>Final Customer</b>				<b>P-Manager</b>		KAM	
5	<b>Country Code</b>				<b>Repr.</b>			
6	<b>Countr Code Cust.</b>				<b>Incoterms</b>			
7	<b>Calculated Unit</b>	<b>FOC soll in Zukunft err. Werden</b>			<b>HK Faktor</b>		<b>1 EUR =</b>	
8	1						1,2500	USD
9	Herstellkosten Summe aller gelben Zeilen				diese Werte benötigt CRO um die korrekte Datenüberleitung aus der TBVA Nummer zu kontr. (Vergleich mit internen techn. Angebot			
10	Type							
11	No.	Description	ETG (CIF)		HK Red.	On-Shore Part in CC		HK Red.
12			Total	per Unit	%	Total	per Unit	%
13								
14								
15	140							
16	100							
17	110							
18	120							
19								
20								
21	130							
22	2							
23	3							
24	4							
25	5							
26	6							
27	7							
28								
29								
30								
31								
32	10							
33	300							
34	310							
35	320							
36								
37	330							
38	340							

**Abbildung 7.7.:** Kalkulationsschema NEU. Aus Datenschutzgründen wurden diese Abbildung in Teilen geschwärzt.

## 7.2. SOLL-Konzept

- Der Benutzer soll die Kalkulation mittels des neuen, SAP-nahen Kalkulationsschemas durchführen können, da das bisherige Schema nicht mehr den Bedürfnissen der STA Weiz entspricht.
- Der Benutzer soll die On-/Off-Shore-Rechnung als Teil der Kalkulation durchführen können, um das manuelle Berechnen des On-Shore-Anteils in einer externen Anwendung (Excel-Tabelle) sowie das dabei benötigte Kopieren von Projektdaten zwischen Anwendungen zu vermeiden.
- Der Benutzer soll die Risikokalkulation als Teil der Kalkulation durchführen können, um das manuelle Berechnen der Risikokalkulation in einer externen Anwendung (Excel-Tabelle) sowie das sowie das dabei benötigte Kopieren von Projektdaten zwischen Anwendungen zu vermeiden.
- Der Benutzer soll für Budget-Kalkulationen die technischen Kosten selbst bearbeiten können, da der Prozess in diesem Fall das Kontaktieren der technischen Abteilung nicht erfordert.
- Das System soll Mitarbeiter aus der Abteilung CRT und davon abhängige Abteilungen (CRS, LAL) automatisch über die Aufforderung, ein Angebot zu stellen, informieren, um die manuelle Erstellung einer solchen Anfrage zu vermeiden.
- Der Benutzer soll Versionen der Kalkulation einfrieren und später wieder abrufen können, um auch vorherige Versionen der Kalkulation nachvollziehen zu können.
- Das System soll bei der Änderung von Stückzahlen eines Subprojektes Mengeneffekte in der Kalkulation berücksichtigen, um eine Neuerstellung eines Internen Technischen Angebotes (wie im bisherigen System benötigt) zu vermeiden.

### 7.2.2. Rollen

#### CRO-Mitarbeiter

Der CRO-Mitarbeiter ist für die kaufmännische Kalkulation und die Gesamtkalkulation verantwortlich. Er trägt kaufmännische Daten ein, überschreibt wenn nötig Kosten aus der Technik- oder Service-Abteilung und wählt einen dazugehörigen Angebotspreis, wobei die Differenz als Gewinn ausgegeben wird.

Der zuständige CRO-Mitarbeiter wird als kaufmännischer Verantwortlicher für ein Projekt definiert.

## 7.2. SOLL-Konzept

### **CRT-Mitarbeiter**

Der CRT-Mitarbeiter ist für die technische Auslegung zuständig. Darüber hinaus beauftragt er einen CAE-Mitarbeiter zur Berechnung der Werksherstellkosten basierend auf der technischen Auslegung und führt eine Plausibilitätskontrolle dieser Kosten durch. Er beauftragt auch CRS und LAL.

Der zuständige CRT-Mitarbeiter wird als technischer Verantwortlicher für ein Projekt definiert.

### **CAE-Mitarbeiter**

Der CAE-Mitarbeiter führt eine Berechnung der Werksherstellkosten durch. Er wird vom CRT-Mitarbeiter dazu beauftragt. Die Kalkulation wird im Programm GROTWI durchgeführt und basiert auf der technischen Auslegung.

Der CRS-Mitarbeiter ist für das Service und im Zuge dessen für die Kalkulation der Service-Kosten verantwortlich. Dies geschieht auf Anfrage des für das Projekt verantwortlichen CRT-Mitarbeiters. Der zuständige CRS-Mitarbeiter wird als Verantwortlicher für das Service für ein Projekt definiert.

### **LAL-Mitarbeiter**

Der LAL-Mitarbeiter (Logistikabteilung Linz) ist für die Logistik und im Zuge dessen für die Kalkulation der Transport-Kosten zuständig. Dies geschieht auf Anfrage des für das Projekt verantwortlichen CRT-Mitarbeiters. Der zuständige LAL-Mitarbeiter wird als Verantwortlicher für den Transport für ein Projekt definiert.

### **CR**

Das Management des Vertriebs ist für die Entscheidung über die Angebotslegung verantwortlich.

### **MD**

Die Geschäftsführung ist für die letztendliche Bestätigung eines Angebots verantwortlich.

### **System**

Das System übernimmt die automatische Kalkulation eines Projektes. Die dazugehörigen Daten werden von den Benutzergruppen (CRO, CRT, CRS) im System gewartet.

### 7.2.3. Prozess

Die Kalkulation wird im Prozess hauptsächlich in den Phasen Angebotserstellung und Vertragsverhandlung benötigt. In der Angebotserstellung wird ein erstes Angebot erstellt, das sowohl technische als auch Service- und Logistikkalkulationen beinhaltet. Bei der Vertragsverhandlung kann es noch zu Veränderungen im Bereich der Kalkulation kommen.

Eine grafische Darstellung des Prozesses der Auftragsakquisition findet sich im am Ende dieses Dokumentes. Hierbei ist die Prozessvariante mit einem Budget Offer nicht inkludiert. Im Folgenden sind wesentliche Punkte des Prozesses aufgelistet:

#### Übersicht über den Prozess

1. CRO analysiert Kundenanfrage
2. CRO gibt Projekt für internes Angebot frei
  - „Product Main Data“-Dokument
  - Benachrichtigung soll automatisch in PV erfolgen
3. Internet Angebote werden erstellt
  - CRT erstellt technische Auslegung (externes Programm)
  - Benachrichtigung von CRS und LAL
  - Kalkulation der Werksherstellkosten in CAE (wird in PV importiert)
  - Plausibilitätsprüfung durch CRT
4. CRS erstellt Service-Kalkulation
5. LAL erstellt Logistik-Kalkulation
6. CRO importiert Kalkulation
7. CRO erweitert aggregierte Kalkulation um kaufmännische Zahlen

#### Herstellkostenupdate

Wenn die Gültigkeit des Angebots abgelaufen ist oder der Kunde Änderungen am Angebot wünscht, müssen die Herstellkosten upgedatet werden.

1. CRO benachrichtigt CRT
2. CRT benachrichtigt CAE

## 7.2. SOLL-Konzept

3. CAE führt Kalkulation von Werksherstellkosten durch
4. Plausibilitätsprüfung durch CRT

### 7.2.4. Szenarien

Dieser Abschnitt umfasst die wesentlichen Abläufe im Bereich Kalkulation der PV-NEU. Die Benennung von Menüpunkten ist exemplarisch, um das Verständnis zu erleichtern.

#### **Budget-Offer erstellen**

Der CRO-Mitarbeiter möchte ein Budget Offer für einen Kunden erstellen.

*Vorbedingung:* Zugehöriges Projekt/Unterprojekt bereits angelegt, Projekt geöffnet

1. CRO-Mitarbeiter öffnet Kalkulation zu Unterprojekt
2. System zeigt Kalkulationsübersicht an
3. CRO-Mitarbeiter wählt „Budget Offer“
4. System zeigt Kalkulationsmaske an
5. CRO-Mitarbeiter trägt (grob geschätzte) Werte ein
6. System berechnet gesamten Angebotspreis
7. CRO-Mitarbeiter wählt „Kalkulation speichern“
8. System speichert Kalkulation

*Nachbedingung:* Budget Offer-Kalkulation mit eingetragenen Werten zu Unterprojekt abgespeichert

#### **7.2.5. Kalkulation für CRT/CRS/LAL freigeben**

Der CRO-Mitarbeiter möchte CRT/CRS/LAL bitten, eine Kalkulation durchzuführen.

*Vorbedingung:* Zugehöriges Projekt/Unterprojekt bereits angelegt, Projekt geöffnet

1. CRO-Mitarbeiter öffnet Kalkulation zu Unterprojekt
2. System zeigt Kalkulationsübersicht an
3. CRO-Mitarbeiter wählt „Kalkulation freigeben“

## 7.2. SOLL-Konzept

4. System erstellt PMD
5. System leitet Benachrichtigung und Product Main Data an CRT weiter

*Nachbedingung:* Kalkulation für CRT/CRS/LAL freigegeben, entsprechende Stellen benachrichtigt

### Laden der Kalkulationen

Der CRO-Mitarbeiter möchte technische, Service- und Logistik-Kalkulation laden

*Vorbedingung:* Zugehöriges Projekt/Unterprojekt bereits angelegt, Projekt geöffnet, Kalkulation für CRT, CRS, LAL freigegeben, CRT, CRS und LAL-Kalkulation über Schnittstelle eingespielt

1. CRO-Mitarbeiter öffnet Kalkulation zu Unterprojekt
2. System zeigt Kalkulationsübersicht an
3. CRO-Mitarbeiter wählt „Kalkulationen laden“
4. System zeigt Maske an, um Art der zu ladenden Kalkulationen auszuwählen
5. CRO-Mitarbeiter wählt CRT, CRS und LAL
6. System lädt die Teile der CRT-, CRS- und LAL-Kalkulation in die PV-Kalkulation (vorher existierende Werte und Reduktionen werden überschrieben)
7. System berechnet Gesamtsumme
8. CRO-Mitarbeiter wählt „Kalkulation speichern“
9. System speichert Kalkulation

*Nachbedingung:* Kalkulation mit Werten der CRT-, CRS-, LAL-Kalkulation gespeichert

### Kalkulation einfrieren

Der CRO-Mitarbeiter möchte die aktuelle Version der Kalkulation unter „Angebotskalkulation“ abspeichern.

*Vorbedingung:* Zugehöriges Projekt/Unterprojekt bereits angelegt, Projekt geöffnet, Kalkulation mit Werten befüllt

1. CRO-Mitarbeiter öffnet Kalkulation zu Unterprojekt

## 7.2. SOLL-Konzept

2. System zeigt Kalkulationsübersicht an mit zuvor gespeicherter Kalkulation
3. CRO-Mitarbeiter wählt „Kalkulation einfrieren“
4. System zeigt Maske zur Parametereingabe an
5. CRO-Mitarbeiter gibt „Angebotskalkulation“ ein
6. System speichert aktuelle Kalkulation mit Timestamp unter „Angebotskalkulation“

*Nachbedingung:* Aktuelle Kalkulation unter Namen „Angebotskalkulation“ eingefroren

### **Eingefrorene Kalkulation laden**

Der CRO-Mitarbeiter möchte die zuvor eingefrorene Kalkulation „Angebotskalkulation“ laden und daran Werte ändern.

*Vorbedingung:* Zugehöriges Projekt/Unterprojekt bereits angelegt, Projekt geöffnet, Kalkulation unter dem Namen „Angebotskalkulation“ eingefroren

1. CRO-Mitarbeiter öffnet Kalkulation zu Unterprojekt
2. System zeigt Kalkulationsübersicht an mit zuvor gespeicherter Kalkulation
3. CRO-Mitarbeiter wählt „Eingefrorene Kalkulation laden“
4. System zeigt Liste mit eingefrorenen Kalkulationen an
5. CRO-Mitarbeiter wählt „Angebotskalkulation“
6. System ersetzt aktuelle Kalkulation durch eingefrorene Kalkulation
7. CRO-Mitarbeiter editiert Werte der Kalkulation
8. System aktualisiert automatisch berechnete Summen und Felder
9. CRO-Mitarbeiter wählt „Kalkulation speichern“
10. System speichert aktuelle Werte der Kalkulation

*Nachbedingung:* Werte der modifizierten eingefrorenen Kalkulation als aktuelle Kalkulation gespeichert. Eingefrorene „Angebotskalkulation“ bleibt unverändert.

### 7.2.6. Anforderungen

Die Anforderungen sind nach Rollen strukturiert. Für die CRO-Mitarbeiter wurde eine weitere thematische Unterteilung durchgeführt, um einen besseren Überblick zu gewährleisten.

#### **CRO-Mitarbeiter**

##### *Bearbeiten der Kalkulation*

- Der CRO-Mitarbeiter soll alle Felder der Kalkulation, die die kaufmännische Kalkulation betreffen, bearbeiten können.
- Der CRO-Mitarbeiter soll zu einem Subprojekt einen Verkaufspreis (auf Stückbasis) definieren können.
- Der CRO-Mitarbeiter soll die Summen-Felder, die die technische/Service/Logistik-Kalkulation betreffen, aufklappen können, um die einzelnen Positionen zu dieser Summe zu sehen.
- Der CRO-Mitarbeiter soll einzelne Positionen der technische/Service/Logistik-Kalkulation mit absoluten oder relativen %-Werten (HK-Reduktion) überschreiben können.
- Der CRO-Mitarbeiter soll auf Wunsch zur Kalkulation den On-Shore-Teil einblenden und bearbeiten können.

##### *Fremdwährungen*

- Der CRO-Mitarbeiter soll, wenn nötig, eine Fremdwährung zu einer Kalkulation und einen dazugehörigen Fremdwährungskurs (Kalkulationskurs) definieren können.
- Der CRO-Mitarbeiter, soll, wenn nötig, neben dem Kalkulationskurs einen gesicherten Fremdwährungskurs (Sicherungskurs) eingeben können (siehe Arbeitspaket Kurssicherung).
- Der Benutzer soll auswählen können, ob Übersichten in Fremdwährung (z.B. Gesamtverkaufspreis in Fremdwährung) mit dem Kalkulationskurs oder dem gesicherten Kurs berechnet werden.

##### *Einfrieren von Kalkulationen*

- Der CRO-Mitarbeiter soll die aktuelle Version der Kalkulation mit einem frei wählbaren Namen, Kommentar und automatisch generierten Zeitstempel einfrieren können.

## 7.2. SOLL-Konzept

- Der CRO-Mitarbeiter soll zuvor eingefrorene Kalkulationen angezeigt bekommen und auf Wunsch laden können, sodass die aktuelle Kalkulation durch die Werte der eingefrorenen Version überschrieben wird.

### ***Budget-Kalkulation***

- Der CRO-Mitarbeiter soll die Kalkulation eines Subprojekts als Budget-Kalkulation definieren können.

### ***Standard-Kalkulation***

- Der CRO-Mitarbeiter soll die Kalkulation eines Subprojekts als Standard-Kalkulation definieren können.
- Der CRO-Mitarbeiter soll ein Subprojekt zur technischen Kalkulation freigeben und ein Due Date für die Hinterlegung der technischen, Service- und Logistik-Kalkulation eingeben können.
- Der CRO-Mitarbeiter soll die technische, Service- und Logistik-Kalkulation zu einem Subprojekt *manuell* laden können. Dabei werden eventuell vorhandene HK-Reduktionen gelöscht.

### ***Risiko-Kalkulation***

- Der CRO-Mitarbeiter soll manuell einen Gesamtbetrag für das Risiko eingeben können.
- Der CRO-Mitarbeiter soll alternativ ein Formular zur Risikokalkulation öffnen und darin einzelne Risikopositionen und ihre Bewertung eingeben können.

### ***Berichte***

- Der CRO-Mitarbeiter soll eine interne Kalkulation exportieren können (genaues Format noch zu definieren).
- Der CRO-Mitarbeiter soll ein internes technisches Angebot (ITA) exportieren können (Bericht über technische Angebotskalkulation).
- Der CRO-Mitarbeiter soll ein Montageangebot (Service-Kalkulation) exportieren können.
- Der CRO-Mitarbeiter soll ein Transportangebot (Logistik-Kalkulation) exportieren können.
- Der CRO-Mitarbeiter soll einen On-/Off-Shore-Bericht exportieren können.
- Der CRO-Mitarbeiter soll eine Risikokalkulationsübersicht exportieren können.

## 7.2. SOLL-Konzept

### **CRT-Mitarbeiter**

- Der CRT-Mitarbeiter soll eine technische Auslegung mit einem Subprojekt verbinden können (über ein externes System).
- Der CRT-Mitarbeiter soll die Kalkulation der Werksherstellkosten überprüfen und bestätigen oder zurückweisen können.
- Der CRT-Mitarbeiter soll CAE, CRS und LAL über Aufforderung zur Erstellung einer Kalkulation benachrichtigen können

### **CAE-Mitarbeiter**

- Der CAE-Mitarbeiter soll die Kalkulation der Werksherstellkosten importieren können. Dabei sollen auch Ersatzteile (spare parts) importiert werden können.
- Der CAE-Mitarbeiter soll die Kalkulation der Werksherstellkosten zur Überprüfung durch den CRT-Mitarbeiter freigeben können.

### **CRS-Mitarbeiter**

- Der CRS-Mitarbeiter soll die Service-Kalkulation von für ihn freigegebenen Subprojekten bearbeiten können.
- Der CRS-Mitarbeiter soll die Service-Kalkulation als abgeschlossen markieren können.

### **LAL-Mitarbeiter**

- Der LAL-Mitarbeiter soll die Service-Kalkulation von für ihn freigegebenen Subprojekten bearbeiten können.
- Der LAL-Mitarbeiter soll die Service-Kalkulation als abgeschlossen markieren können.

### **System**

- Das System soll Trafos mit mehreren Teilen automatisiert in eine Kalkulation zusammenfassen (Trafos mit Einbautrafo/-drossel, Phasenschieber).
- Das System soll die Kalkulation im neuen Kalkulationsschema anzeigen.
- Das System soll bei Änderungen von Werten in der Kalkulation alle davon abhängigen Summen des Kalkulationsschemas automatisiert berechnen.

## 7.2. SOLL-Konzept

- Das System soll bei Änderungen automatisch folgende Gesamtaufschlüsselungen bieten:
  - Herstellkosten (pro Stück/gesamt, fix/variabel)
  - Verkaufspreis (pro Stück/gesamt, Eigenwährung/Fremdwährung)
  - Gewinn/Verlust (pro Stück/gesamt, absolut/prozentuell)
  - Deckungsbeitrag
- Das System soll bei der Erhöhung/Verminderung der Stückzahlen eines Subprojektes (z.B. 3 Trafos statt 2) Mengeneffekte automatisch berücksichtigen.
- Das System soll bei manueller prozentueller Kostenreduktion oder -erhöhung durch den CRO-Mitarbeiter automatisch den dazugehörenden Wert mit diesem Faktor neu berechnen.
- Das System soll beim Einfrieren von Kalkulationen alle Daten (Werte, Kommentare) der momentanen Kalkulation speichern.
- Das System soll den Zeitpunkt, an dem ein Subprojekt zur Kalkulation der technischen/Service/Logistik-Kosten freigegeben wurde, automatisch aus dem IST-Datum ermitteln, speichern und anzeigen.
- Das System soll den zuständigen CRT-/CRS-/LAL-Mitarbeiter benachrichtigen, wenn ein Subprojekt zur Kalkulation der technischen/Service/Logistik-Kosten freigegeben wurde.
- Das System soll den zuständigen CRO-Mitarbeiter benachrichtigen, wenn eine technische/Service/Logistik-Kalkulation als abgeschlossen gemeldet wurde. Das System soll die CRO-Kalkulation dabei *nicht* automatisch aktualisieren.

### 7.2.7. Verbesserungspotentiale

Das hier vorgestellte Konzept bietet folgende Verbesserungspotentiale:

- Durch die Integration externer Berechnungsfunktionalität in das System erfolgt einerseits eine Arbeitszeiterparnis, andererseits ist die Anfälligkeit für Copy&Paste-Fehler geringer. Darüber hinaus bedeutet die Integration der bisher externen Berechnungsschritte eine Steigerung der Datenqualität.
- Durch die Versionierung (Einfrieren) von Kalkulationen sind diese auch in einer früheren Version abrufbar. Dies resultiert in einer Steigerung der Datenqualität.

## 7.2. SOLL-Konzept

- Die Änderung von Stückzahlen unter Berücksichtigung von Mengeneffekten direkt in der PV resultiert einerseits in einer Arbeitszeiterparnis, andererseits lassen sich dadurch Schnittstelleneffekte ausnutzen.
- Durch eine gemeinsame Kalkulation von aus mehreren Bauteilen bestehenden Trafos erfolgt eine Arbeitszeiterparnis, darüber hinaus steigert dies auch die Datenqualität.

**Teil IV.**

**Abschluss**

## 8. Fazit und Ausblick

**Fazit** Im Zuge dieser Arbeit wurde das im Vertrieb der Siemens Transformers Austria Weiz eingesetzte System „Projektverwaltung“ (PV) analysiert und auf Probleme untersucht. Darauf aufbauend wurde für insgesamt zwölf Module jeweils ein Sollkonzept erstellt, das eine optimale Unterstützung der Vertriebsmitarbeiter bei ihrer täglichen Arbeit mit dem System gewährleisten soll. Dies geschah durch die Verwendung moderner Methoden des Requirements Engineering und der Modellierung von Informationssystemen. Dabei wurden etwa 50 Verbesserungspotentiale in den Bereichen Arbeitszeiterparnis, Reduktion der Fehleranfälligkeit, Steigerung der Datenqualität sowie Ausnutzen von Schnittstelleneffekten identifiziert. Die Entwicklung der PV-NEU erfolgt auf Basis dieser Sollkonzepte.

Als Grundlage für die Sollkonzepte wurde das bisherige System und die Prozesse, in denen das System genutzt wird, in Anlehnung an das ARIS-Framework aus unterschiedlichen Perspektiven – Organisation, Funktion, Daten und Prozess – betrachtet. Hierbei wurden Probleme mit dem bisherigen System identifiziert und dokumentiert, aber auch den Benutzer gut unterstützende Funktionalität als Grundlage für die weitere Konzeptionierung hervorgehoben. Ausgehend von der grundlegenden Problemstellung wurden Anforderungen an das System gewonnen, dokumentiert und validiert.

Die Vorgangsweise mit Kleingruppen im Workshop-Format erwies sich als gut geeignetes Mittel zur Gewinnung von Anforderungen. Hierbei konnten durch offene Diskussionen innovative Ideen, die Einzug in die Sollkonzepte hielten, direkt von den Benutzern gewonnen werden. Zugleich gewährleisteten die Kleingruppen eine effiziente Diskussion, insbesondere auch durch die zielbasierte Vorgehensweise, die den Fokus der Diskussion auf die für das jeweilige Modul wesentlichen Arbeitspunkte lenkte. Die Verwendung von Szenarien (Interaktionsfolgen zwischen Benutzer und System) unterstützte die Diskussion durch die Konkretisierung der zumeist abstrakten Ziele und Anforderungen; darüber hinaus können aus den Szenarien Testfälle für einen Abnahmetest erstellt werden. Dies resultierte in Sollkonzepten, die sich stark an den Bedürfnissen der zukünftigen Benutzer orientieren.

Die Tätigkeit im Zuge dieser Arbeit zeigte auch die Wichtigkeit der Validierung von Anforderungen, da insbesondere durch die kleine Anzahl direkt an der Konzeptionierung eines spezifischen Moduls beteiligter Mitarbeiter die Gefahr bestand, wichtige Anforderungen zu übersehen

## 7.2. SOLL-Konzept

oder Mehrwerte falsch einzuschätzen. Die Validierung erfolgte durch Besprechen der wesentlichen Punkte der Sollkonzepte innerhalb einer Großgruppe. So gefundene Probleme und Kritikpunkte wurden im weiteren Verlauf der Konzeptionierung in die Sollkonzepte aufgenommen.

**Ausblick** Die Sollkonzepte dienen als Basis zur Erstellung eines Pflichtenhefts. In weiterer Folge übernimmt ein externer Softwarelieferant die Entwicklung und Wartung der PV-NEU. Für die Wartungsphase ist ein klar definierter Wartungsprozess mit Releases in festen Zeitabständen empfehlenswert; dies vor allem vor dem Hintergrund, dass viele der Probleme der aktuellen PV durch unklare und sich oft ändernde Benutzeranforderungen, die zumeist innerhalb eines kurzen Zeitraums in das System implementiert wurden, entstanden. Hierbei könnte sich ein agiler Entwicklungsprozess wie Extreme Programming oder das darauf aufbauende Scrum-Framework als geeignetes Mittel erweisen.

**Teil V.**

**Anhang**

## A. Literatur

- Balzert, Helmut (1996). *Lehrbuch der Software-Technik. Software-Entwicklung*. Bd. 1. Spektrum Akademischer Verlag. ISBN: 3-8274-0042-2.
- Banga, Balbinder et al. (2003). »Natural resources information management forum. Putting knowledge to work«. In: Hrsg. von T. Innes. FORREX. Kap. Gathering User Needs: A Case Study on the Process Used to Gather User Needs for the Adaptive Management Area's Forestry Portal, S. 123–131.
- Beck, Kent (2000). *Extreme Programming. Die revolutionäre Methode für Softwareentwicklung in kleinen Teams*. Addison-Wesley.
- Boehm, Barry W. (1981). *Software engineering economics*. New York, NY, USA: Prentice Hall PTR, S. 641–686. ISBN: 3-540-43081-4.
- (1987). »Industrial software metrics top ten list«. In: *IEEE Software* 4.5, S. 84–85.
- Boehm, Barry W. und Victor R. Basili (2001). »Software Defect Reduction Top 10-List«. In: *IEEE Computer Society*, S. 135–137.
- Broy, Manfred et al. (2007). »Ein Requirements-Engineering-Referenzmodell«. In: *Informatik-Spektrum* 30.3, S. 127–142.
- Chen, Peter Pin-Shan (1976). »The entity-relationship model—toward a unified view of data«. In: *ACM Transactions on Database Systems (TODS)* 1.1, S. 9–36.
- Cooper, Alan und Robert Reimann (2003). *About face 2.0. The essentials of interaction design*. Wiley. ISBN: 0-7645-26413.
- Davis, Rob und Eric Brabänder (2007). *ARIS Design Platform. Getting Started with BPM*. Springer. ISBN: 978-1-84628-612-4.
- Finkelstein, A. und J. Dowell (1996). »A comedy of errors: the London Ambulance Service case study«. In: *Proceedings of the 8th International Workshop on Software Specification and Design*. IEEE Computer Society Press, S. 2–4.
- Gadatsch, Andreas (2008). *Grundkurs Geschäftsprozessmanagement*. 5. Aufl. Vieweg Verlag. ISBN: 978-3-8348-0363-4.
- Hall, T. et al. (2002). »Requirements problems in twelve software companies: an empirical analysis«. In: *Software, IEEE Proceedings* 149.5 (Okt. 2002), S. 153–160. ISSN: 1462-5970.

## 7.2. SOLL-Konzept

- Hofmann, Hubert F. und Franz Lehner (2001). »Requirements Engineering as a Success Factor in Software Projects«. In: *IEEE Software* 18.4, S. 58–66.
- Holtzblatt, Karen und Hugh R. Beyer (1998). »Requirements gathering: the human factor«. In: *Communications of the ACM* 38.5 (Mai 1998), S. 31–32.
- IEEE-SA Standards Board (1998). *IEEE 830-1998. IEEE Recommended Practice for Software Requirements Specifications*. The Institute of Electrical und Electronics Engineers, Inc. ISBN: 0-7381-0332-2.
- Keller, G. et al. (1992). »Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK)«. In: *Veröffentlichungen des Instituts für Wirtschaftsinformatik*. Hrsg. von August-Wilhelm Scheer. 89.
- Kleuker, Stephan (2009). »Grundkurs Software-Engineering mit UML«. In: Vieweg+Teubner. Kap. Anforderungsanalyse, S. 51–85.
- Kniberg, Henrik (2007). *Scrum and XP from the Trenches. How we do Scrum*. C4Media. ISBN: 978-1-4303-2264-1.
- Leffingwell, Dean und Don Widrig (2000). *Managing Software Requirements. A Unified Approach*. Addison-Wesley.
- MCP (2009a). »Abschlusspräsentation Projektverwaltung«. Interne Präsentation. 9. Sep. 2009.
- (2009b). »Steering Meeting. STA Weiz Projekt PPS 15.000 MVA+«. Interne Präsentation. 13. März 2009.
- Mielke, Carsten (2002). *Geschäftsprozesse. UML-Modellierung und Anwendungsgenerierung*. Spektrum Akademischer Verlag. ISBN: 3-8274-1132-7.
- Möller, Karl-Heinrich (1996). »Software-Metriken in der Praxis«. In: Hrsg. von Christof Ebert und Reiner Dumke. Springer. Kap. Software-Metriken in der Praxis, S. 105–116.
- Mylopoulos, J. (1998). »Information Modeling in the Time of the Revolution«. In: *Information systems* 23.3-4, S. 127–155.
- Nuseibeh, Bashar und Steve Easterbrook (2000). »Requirements engineering: a roadmap«. In: *Proceedings of the Conference on The Future of Software Engineering*, S. 35–46.
- OMG (2009a). *Business Process Model and Notation. Version 1.2*. 3. Jan. 2009. URL: <http://www.omg.org/spec/BPMN/1.2/PDF/> 24. 11. 2009.
- (2009b). *Business Process Model and Notation (BPMN). FTF Beta 1 for Version 2.0*. Aug. 2009. URL: <http://www.omg.org/cgi-bin/doc?dte/09-08-14.pdf> 24. 11. 2009.
- Openheim, A.N. (1999). *Questionnaire Design, Interviewing and Attitude Measurement*. Pinter.
- Oracle (2009). *Oracle Rdb Statement of Direction*. URL: [http://www.oracle.com/technology/products/rdb/htdocs/rdb7/rdb\\_statement\\_of\\_direction.html](http://www.oracle.com/technology/products/rdb/htdocs/rdb7/rdb_statement_of_direction.html) 09. 11. 2009.

## 7.2. SOLL-Konzept

- Owen, Martin und Jog Raj (2003). *BPMN and Business Process Management. Introduction to the New Business Modelling Standard*. Popkin Software. URL: [http://www.bpmn.org/Documents/6AD5D16960.BPMN\\_and\\_BPM.pdf](http://www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf) 24. 11. 2009.
- Partsch, Helmuth (1998). *Requirements-Engineering systematisch. Modellbildung für software-gestützte Systeme*. Springer. ISBN: 3-540-64391-5.
- Pohl, Klaus (2008). *Requirements Engineering. Grundlagen, Prinzipien, Techniken*. 2. Aufl. dpunkt.verlag. ISBN: 987-3-89864-550-8.
- Porter, Michael E. (1985). *Competitive Advantage: Creating and Sustaining Superior Performance*. The Free Press. ISBN: 0-684-84146-0.
- Project Cartoon und Gernot R. Bauer (2009). *Wie Projekte wirklich funktionieren (in 6 Abbildungen)*. URL: <http://www.projectcartoon.com/cartoon/60031/> 16. 09. 2009.
- Recker, Jan et al. (2005). »Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN«. In: *16th Australasian Conference on Information Systems*.
- Scheer, August-Wilhelm (1990). *EDV-orientierte Betriebswirtschaftslehre*. 4. Aufl. Springer, S. 327. ISBN: 9780387523972.
- (1996). »ARIS-House of Business Engineering. Von der Geschäftsprozessmodellierung zur Workflow-gesteuerten Anwendung; vom Business Process Reengineering zum Continuous Process Improvement.« In: *Veröffentlichungen des Instituts für Wirtschaftsinformatik Heft 133*. 133.
- Scheer, August-Wilhelm und W. Jost (1996). »Geschäftsprozessmodellierung innerhalb einer Unternehmensarchitektur«. In: *Geschäftsprozessmodellierung und Workflow-Management*. Hrsg. von G. Vossen, S. 29–46.
- Scheer, August-Wilhelm et al. (1995). »Rahmenkonzept für ein integriertes Geschäftsprozessmanagement«. In: *Wirtschaftsinformatik 37.5*, S. 426–434.
- Siemens (2008). »PM@Transformers. Projektmanagement-Richtlinie für Power Transformer Projekte innerhalb Energy T TR«. Internes Dokument. Nov. 2008.
- Siemens Transformers Austria (2009). »PV. Ein kurzer Überblick«. Internes Dokument.
- Sommerville, Ian (2001). *Software Engineering*. 6. Aufl. Pearson Studium. ISBN: 3-8273-1001-9.
- (2005). »Integrated Requirements Engineering. A Tutorial«. In: *IEEE Software* 22.1, S. 16–23.
- Stephens, Matt und Doug Rosenberg (2003). *Extreme Programming Refactored. The Case Against XP*. Apress. ISBN: 1590590961.
- The Standish Group (1995). *Chaos Report*. URL: <http://www.projectsmart.co.uk/docs/chaos-report.pdf> 13. 09. 2009.
- White, Stephen A. (2004). *Introduction to BPMN*. IBM. Apr. 2004. URL: [http://www.bpmn.org/Documents/Introduction\\_to\\_BPMN.pdf](http://www.bpmn.org/Documents/Introduction_to_BPMN.pdf) 24. 11. 2009.

## 7.2. SOLL-Konzept

Yu, Eric (1997). »Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering«. In: *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE'97)*, S. 226–235.

## B. Abbildungsverzeichnis

2.1. Humoristische Darstellung der Problematik in Softwareprojekten . . . . .	5
2.2. Gründe für das Scheitern oder Nicht-Einhalten von Zielvorgaben in Software-Projekten . . . . .	6
2.3. Relative Fehlerbehebungskosten . . . . .	7
2.4. Wasserfall-Modell . . . . .	9
2.5. Ablauf der Anforderungsanalyse . . . . .	12
2.6. Probleme und Lösungen im Requirements Engineering . . . . .	13
2.7. Szenarien als Middle-Level-Abstraction . . . . .	18
3.1. Integriertes Prozess- und Workflow-Management . . . . .	29
3.2. ARIS-Haus des Business Engineerings . . . . .	31
3.3. ER-Notation . . . . .	32
3.4. Zerlegung eines Prozesses auf unterschiedlichen Ebenen . . . . .	34
3.5. Flow Objects in BPMN . . . . .	37
3.6. Connecting Objects in BPMN . . . . .	37
3.7. Swimlanes in BPMN . . . . .	37
3.8. Artifacts in BPMN . . . . .	38
4.1. Geplante Systemlandschaft innerhalb der STA Weiz . . . . .	43
4.2. Projektplan Projekt PPS 15.000 MVA+ . . . . .	44
5.1. Identifizierte Anforderungsquellen . . . . .	49
5.2. Iterative Vorgangsweise zur Erarbeitung des Sollkonzeptes . . . . .	51
6.1. Die PV im Kontext der Systemlandschaft der STA Weiz . . . . .	57
6.2. Organisation der STA . . . . .	59
6.3. Skizze der Datenbankstruktur . . . . .	62
6.4. Akquisitions- und Projektabwicklungsprozess . . . . .	63
7.1. Kalkulationsblatt „Factory Production Cost“ . . . . .	67

## Abbildungsverzeichnis

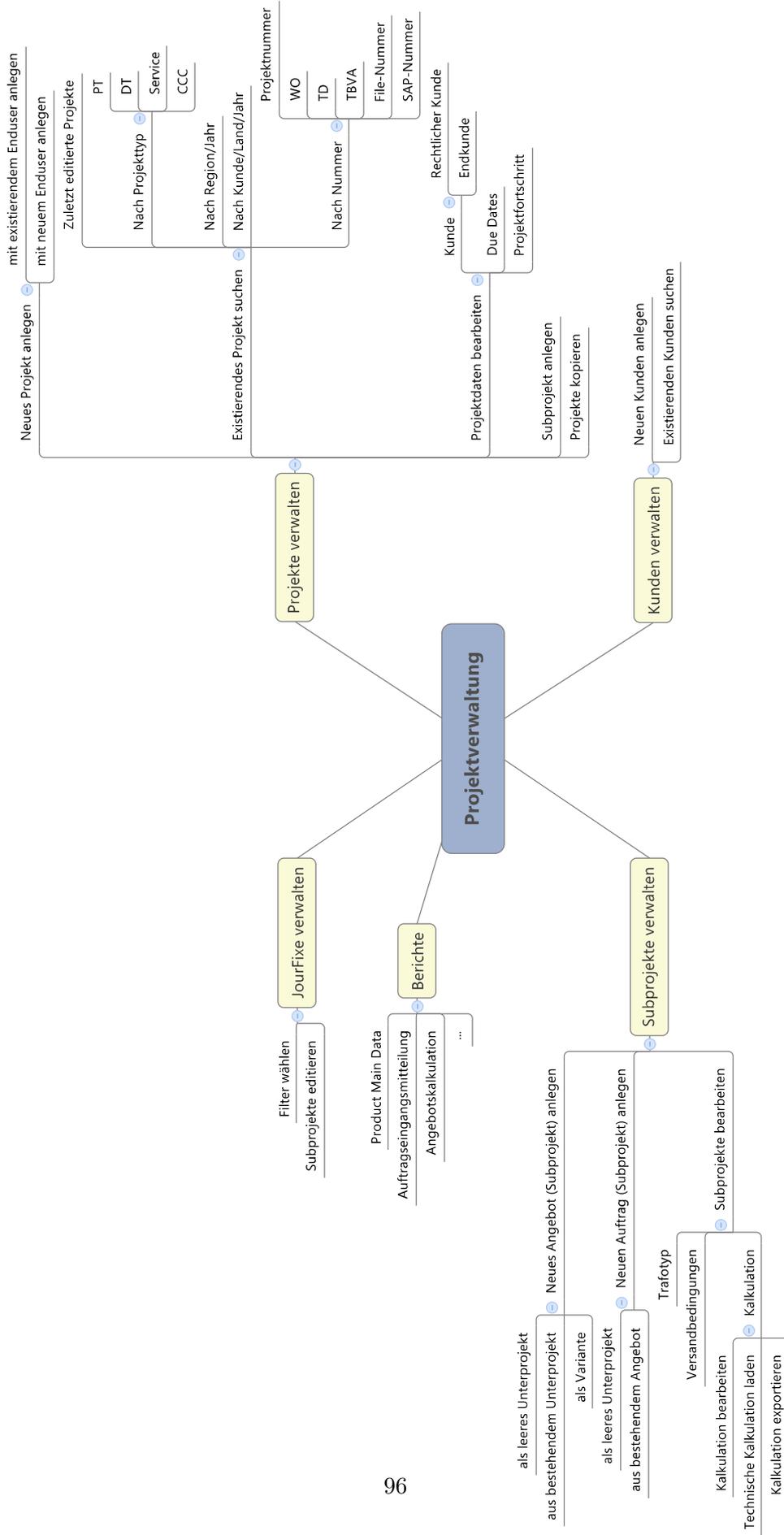
7.2. Detailformular „Manufacturing Costs Detail“ . . . . .	67
7.3. Kalkulationsblatt „Factory Prime Cost“ . . . . .	68
7.4. Kalkulationsblatt „Total Cost“ . . . . .	69
7.5. Risikokalkulation . . . . .	71
7.6. On-/Off-Shore-Split . . . . .	71
7.7. Kalkulationsschema NEU . . . . .	72

## C. Tabellenverzeichnis

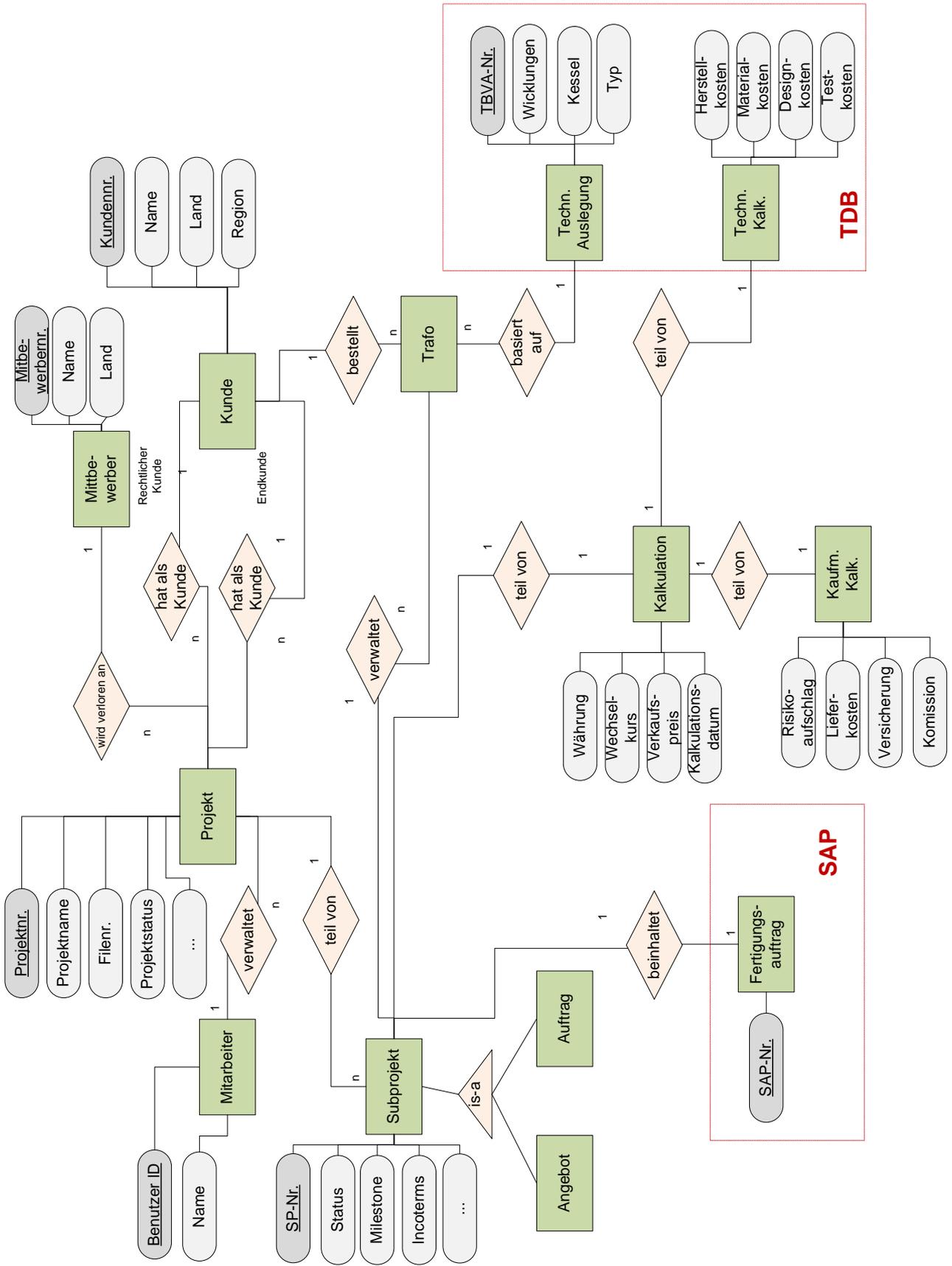
2.1. Probleme im Anforderungsprozess . . . . .	6
2.2. Eignung von Szenariodokumentationsarten . . . . .	21
2.3. Übersicht der Gewinnungstechniken im Requirements Engineering . . . . .	22
5.1. Projektphasen im Zuge der Spezifikation der PV . . . . .	47
5.2. Hauptsächlich eingesetzte Methoden in den jeweiligen Projektphasen . . . . .	48

## D. Diagramme

# Überblick über die Funktionalität der PV



# IST-Analyse der Datenstrukturen der PV



# Auftragsakquisitionsprozess

